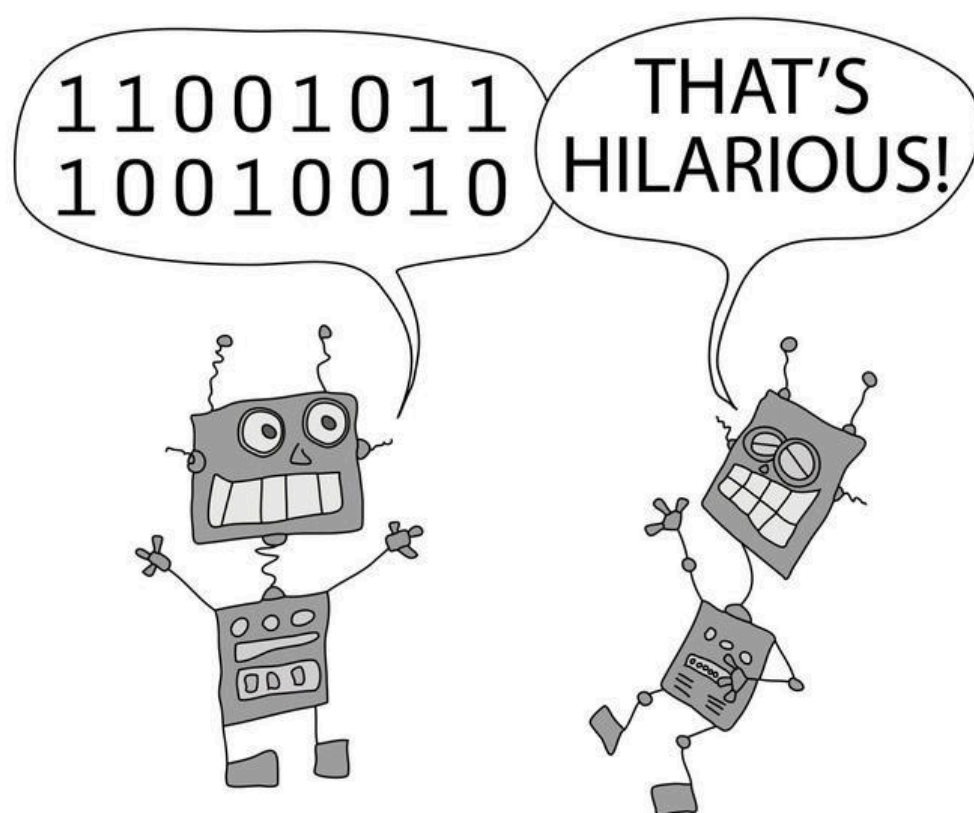


A Guide to IB Computer Science



CHAPTER ONE - SYSTEM FUNDAMENTALS	7
PLANNING A NEW SYSTEM	7
WHY DOES CHANGE NEED TO BE MANAGED?	7
COMPATIBILITY ISSUES	8
SOFTWARE – AS – A SERVICE	8
INSTALLING SYSTEMS	9
PARALLEL RUNNING	9
PILOT RUNNING	9
DIRECT CHANGEOVER	10
PHASED CONVERSION	10
TESTING A SYSTEM	11
DATA MIGRATION	12
USER DEVELOPMENT	13
USER DOCUMENTATION	14
SYSTEM BACKUP	14
KEEPING SOFTWARE UP TO DATE	16
SYSTEMS DESIGN BASICS	16
SYSTEMS DESIGN AND ANALYSIS	18
THE SYSTEMS DESIGN CYCLE:	19
ANALYSIS STAGE: DATA COLLECTION METHODS	19
DESIGN STAGE	21
SYSTEMS FLOW CHART	21
DATA FLOW DIAGRAMS	23
STRUCTURED CHARTS	24
HUMAN INTERACTION WITH THE SYSTEM	25
ACCESSIBILITY	25
USABILITY	26
CHAPTER ONE – REVIEW QUESTIONS	28
SHORT RESPONSE QUESTIONS	28
EXTENDED RESPONSE QUESTIONS	28
CHAPTER TWO - COMPUTER ORGANISATION	29
COMPUTER ARCHITECTURE	29
WHAT ROLE DOES MEMORY PLAY?	30
MACHINE INSTRUCTION CYCLE:	31
SECONDARY MEMORY	31
OPERATING SYSTEMS	33
WHAT IS THE JOB OF AN OPERATING SYSTEM?	34
APPLICATION SOFTWARE	37
BINARY REPRESENTATION	38
BINARY CONVERSION	40
CONVERSION FROM DENARY (DECIMAL) TO BINARY	41
BINARY ADDITION	41
WHAT IS THE PURPOSE OF HEXADECIMAL?	42
FILE EXTENSIONS	42
TEXT	43
PICTURES/GRAPHICS	43
CALCULATING MEMORY REQUIREMENTS	46
METADATA	46
COMPRESSION	46
BOOLEAN LOGIC	48
BOOLEAN OPERATORS	48

USING BOOLEAN EXPRESSIONS	48
CIRCUIT DIAGRAMS	50
CHAPTER TWO – REVIEW QUESTIONS	51
<u>CHAPTER THREE - NETWORKS</u>	<u>54</u>
NETWORK FUNDAMENTALS	54
TYPES OF NETWORKS:	54
VIRTUAL PRIVATE NETWORKS (VPN)	55
DATA TRANSMISSION	56
OPEN SYSTEMS INTERCONNECTION (OSI MODEL)	56
PROTOCOLS	57
SENDING DATA	58
DATA PACKETS:	58
HOW IS DATA TRANSFERRED?	59
PACKET SWITCHING	59
CIRCUIT SWITCHING	59
TRANSMISSION MEDIA:	60
WIRELESS NETWORKS	60
COMPONENTS OF A WiFi NETWORK:	61
FUTURE DEVELOPMENTS:	62
NETWORK SECURITY	62
PHYSICAL	62
CODED	62
CHAPTER THREE – REVIEW QUESTIONS	63
<u>CHAPTER FOUR - COMPUTATIONAL THINKING</u>	<u>64</u>
WHAT IS COMPUTATIONAL THINKING?	64
THINKING PROCEDURALLY	65
THINKING LOGICALLY	68
CHAPTER FOUR – REVIEW QUESTIONS	77
<u>CHAPTER FIVE-RESOURCE MANAGEMENT (HL ONLY)</u>	<u>79</u>
WHAT IS INSIDE A MACHINE?	79
THE OPERATING SYSTEM	82
MANAGING MEMORY	83
MANAGING VIRTUAL MEMORY	84
MANAGING PERIPHERALS	85
VIRTUALIZATION OF THE SYSTEM	85
WHEN THINGS GO WRONG!	86
DEDICATED OPERATING SYSTEM	86
CHAPTER FIVE – REVIEW QUESTIONS	87
<u>CHAPTER SIX - CONTROL (HIGHER LEVEL ONLY)</u>	<u>88</u>
SENSORS	88
MAKING A SYSTEM	89
SAMPLING	90
EMBEDDED SYSTEMS	91
LOOP SYSTEMS	93
OPEN LOOP SYSTEMS	93
CLOSED LOOP SYSTEM	94
DISTRIBUTED VS. CENTRALISED SYSTEMS	95
AUTONOMOUS AGENTS	95

CHAPTER SIX - REVIEW QUESTIONS	98
WHAT IS PROGRAMMING?	100
HIGH LEVEL LANGUAGES	101
TRANSLATION	101
OBJECT ORIENTATION	103
WHAT IS IT?	103
WHAT IS A CLASS?	103
INHERITANCE	105
PARAMETERS	108
VARIABLES	109
VARIABLE TYPES	109
DECLARING VARIABLES	110
STRINGS	110
SCOPE OF VARIABLES	110
CONSTANT VARIABLES	111
CASTING	111
RELATIONAL OPERATORS FOR PRIMITIVE VARIABLES	111
RELATIONAL OPERATORS FOR STRING VARIABLES	112
BOOLEAN OPERATORS	112
METHODS	113
METHOD SIGNATURES	113
RETURN VALUES	113
CONSTRUCTOR METHODS	114
ACCESSOR METHODS	114
MUTATOR METHODS	114
RESERVED WORDS ASSOCIATED WITH METHODS	115
ENCAPSULATION	117
LIBRARIES	119
IF STATEMENTS	121
LOOPS	123
CONDITIONAL LOOPS:	123
FIXED LOOPS:	124
ARRAYS	125
DECLARING ARRAYS	126
ARRAYS OF CLASSES	128
STANDARD ALGORITHMS	130
TWO DIMENSIONAL ARRAYS	133
SORTING ALGORITHMS	136
ALGORITHM EFFICIENCY	136
ADDITIONAL PROJECTS - SL	141
PROJECT ONE – THE SUPERHERO DECIDER	141
PROJECT TWO – PASSWORD PROTECTOR	141
PROJECT THREE – THE ALIAS GENERATOR	142
PROJECT FOUR – AMAZON WISHLIST	143
CHAPTER EIGHT - PROGRAMMING – HIGHER LEVEL	144
RECURSION	144
ABSTRACT DATA STRUCTURES (ADT's)	148
ADT – ARRAY LIST	149
ADT – LINKED LIST	151
ADT – STACK	156
ADT – QUEUE	158
ADT – BINARY TREE	160

CONVENTIONS	168
PRACTICAL ONLY – FILE INPUT AND OUTPUT	169
SIMPLE FILE I/O	169
FILE I/O USING BYTES	170
FILE I/O - RANDOM ACCESS FILES	172
FILE I/O – USING FILES WITH ADT’S	175
EXTENSION – GRAPHICAL USER INTERFACES	178
ADDITIONAL PROJECTS – HL	181
PROJECT ONE – PENGUINS WITH PROBLEMS	181
PROJECT TWO – BINARY CONVERTER	183
PROJECT THREE – DISTANCE CALCULATOR	183
PROJECT FOUR – PIG LATIN	183
PROJECT FIVE – VOWEL COUNTER	183
PROJECT SIX – PALINDROME CHECKER	184
PROJECT SEVEN – FORTUNE TELLER	184
PROJECT EIGHT – RANDOM GIFT SUGGESTOR	184
PROJECT NINE – RECIPE MANAGER	184
PROJECT TEN – VENDING MACHINE	184
PROJECT ELEVEN – JOSEPHUS PROBLEM	185
PROJECT TWELVE – FAMILY TREE CREATOR	185
PROJECT THIRTEEN – DOCTOR SCHEDULER	185
PROJECT FOURTEEN – THE PASSWORD SAFE	185
PROJECT FIFTEEN – QUIZ MAKER	186
PROJECT SIXTEEN – THE COLLECTOR	186
PROJECT SEVENTEEN – HANGMAN	186
PROJECT EIGHTEEN – PLAY YOUR CARDS RIGHT	186
PROJECT NINETEEN – MAGIC EIGHT BALL	187
PROJECT TWENTY – BATTLE SHIPS	187
PROJECT TWENTY ONE – GRAND NATIONAL	187
PROJECT TWENTY TWO – POKER	188
CHAPTER SEVEN / EIGHT - REVIEW QUESTIONS	189
 CHAPTER NINE – INTERNAL ASSESSMENT	 192
WHY IS IT IMPORTANT?	192
WHAT IS IT?	192
CHOOSING A TOPIC	193
ACADEMIC HONESTY	194
PLANNING SECTION - MARKS	195
SOLUTION OVERVIEW - MARKS	196
DEVELOPMENT - MARKS	197
FUNCTIONALITY AND EXTENSIBILITY OF PRODUCT - MARKS	198
EVALUATION - MARKS	199
CHOOSING A TOPIC - IDEAS SHEET	200
PROPOSAL FORM	201
PLANNING SECTION HELP DOCUMENTS	203
SOLUTION OVERVIEW HELP DOCUMENTS	204
UML DIAGRAM	204
CLASS DIAGRAMS	205
EXPLANATION	205
RECORD OF TASKS	205
DEVELOPMENT – HELP SHEET	206
VIDEO HELP SHEET	207
EVALUATION HELP SHEET	208

EXAMPLE PLANNING SECTION	209
EXAMPLE SOLUTION OVERVIEW SECTION	210
DEVELOPMENT EXAMPLE	215
EXAMPLE EVALUATION SECTION – CRITERION E	219
APPENDIX ONE – PENGUINS FILE	222

Chapter One - System Fundamentals

ATL Skills:

Communication: Use appropriate forms of writing for different purposes and audiences.

Thinking: Use models and simulations to explore complex systems and issues

...s: "Careful planning is required in complete a project"

...ted"?

...hack?

...nology increased our standard of

...sfully gathered from clients?

...e successfully delivered?

...ally matter or can we be ad hoc?

...ces into businesses?

...cal and online do businesses and

The purpose of this unit is to investigate why systems are needed. We no longer live in a world where many processes are carried out on paper but new systems are still being created all the time. Why if most things are computerized anyway do we need new systems? We will also look at how systems analysts get data from clients, how systems are implemented and how we train staff to use

these systems. This unit will also deal with the legal and ethical issues surrounding computer systems.

Planning a new system

Systems that have been in place for a long time often become unusable for several reasons:

- The hardware they are based upon or interact with has become unusable and obsolete.
- The software they interact with has been replaced and the software can no longer communicate with each other.
- The company, which uses the software, has changed and requires the system to reflect this.

Once the company recognizes the need for a new system they often call in system analysts. These analysts will investigate what is required in the new system and help the company to work through the issues surrounding the implementation of new system. Often people have high expectations of a new system, such as, it will solve all the problems or it will make life easier. Systems analysts help companies to understand that while a new system will help to solve many of the current problems there will still be issues that need to be addressed by employees.

Why does change need to be managed?

When changing a system there are many factors that need to be managed both technologically and with people. Firstly, technology systems have to be implemented to a timetable. This means the designers, the coders, the testing and the implementation need to be kept on schedule. The manager needs to ensure that each phase sticks to the schedule as far as possible and if there are any unforeseen circumstances these need to be managed. Employees during this time will naturally be nervous, people are always wary of change. People maybe fearing the loss of their job or how lost they will feel when the new system is in place. It is the job of the manager to reassure people that there will be adequate training for use with the new system and also that if there are redundancies as a result of the new system they will be handled fairly.

Compatibility Issues

Compatibility in this context means the ability of one system to communicate with another system and is especially important when thinking about systems that are global. To be able to communicate with others is one of the fundamental requirements of global businesses. Therefore you need to ensure that the relevant stakeholders can read documents and files produced by the system you create. Whenever possible standard file formats should be used and when coding a character set that is capable of representing many different languages. Another compatibility issue to consider is the differences in international standards for example 2.4.16 in Europe is the 2nd April 2016 however in American this would be the 4th February 2016. If you are transferring between systems it is important to consider the issues that can occur with this.

Software – as – a Service

When planning systems it is necessary to consider whether you will pay for the software and use your own machines to run the system or whether you will use service software. Software-as-a-Service is usually subscription based software and is hosted on the providers servers not the companies. It can be described as “renting” software as you only have the software while you pay or are subscribed to it. Some examples of software as a service are: oneDrive, Microsoft Office online, the Adobe suite online, Google Drive, Box.com.

Some examples of SAAS are:

- No additional hardware costs, all the cost associated with running the program are contained on the host server.
- No initial set up costs, only the subscription
- You pay for what you use, therefore if only certain people need certain applications they only pay for them applications.
- Updates are automatic, the company hosting the software rolls out the updates, and therefore the software is always up to date.
- There is cross device compatibility, often people bring in devices running different O/S's with SAAS companies don't need to be as concerned about this
- Access can be from anywhere, for example a company using Dropbox can allow employees access anywhere in the world.
- Some SAAS has interactive editing (e.g. Google Drive) this means that members of the same team can work collaboratively on a project they don't necessary need to be together, which saves the company money.

Drawbacks of SAAS:

- The data stored by the program is often held on the host server, which does lead to data security issues.
- Some business that require integration with customer systems may have trouble as this may large volumes of sensitive data being stored externally.
- If the data is all stored in the cloud limited access or slow access to web services becomes an issue.
- The laws governing the data are in host's country not necessarily the same country as the company. This is especially a problem when using American companies from

Europe, as the laws in America are not as stringent as European laws when governing what can be done with our data.

- If the host company goes out of business then the company may lose the data being stored on the host servers.
- If the updates are out of control of the company then training issues may arise if there is a big change without notice.

Installing Systems

After a system has been planned it needs to be installed, this is a delicate process, as it requires the company to put the trust of all their data into the system for the first time. There are several different options for transferring the data each comes with their own associated benefits and risks.

Parallel Running

In this installation procedure the old system and the new system are running alongside each other.

Advantages	Disadvantages
- Employees have time to learn the new system at a pace that suits them. - The old system can be used as a back up in case of a new system failure. - There will be time to learn if there are any unforeseen problems with the new system while the old system is still in place.	- It is expensive to retain two systems at the same time. - It may take the employees longer to learn as they have to use the old system while using the old system.

Pilot Running

In this installation procedure a small group of users begin using the systems (a select group of employees). This group then reports back to the development team highlighting any issues or items, which need changing before the system is rolled out to the whole company.

Advantages	Disadvantages
- Problems with the system will be rectified before the system is given to everyone. - Small groups of users using the system are easier to manage. - The pilot allows the system to be 'perfected' before it	- It takes longer than direct change over, as you have to have the pilot scheme running for a significant time before it rolled out. - The select users may not use the system to the same extent as other users. - It is quite expensive to have the old system and

is passed onto the whole group.	the pilot system running at the same time.
---------------------------------	--

Direct Changeover

In the direct changeover process the new system is put in place and the old system removed immediately.

Advantages	Disadvantages
- It is cheaper than maintaining two systems at the same time. - There is no duplication of data as only one system is being used.	- There is a steep learning curve for staff. - If something goes wrong there is no back up system in place so the company might be out of action until it is fixed.

Phased Conversion

The phased conversion strategy allows one section of the system to be changed over at a time. For example one department might start using the new system and become comfortable with it before moving onto the next department.

Advantages	Disadvantages
- Training can be focused on one department at a time. - The conversion process is smoother per department as any help can be focused toward them. - There is a back up system in place, in case anything goes wrong.	- Maintaining two systems is expensive. - Switching from one system to the next over departments may mean that departments cannot share data until they are all using the same system.

Installation problems: as discussed in the tables above things can and do go wrong when installing a new system. One example of this would be when they changed systems in Terminal 5 at Heathrow Airport (see the case study box). Some of the things that could go wrong include:

- Incompatible file formats: one example of this is Apple Pages and Microsoft Word, if a company uses these file formats, they do not talk to each other which renders files unreadable. To ensure this does not happen when changing a system standard file formats should be used whenever possible.
- Incompatible data structures: Data is stored in data structures when in use by the computer, if the system is trying to use a data structure it is not used to then it is unlikely to be able to read and write from the structure.
- Incomplete data transfer: Often when transferring a lot of data the data becomes lost. Whether this is technical error or human error it is often not found for a long time by which point the data is irretrievable.

- Different international conventions with formats, currencies and character sets: 05/06/2013 is either the 5th of June or 6th of May depending on where in the world you are from. If you try to import dates from a system that does not use the same system as you then it is likely that you will get errors in your data. The same as if the currencies are different or you have not encoded text in the same way.

Heathrow Terminal Five – Case Study

Heathrow Terminal Five was set to be the flagship terminal for Heathrow airport, British Airways were going to solely fly from this terminal. It was state of the art with new baggage systems and hailed as a triumph of airport design. However, on the day of opening 28th March 2008, it became apparent very quickly the baggage was not working and at least 28000 bags were lost in March and April. The cause of the failure? Not enough user testing, not enough time to check the installation process had worked and too many passengers on the first day. This was one of the most embarrassing events for BAA (airport owners) and British Airways. It was so flawed and so embarrassing they had to run national adverts to ensure people knew that Heathrow was safe to fly through once they had sorted out the problems.

Testing A System

As can be seen from the Heathrow case study it is essential to fully test a system. Although it is impossible to test every eventuality it is possible to thoroughly test systems to try and ensure system installation and transfers are as smooth as possible. If a system does not work as expected it can lead to the system being unreliable, companies could lose clients due to incorrect data, and the users of the system may lose trust in the system and become dissatisfied. The main types of testing are as follows.

Alpha Testing / Debugging: Alpha testing is the first stage of testing it is the stage that happens while programming. The programmers complete this stage of testing to check there are no syntax or logic errors in the code and that it is functionally correct. Some of this testing can be automated.

Beta Testing: Beta testing is when the software is released to a select group of users, they either use the program as intended or are given tasks to complete by the programming team. It is their job to report any errors to the developers. Many people do this for games companies (see case study).

User Acceptance Testing: This is the final stage of testing. Once the company has the software and it is in place they “sign off” on the project effectively saying that now it is in the company and has been rolled out to all employees they are happy with it and have adopted it as their system.

Beta Testing for a Games Company

Sounds like a dream job – getting paid to play games that have not yet been released to the public. However often developers want to find out if there are any hidden flaws in the game. Sometimes Beta testers for games have to get to a certain point in a game and then jump off a building 100 times, just to check the game environment functions properly. They also get paid a small amount for a lot of time and effort but at least they get one of the first copies of the game.

Data Migration

One of the reasons that testing is so important is because migrating data from one system to another can lead to problems. Testing can help to foresee and eliminate problems before they become a big issue.

Possible causes of data migration problems are detailed in the following table:

Issue	Description	Example
Incompatible file formats	When the files being transferred to the new system use a storage format that is unrecognized by the new system.	An example of this might be if you use pages and your teacher uses MS Word. Unless you save the file in a neutral format such as .rtf or .pdf your teacher cannot review your work.
Incompatible data structures	If the current system makes use of a data structure that is not used by the new system it would be difficult to save the files.	An example of this would be if you stored data in your program in an Array list but your friend used a linked list, these are not interchangeable one of you would have to reprogram part of your code.
Different validation rules	Different organizations / Countries have different rules about who can subscribe/buy items etc. These are usually built in validation rules, if you are moving the system to a new country these may have to be adapted.	An example of this would be buying alcohol. In the USA (most states) you have to be over 21 to buy alcohol, in the UK over 18 in Switzerland over 16 or 18 depending on the type. How could a company create a system that caters for all these rules?
Incomplete data transfer	When the data that is being moved to the new system does not completely transfer.	An example of this might be if you use cloud-based storage. If you start to upload from your machine but the connection drops and you attempt to access the data from another device it will not be there as expected.
Different data formats	Different countries use different standards for using dates. This should be considered when migrating data as dates may have different meanings.	An example of this is if your birthday is: 03/05/1996. In the UK (and most of Europe) this means you were born on the 3 rd of May 1996. In the USA it means you were born on the 5 th March 1996. Other birthdays such as 27/11/98 become invalid.

Different currency formats	Different countries use different formats for displaying currency. This includes the currency sign and the use of placeholders.	In the USA the \$ is used and. used to show decimal points. In Switzerland the format CHF is used and, to show the decimal point. This can lead to CHF3,99 becoming \$399 in the USA
Different Character Sets	In the past when programs have been coded they have tended to use the local character sets and to save space they have used to most economical way to store the data.	Internationalisation means that software has to be capable of storing all the characters in all languages. To enable this you might need to use more memory. If you do not rather than getting characters that make sense you will get symbols.

User Development

The processes described above are discussing the logistics of creating or adapting a system. In general how do the systems development team create and implement the system and the issues that arise as a result of this? However, many systems are designed to be used by people who are good at their job and not necessarily brilliant using computers. How are they prepared for the change? How can the development team best support them?

The first way in which the team can help with the change in the system and get people to use the system is through **user training**. User training as the name suggests is educating the employees of a company about how to use the system. It is essential that the management provides high quality, in depth and professional training otherwise the system implementation will be delayed or could fail. There are several ways to deliver training these are described below:

Training Method	Description	Disadvantages	Advantages
Self Instruction	Users are given training materials and work through this by themselves	The employee is not an expert and they have no one to ask if they are unsure The employee may be expected to learn and work at the same time	Cheap – the company only needs to buy the resources
One to One	An instructor sits with an employee explaining in detail the new system and how they can interact with it	Very expensive – the trainer may have to be employed for a long time if they are to sit with everyone in the company Slow – it will take longer to train people one by one	Waterfall method – one person in the group could be trained to “expert” level and then guide the rest of their team through training

Group Training	An instructor sits with a whole team explaining how to use the system and how they can interact with it	Expensive – the trainer has to be employed Loss of work time – groups of people have to be removed from the work place Slow – it could take a long time to train all groups	Many people will have had the same training and can help each other in the workplace when the system is in place
Online Training	Employees have access to classes and resources on line that they can follow. A tutor is available to help if required.	There is no face to face contact you have to wait for a tutor response Some people find online work distracting	The materials are available at all times to refer to.

User Documentation

User Documentation is also a very important aspect in helping people to adapt to the new system. High quality and helpful user documentation should be provided in order to help the staff with the new system. Nobody can remember everything they have been told, they need references until it becomes routine. Having the right documentation available to the people using the system can help the new system become second nature to people quickly, poor documentation can hinder this process.

User Documentation	Description	Example
Help Files	Files that are available within the program to be searched when the user has a problem	When you are unsure of what to do in a program so you use the help menu. It was standard to do this before the arrival of the Internet.
Online Support	Instructions, tutorials and forums available on the internet that help people when they have issues.	An example of this is the webpages associated with their programs. They contain FAQ's, tutorials, information about the product.
Printed Manuals	Books that contain information about the product, frequently asked questions and common commands in the program	These used to come with every software you bought however, now they are mainly associated with games.

System Backup

Once you have a system in place and it is integral to the company, a large volume of irreplaceable data will be produced and stored on a daily basis. It is important to the company that this data is not lost as losing data or data falling into the wrong hands can cause people to lose faith in the company.

There are many causes of data loss for each there are ways to prevent it happening. These are detailed below:

Natural Disasters: Natural disasters can happen, this includes flooding and fire. If the building in which the data is held is affected by one of these disasters then the data can be lost. One way to prevent this would be to have a full back up of the company's data once per week and then back up the changed files once per day. These files should be stored on removable devices that can be stored in a location away from the original data. Another way to prevent this happening would be to back up the data to cloud storage facilities. Failover systems could also be put into place, these are systems that are put into place when the primary systems fail. They are often used in critical systems.

Malicious Activity (Hacking, Viruses): Companies can be susceptible to hacking or virus attacks. Any company that stores information is a target for hacking. Networks that allow people to bring in memory stick devices or access the internet are susceptible to viruses that could spread quickly across the network. Setting up a firewall can help prevent hacking and anti virus software can be used to prevent the spread of viruses. Encrypting sensitive data can also help prevent information being useful to hackers which can help companies maintain trust if the worst should happen.

Intruders: It is possible that if an intruder enters a building then they can damage servers that store the data or they can attempt to steal the data. As well as having physical security such as locks and alarms on the buildings, all employees should be encouraged to have strong passwords so access to machines is not easily granted.

Employees of the Company (accidental): Sometimes employees can do something to a system that will make the system break. Having back up systems as described in the natural disasters section will prevent this from being more than an inconvenience to the company.

Employees of the Company (Malicious): Employees that are upset or angry at the company often try to sabotage data before they leave. This can be prevented by escorting staff from the premises if they have been dismissed. Data should also be stored by access rights, with only the highest level of staff having full access to all areas of the system.

Consequences of Data Loss

We have discussed in abstract the consequences of data loss but what happens to companies when it does happen. In May 2014 ebay.com was hacked and hackers stole information and encrypted passwords of 148 million users. What does this mean? Firstly people can have their identity cloned, people can purchase items using their account and linked payment data. Given that most people have one or two passwords, it would not take long for other accounts of the user to be hacked. The consequences for eBay were the lack of consumer confidence in their product and the cost of fixing the problem.

Keeping Software Up to Date

The story does not end once software has been deployed, technology is an ever-changing story. Devices that should work with the software change, data interacting with the software changes. Over the course of use bugs become apparent that may not have appeared at the testing phase. Companies that develop software are required to manage these products and they do this in the following way:

- Patches – small pieces of code that are sent out to users to fix problems that have occurred after deployment.

These can either be user driven updates or automatic. User driven updates are when the user receives a desktop notification or email and they must click on a link or button to apply the updates. Automatic updates happen when the operating system is connected to the Internet, pulls the information from the server and updates the relevant components at the next restart.

Patch Tuesdays

Patch Tuesday is the term given to the Tuesdays when Microsoft updates their products. On the second (and sometimes fourth) Tuesday of the month, Microsoft sends updates to its operating systems. The reason for this is to fix “glitches” that have been reported back to the development team or to fix security flaws that have been found in the system. Therefore it is always a good idea to restart a Microsoft device after these days.

Systems Design Basics

Computer systems have many different components. In this unit you need to be able to define the following:

Hardware – The parts of the computer you can see and touch for example: the monitor, the hard drive, the RAM

Software – the programs on the machine for example the operating system, the web browser, the word processing package

Peripheral - additional items that allow the computer to be functional (it used to be everything connected using a wire) for example the keyboard, the mouse, the printer

Network – computers and servers that are connected together in order to share information and resources.

Human Resources - The department in a company that is usually responsible for the hiring of people, the training of people and the administration associated with employees. They would be involved in helping people to do their job correctly.

There are many different roles a computer can perform in a networked world. These include the roles outlined below. We will look at these in more detail within the networks unit.

Client: A computer connected to the server, that gets information from the server.

Server: A computer that stores information and resources. Such as files, programs etc.

Email Server: An email server is responsible for receiving and distributing emails to people on that email server.

DNS Server: A Domain Name Server decodes the web address given and helps us to connect to the websites we require.

Router: A router gathers data packets and redirects them to the correct machine on the network.

Firewall: A firewall helps to protect the network machine against malicious attacks.

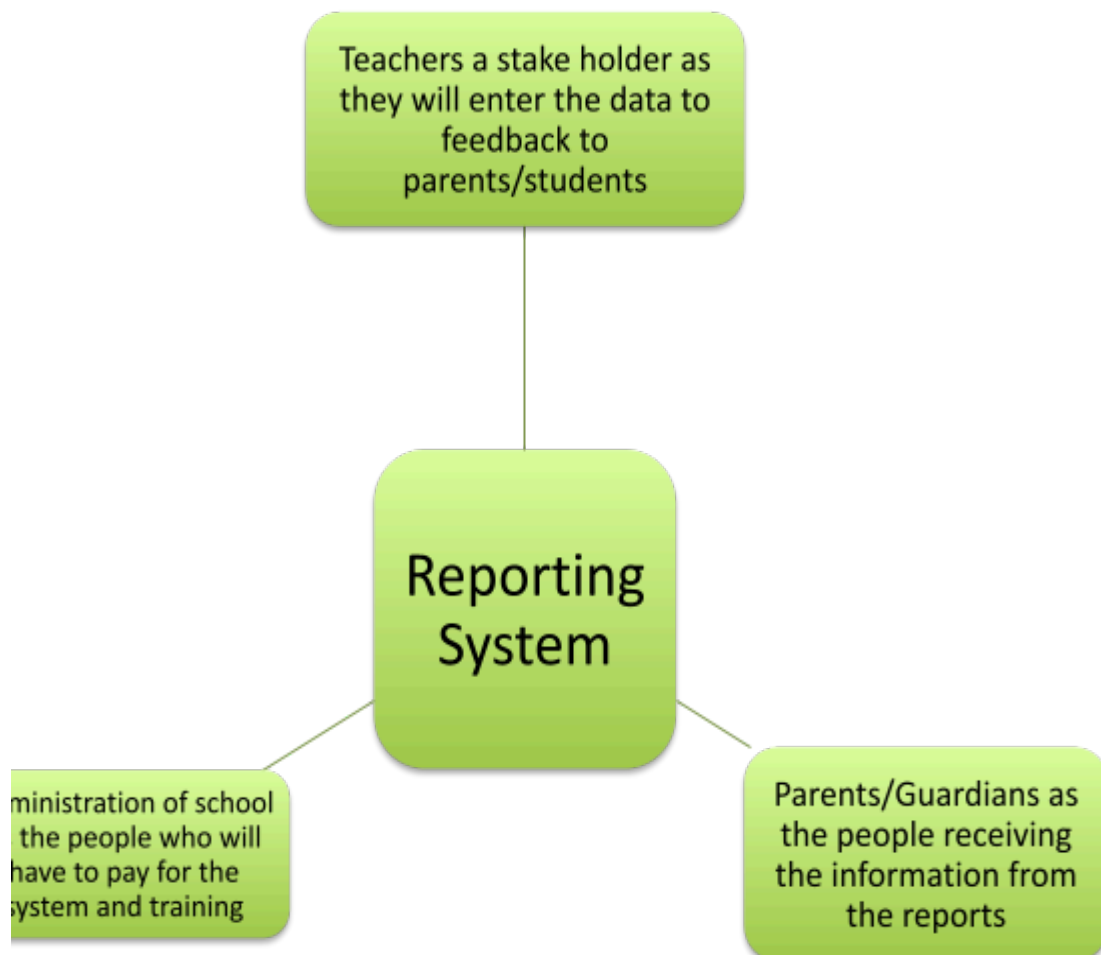
Area of Knowledge: Ethics

Now that most machines are networked it means that many people can access all information both within and out with the work environment. What are the social benefits and social drawbacks of this? How have working patterns changed as a result? Do you think education and your responsibilities as a student have changed as a result?

Things have changed as a result of networking people have different opinions about how and whether this change is good or not. You should consider the questions above as an aid to form your own conclusions

Systems Design and Analysis

Stakeholders are the people involved with the system. They are the people who will be affected by the change in the system. One of the first jobs that needs to be completed when starting the systems analysis of a system is to identify who the stakeholders are. An example could be choosing a new reporting system for the school:



However not every teacher, every parent and every member of administration could be asked for their opinion selective people are chosen and their opinions are used. In an ideal world we would be able to ask everyone but we can't.

Area of Knowledge: Ethics

Not being able to consult everyone leads us to one of the key TOK questions in Computer Science.

"What ethical limitations, if any, should be placed on research conducted in Computer Science?"

Should we be able to ask everyone what we like? Or is it ok to have limitations placed on our research?

Is it ever ethical to hack?

Once we have identified the stakeholders we enter the systems design process. Traditionally the systems design process went from one stage to the next meaning that no feedback was involved. This led to uninformed and poor end products as the end user was not involved. It is now accepted that we need to use **iteration**. The iterative cycle allows the feedback of the end user to order to inform each stage of our process, by constantly seeking feedback and using that feedback to revisit and reevaluate each stage we end up with a product tailored towards the end user.

The Systems Design Cycle:



Analysis Stage: Data Collection Methods

The first stage of the systems design process is analysis during the analysis stage we need to decide which people we are going to interview in order to get the information from them. There are several ways to do this each with their own advantages and disadvantages. The key skill that is required in order to gather the information is effective communication and collaboration. Stakeholders, especially employees, often find it difficult to express their true feelings in an environments when they think they are being judged or they think there will be consequences, one of the key jobs of the systems analysts is to encourage to talk even if they have these feelings. When analyzing a system a mix of the following methods will be used.

Method One: Interviews

The systems analyst meets with the stakeholders, individually or as part of a group.

Advantages:

- Can ask follow up questions.
- Can judge the feeling in the room.
- Can encourage responses.

Disadvantages:

- People often give what they think is the right answer rather than their answer.
- People fear being judged.
- It is time consuming.

Method Two: Surveys

The systems analyst conducts surveys in which people write down their opinion. This can include a mixture of open and closed questions.

Advantages:

- Less time consuming than interviews to conduct the research.
- More people can respond.
- Questions can be used to gain general understanding of the company.

Disadvantages:

- No follow up questions can be asked.
- Could be difficult to make conclusions from the information.

Method Three: Direct Observations

The systems analyst views the employees in the workplace to see what they do and how they do it. This allows them to form opinions about what might need to be changed.

Advantages:

- Observe people in the work place can see what they do.
- Can get a deeper understanding of how the process fits together.

Disadvantages:

- Time consuming.
- If you do not understand the business fully it is difficult to understand what is happening.

Once you have gathered the data from the stakeholders you then need to get information about the system and how it could be improved. This could be done using the following methods:

- Examining current systems.
- Examining competing products.
- Examine the hierarchy and organization of the company.
- Conduct literature searches.

One thing that the system analyst should be aware of when gathering information in this way about the system is that Intellectual property rights need to be observed.

What are Intellectual Property Rights?

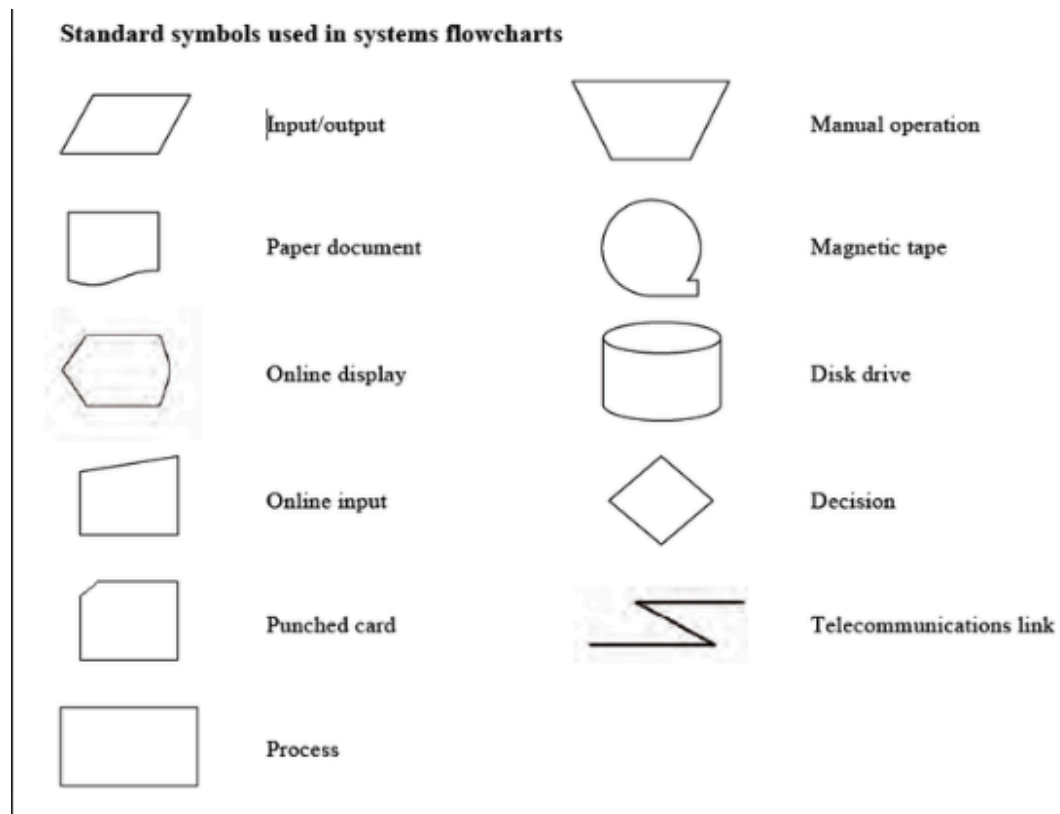
Intellectual property rights include trademarks, copyright, patents and industrial design rights. They cover the right of the person/company who made the content the rights to the use of the content and the right to stop others using their content without their express permission. It means that if people use your content without your permission then you have the legal right to seek compensation.

Design Stage

The next stage is the design process during this process the analyst team will produce several charts that are used to inform the development team how the system should function.

Systems flow chart

The first of these will be the systems flow chart. The systems flow chart shows how information flows through the system when a certain process occurs. The standard symbols are shown below:

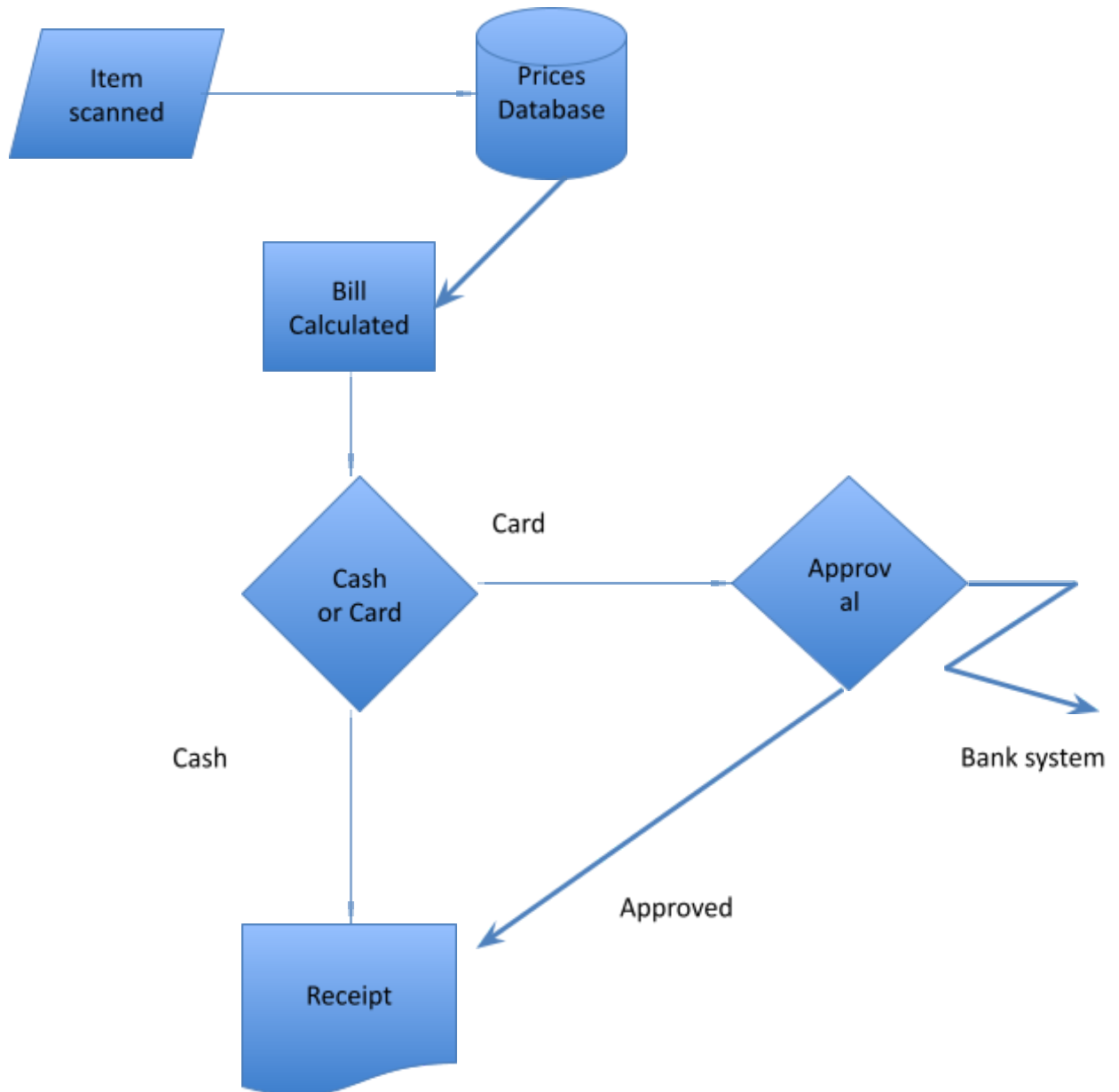


Source:
<https://sdd-hsc-online.wikispaces.com/Modelling+Tools+-+System+Flowcharts>

As you can see from the symbols involved the systems flow charts shows where the data is gathered, stored and processed when an action takes place.

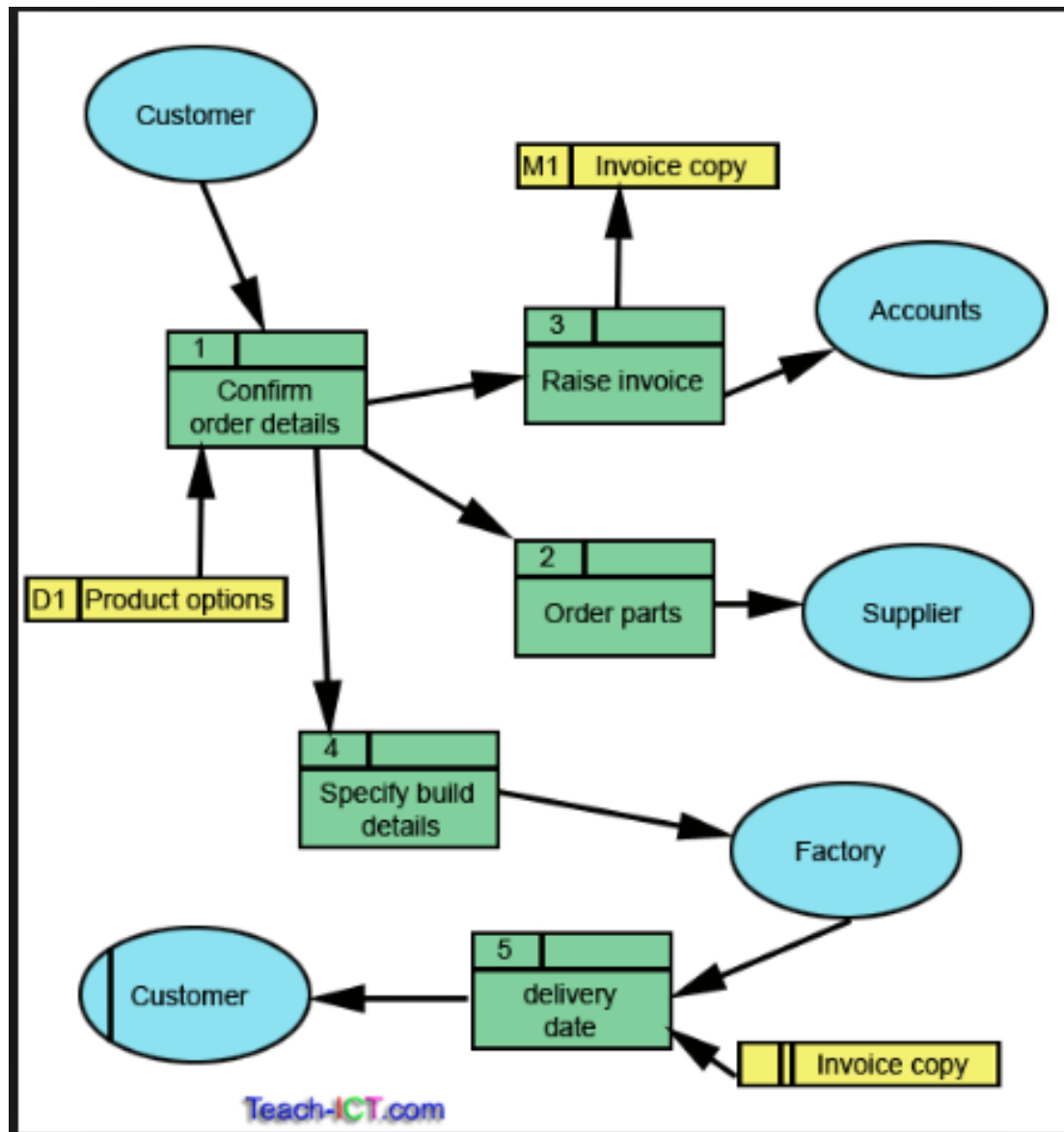
For Example:

A shop stores information on its database, each time an item is scanned the database is queried for the price and this is added to the bill. If the person pays by card the bank is contacted to approve the transaction. Once payment is received a receipt is printed and handed to the customer. Show this in a system flow diagram.



Data Flow Diagrams

Another diagram that may be produced is a data flow diagram. It is similar to the system flow diagram however rather than showing the physical items that need to be involved in the decision making it shows the objects, methods and outputs. An example below is taken from the teach-ict.com website:

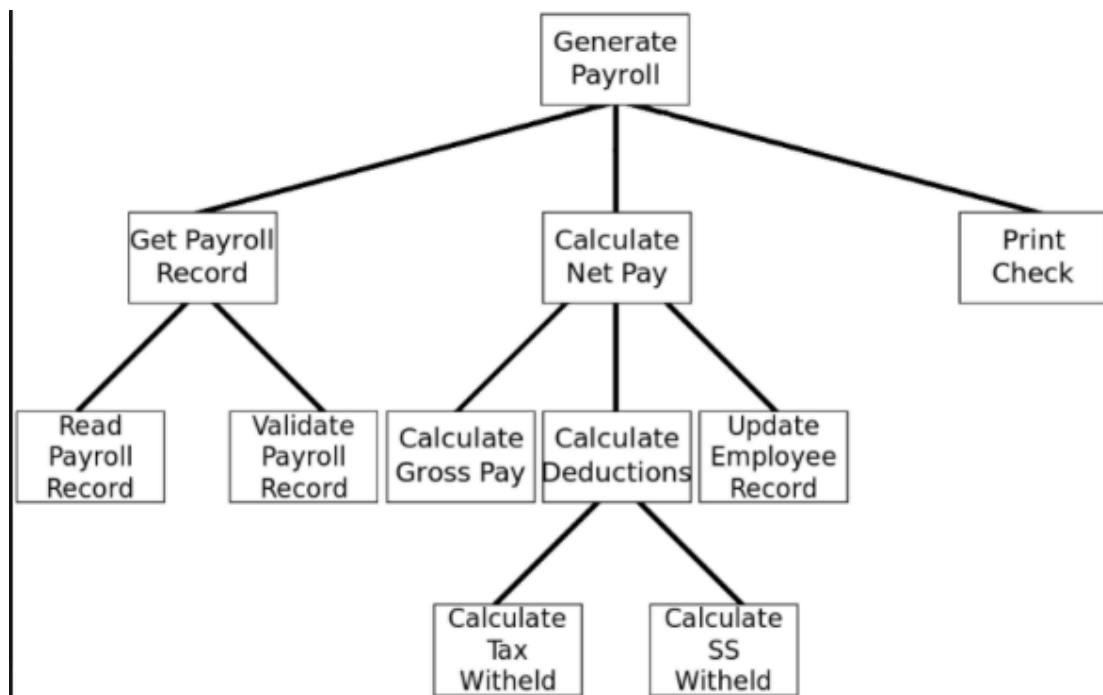


This diagram shows the objects involved in each transaction and then how they all fit together.

The customer confirms the details of their order. From this an instruction to order the parts is generated and the parts are ordered from the supplier. An invoice is raised using the accounts object and a copy of the invoice is printed. The build details are specified and built by the factory object. The delivery date is confirmed by the factory object and the object along with the invoice is sent to the customer.

Structured Charts

Another chart that can be used to show the operation in an organization is the structured chart. The structured chart allows us to take a big process and refine it into smaller more manageable processes. A common example of how this can be used is shown below. It makes use of generating an employees pay, this is a complex procedure as it involved deducting tax etc.



Once the programmers start to develop the system they will also use flowcharts and pseudocode to model the system, this will be covered in the computational thinking unit.

Prototypes

As part of the design phase the developers may produce a prototypes. Prototypes are mock ups of the system or very basic versions of the system that are given to the stakeholders to use and gather feedback allowing refinement of the system to happen before development fully begins. Prototypes can be used to show where the problems are and what does and does not work for the end users.

Examples of prototypes:



Evolution eines Prototypen

Source: iese.fraunhofer.de



Source: www.pcmag.com

The first picture shows the original design of the interface and then how it was created in a basic form on the computer. Users would then have been asked to interact with the product and provide feedback before it reached its final version as shown. Google glasses could be an example of a prototype, Google created a version of the glasses and you could register to buy a pair, if you bought the pair you had to participate in the development of the glasses by giving feedback to the product. Google quietly retired the project after receiving negative feedback. If the companies had not involved the end user they may have ended up with an disliked / unworkable product.

Area of Knowledge: Ethics

It is impossible to discuss the design and development of systems without thinking about the ethics behind the system. Is it ok to ask people to regularly change their working habits because a new system has been developed? Many people feel uncomfortable with change, is it ok to make people feel unhappy in their work on order to progress?

Also it is undeniable that technology makes processes more streamlined, reducing the need for as many workers. To what extent is making people redundant acceptable in the pursuit of a "better" product?

Human Interaction with the System

Accessibility

It is important that devices and systems are usable. System designers spend a lot of time and effort ensuring everyone can use their product however, there will be accessibility issues that need to be taken into consideration. Accessibility issues are the issues people may have when they are trying to use the system. The table below shows common devices, the usability/accessibility and the solutions to these.

Device	Issues	Solutions
PC's	The writing on the screen can be too small, not accessible to people with sight problems. The colours can make it difficult to read for some people	Change the contrast of the monitor to high contrast settings, Employ voice recognition and text to speech software Use a braille keyboard
Laptops	As above with the addition of some people find the fixed keyboard and track pad difficult	Have external keyboards and external mice that can be moved around. Use touch screen capabilities on surface style laptops
Mobile Phone	Many people find it difficult to use the small buttons on screen on a mobile phone.	Use the voice recognition software to help with instructions and composing messages etc.
Games Console	One problem people may find troublesome with games consoles is the use of the controller, it has many different buttons for different items	One way this could be solved is by changing the button configuration.

For any device you use it is possible to think of problems that may occur for some people. In most cases there is technology available that can compensate for the issues.

Usability

Usability is problems that may occur when people are using the system. Things that designers have thought of but which may still occur. These problems are system based not issues that the user may have.

In the following scenarios the usability issues are discussed as well as what can be done to overcome the issues.

Example One – Ticketing Systems

Ticketing systems allow you to buy tickets online. This could be for planes, concerts, the cinema etc. Usually when you buy tickets there is not too much of a rush therefore the ticketing system can handle the amount of enquiries. Sometimes however, there are very popular tickets that will sell out within minutes – it is possible that when selling the ticket is double sold leaving one person disappointed. How does the system cope with these?

One way in which the system copes with this is when you are allocated a ticket a “hold” is placed upon the ticket. The transaction must be completed within a set time otherwise the ticket is placed back into the pool. In theory the ticket could be selected at exactly the same time although this would be a very freak incident as the processing speed of computers is so fast.

Example Two– Voice Recognition Software

Voice recognition allows you to speak to a machine rather than enter the instructions in the traditional manner. This allows people with access issues to access the machine easily, it also means computer equipment can be used by a wider range of people from young children to older adults. But what happens when you have a strong accent? In the UK there are 57 recognised regional accents. How does the computer cope with the accent of someone from the Highlands and Islands of Scotland compared to a broad Liverpoolian accent that has many vowel changes?

The short answer is the system can not very well. Most software is trained to understand the accents of the target market however very strong regional or strong but non descript accents can often confuse the machine.

Area of Knowledge: Ethics

Computer systems contain many components, many made from finite rare materials. It might be worth considering the following question:

To what extent is it our moral responsibility to find sustainable sources of Technology, before developing more technology based on depleting resources?

As computers are being made more 'intelligent' and more intuitive for people to use especially younger children. Should programmers be responsible for making them safe?

Chapter One – Review Questions

Short Response Questions

- 1: Explain the difference between software as a service and traditional software.
- 2: State four kinds of servers that exist in a network world.
- 3: Give three examples of how computers can be adapted to be more usable.
- 4: Why is it important to include the end user in the analysis and design stage?
- 5: Explain why it is essential that the end user is involved in testing.
- 6: State 3 types of user testing.

Extended Response Questions

- 1: KCM telecommunications company are currently using an outdated system and have employed someone to create them a solution. Explain the options available to move to the new system? Which would you recommend for the company? Why?
- 2: It is argued that we have become too reliant on technology. What are the social issues associated with this?
- 3: Xcel properties is a company specializing in finding homes for expatriates. They have a lot of data on their system including names, addresses and salary details. Discuss the ways in which they could ensure they do not lose data due to a technical failure.
- 4: You are developing a new system for MSL enterprises. Discuss the ways in which you could test the system to make sure it functions correctly.
- 5: Why are updates required in a system? How do you release these updates to the users?

Chapter Two - Computer Organisation

Essential Understandings: “Binary is the route of all machines”

Key Questions:

How is data represented inside the machine?

How do computers perform the tasks they are given?

Computers are dumb! Aren't they?

How can binary be interpreted into the information we see on the screen?

ATL Skills:

Communication: Use and interpret a range of discipline specific terms and symbols;

Research: Understand and use technology systems.

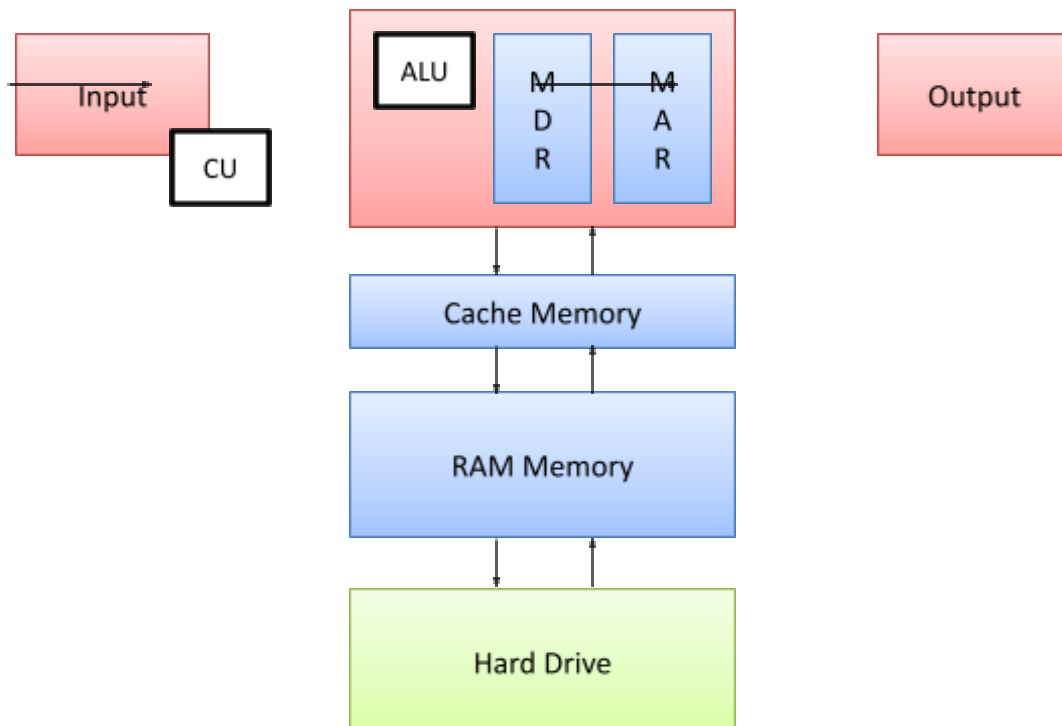
To

most people the inner workings of a computer system are magic, something that is not easily understandable and will always remain a mystery. To these people it might be surprising to learn that computers are actually dumb. All work computers complete have to be simple arithmetic or logic calculations. Eventually computers probably will be able to learn things for themselves and make their own decisions but at the moment they have just been cleverly programmed to look intelligent.

In this unit we will peek inside the box to have a look at what it does.

Computer Architecture

A computer system can be represented using the following diagram:



Computer systems work using input devices, processor, memory and output devices. In this unit we will look at each of the individual components separately and then look at how they work together.

Input – An input into a computer system usually comes from a device. Such as a mouse, a keyboard, a scanner, or a touch screen. The input from the device is processed by the processor before being sent to memory if required.

Inside the processor (also known as the central processing unit) there are four components you need to know about:

ALU – The arithmetic and logic unit, this unit performs all the calculations and logical comparisons for the machine.

CU – the control unit, the control unit is responsible for the overall running of the machine, ensuring instructions are decoded correctly.

MAR – Memory Address Register, this register stores the address of the next instruction of the program.

MDR – Memory Data Register, this register stores the data that is in current use.

Once the processing has occurred the output is displayed on screen if necessary. Otherwise the machine just continues onto the next instruction.

What role does memory play?

First of all there are two types of memory in a computer system. The first is **primary memory**. This is the registers, the cache and RAM. The data in this memory is lost when the computer is switched off and can some times per referred to as *volatile* memory. As explained above the **registers** are the smallest memory inside the computer system. They can only store one single piece of data at any given time. This is the data that is currently being used. As they are the smallest they are also the fastest memory in the machine. The **cache memory** is the memory closest to the processor, as it is closest to the processor and relatively small it is the next fastest memory in the machine. The cache memory stores the most commonly used instructions. This speeds up the computer. The final piece of primary memory in the system is **RAM** – random access memory. Random access memory stores the operating system and the programs/files currently in use. Once you turn off the machine everything in RAM that is not saved to secondary memory is lost. In the instruction cycle the data in RAM is taken to the register when needed. RAM is relatively quick to access.

If the operating system and programs are stored in RAM but RAM is lost when the computer is switched off you may be wondering how the computer ever switches on. There is a special section of primary memory called **ROM**. Read only memory (ROM) is as the name suggests permanent instructions stored on a chip. When you press the power button on your machine the ROM is activated, points the processor towards the start of the operating system and from there the operating system takes over the running of the machine.

How does the process actually work?

Machine Instruction Cycle:

The machine instruction cycle, is the cycle that happens thousands of times per second in your machine. It is required to carry out the instructions in the computer. It is useful to know this cycle as it explains how each section of the computer system is used.

1. Fetch the instruction

The next instruction is fetched from the memory address that is currently stored in the Program Counter (PC), and stored in the Instruction register (IR). At the end of the fetch operation, the PC points to the next instruction that will be read at the next cycle.

2. Decode the instruction

The decoder interprets the instruction. During this cycle the instruction inside the IR (instruction register) gets decoded.

3. Execute

The Control Unit of CPU passes the decoded information as a sequence of control signals to the relevant function units of the CPU to perform the actions required by the instruction such as reading values from registers, passing them to the ALU to perform mathematical or logic functions on them, and writing the result back to a register. If the ALU is involved, it sends a condition signal back to the CU.

4. Store result

The result generated by the operation is stored in the main memory, or sent to an output device. Based on the condition of any feedback from the ALU, Program Counter may be updated to a different address from which the next instruction will be fetched.

Secondary Memory

Secondary memory or backing storage it is where all the files and programs are stored when not in use. In the computer system this is the hard drive, however you also have removable backing storage in the form of SD cards, memory sticks and to some extent CD/DVD's.

Backing storage has too slow a read/write speed to be used in the same way as primary memory however it is required in a computer system to store the data when not in use. If we did not have backing storage the computer would have to be switched on all the time and much more susceptible to data loss. Magnetic tape in some cases is sometimes still used for backing storage, this is because it can store lots of data in a small space.

Random Access: Random access memory means that you can skip straight to the file you want. Therefore it is quick to access files.

Sequential Access: Sequential access memory means that you have to look through all the files in order before reaching the one you want. Files located at the beginning are quick to find but files located at the end would take a long time to reach. Sequential access is only really used for backing up data now as it is not useful to store data this way for everyday use.

It is useful for you to know roughly the size, speed and uses of the following secondary memory devices:

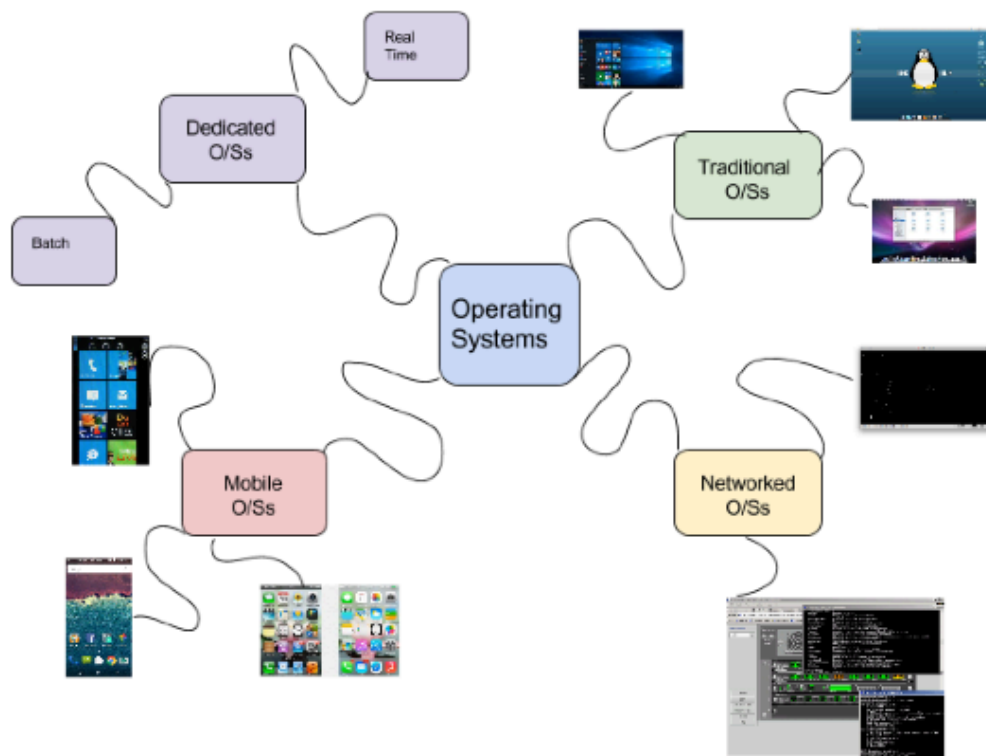
Device	Capacity	Read/Write time	Robustness	Portability	Access
Hard Drive 	Around 1 terabyte	Fastest	Reasonably	Not very as it is inside the machine	Random
SD Card 	Up to 128 Gb but usually 32/64 gb	Reasonably fast	Very robust, takes a lot to break them	Very portable can sit in a pocket and be lost	Random
DVD 	Up to 6.8 gb	Slow	Quite robust although prone to scratching or snapping	Very	Random
Memory Stick 	Up to 128gb	Reasonably fast	Very robust	Very portable	Random
Cloud Storage 	Can be unlimited depending on the service you pay for	Depends upon internet connection – so can be quite slow	Very robust, managed by big companies	Very portable as they can be accessed anywhere with data connection	Random
Magnetic Tape 	Around 200gb	Very slow	Very robust	Portable so they can be stored away from original machine	sequential

Computer Science – History

Before there were CD's which stored 650 mb of data, floppy disks were the normal form of removable backing storage. These held 1.44 mb of data which is roughly 1/3 of a song. In the late 80/90s one of the biggest selling computers was the Amiga 500 it had a massive 512Kb or RAM.

Operating Systems

As discussed throughout this unit, the operating system is essential to the running of the computer. It runs the machine and allows us to open the programs we want to without it being too difficult. Without the graphical user interfaces we are used to computer systems would be too complex for anyone but expert users to understand.



There are many different operating systems each serving different purposes. Mobile operating systems allow us to communicate and using apps on smartphones, network operating systems allow networks to be managed. Traditional operating systems allow us to use laptop and desktop computers, they usually allow us to connect to networks. Dedicated operating systems are operating systems that serve specific purposes for example. They are the systems that allow banks to carry out transaction in batch (processing payments etc.), they allow central heating systems to respond to sensors. Real time operating systems we have not covered as much, they are responsible for systems such as auto pilot, air traffic control, life support machines (systems where “crashing” could be catastrophic). In this unit we are mainly focused on the operating systems we find on laptops and desktops.

What is the job of an operating system?

Interface - The operating system provides a user interface allowing us to interact with the computer. There are three different types of interfaces:

- Command line interface
- menu driven interface
- graphical user interface

Command Line Interface:

Examples include: Terminal, MS-DOS

Advantages:

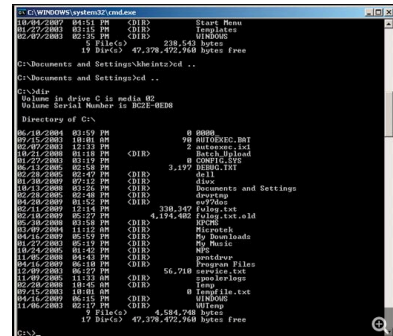
Experienced users can work very quickly without the use of menus.

No need for graphics or memory for the graphics.

Disadvantages:

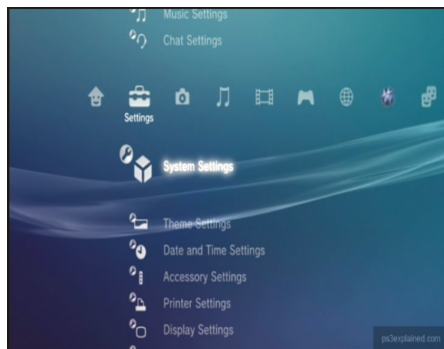
Have to type everything in correctly.

Have to learn many command prompts.



MS DoS interface
www.britannica.com

Menu Driven Interface



PS3 interface

Examples include: Digital Camera, PlayStation menu

Advantages

Easy to operate

Simple and impossible to mess up

Disadvantages

Slow to operate

Cannot skip steps and can only access a function through a specific route

Graphical user interface:

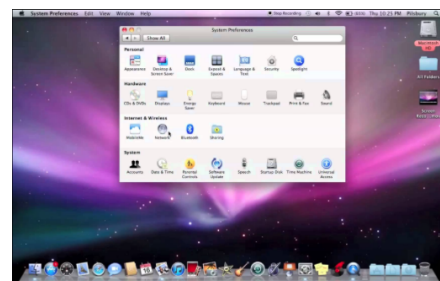
MAC OS and Windows

Advantages

- Good to use and simplifies for user

Disadvantages

You're limited if you want to go more in depth
takes up a lot of memory with the animation



Apple Graphical user interface

Graphical User Interfaces make frequent use of metaphors:

File icons -> within the computer files aren't really represented by respective icons

Desktop -> files aren't actually organised on a desk



One example metaphor I found was the “Trash Can” or “Bin” when we put the files in there we are not literally deleting them. We are marking them for deletion and the computer will deal with them later. In some cases we are only putting them in a different virtualisation of our file area until we tell the computer to ditch them.

Managing Time

The CPU as you know is instructions - the operating CPU time allowing all completed. (this will be resource management unit).

Process Name	% CPU	CPU Time	Threads	Idle	Wake Ups	PID
Adobe Acrobat Updater	8.5	2:17:41.66	9	2	28582	
WindowServer	4.2	1:06:04.14	5	24	145	
launchservicesd	2.8	55:12.07	6	1	77	
hid	2.5	22:45.86	6	0	97	
kernel_task	2.1	1:22:50.43	98	162	0	
mids_stores	1.4	2:43.16	6	0	170	
Spotify	1.4	9:50.78	35	12	41847	
Activity Monitor	1.3	54.38	8	2	37110	
coreaudiod	1.2	13:21.29	4	62	268	
Microsoft Word	1.2	1:02:53.76	17	51	259	
SMARTBoardService	0.8	20:26.40	7	1	390	
sysmond	0.7	8:11.90	6	0	181	
SMART Ink	0.6	10:01.78	13	18	372	
SMARTNotification	0.5	9:51.64	9	0	375	

required to execute system manages the operations to be explored more in the

Multi Task

The operating system allows us to have many programs open at the same time and to be able to perform different operations at the same time. For example you might be working on a report, listening to Spotify, have your email open etc. The operating system divides the processor time between the programs in order to make it look like the programs are running simultaneously.

Manages Memory

The operating system manages the memory of the computer system, it transfers data to and from backing storage and controls the primary memory.

Using your knowledge of the O/S you run list some functions that you think may be used to manage memory in the computer system:

Manages Peripherals

A peripheral device is generally defined as any auxiliary device such as a computer mouse or keyboard that connects to and works with the computer in some way.

Mouse

Keyboard

Headphones

Keyboard

Microphone

Printer

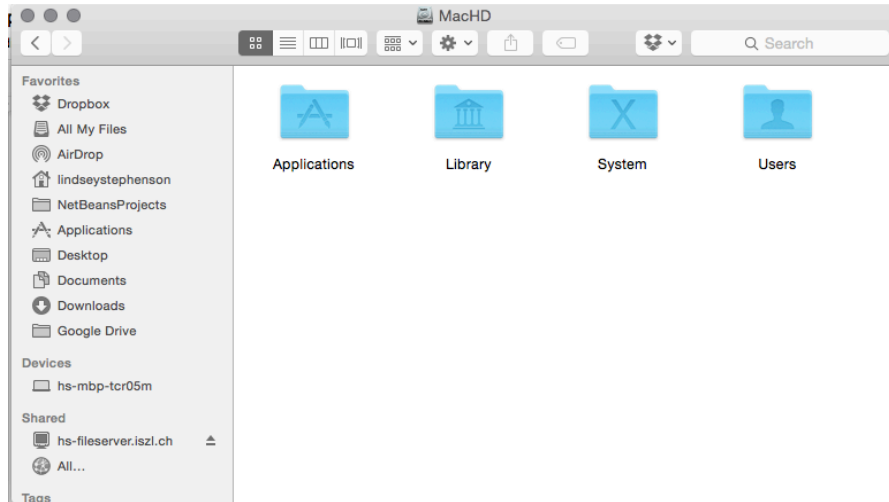
Peripherals need to be managed as they operate at different speeds, the operating system needs to account for the difference between the output speed or input speed of the peripheral and the CPU operating speed. It does this by providing a buffer.

For example sending data to a printer.

- 1: The user presses the print button
- 2: The operating system opens the communication to the printer and checks the protocols
- 3: The operating system then sets up a buffer for the CPU to write the data to
- 4: the printer then reads the data from the buffer

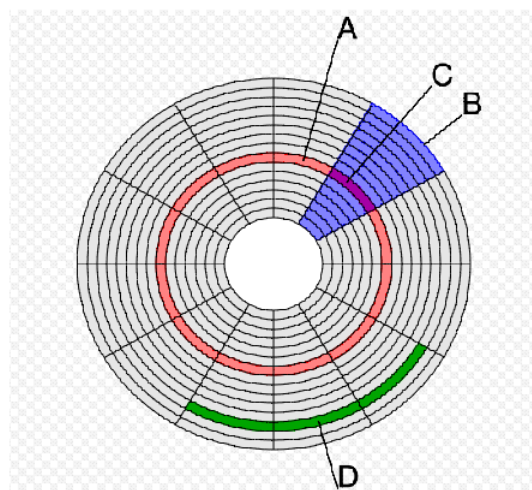
if we did not do this then when we sent information to the printer it would render our computers useless for a while as the CPU would have to wait of the printer to receive the information fully before moving on to the next task and printers read data much slower than CPU's.

Organises Memory



The above graphic shows how we the user see files, organized in to folders easy to find. However the graphic below shows how files are actually stored. They are separated across the hard drive, into separate sectors and tracks, large files stored over many sectors. The operating system manages where the files are saved, and where to retrieve them from if we want them displayed on our screen

- A : Track**
- B : Geometrical Sector**
- C: Track Sector**
- D: Cluster**



Security

The operating system also provides security for the system.

- User accounts allow us to have different users set up on the machine, this operating system keeps each users file separate from the other
- User privileges, the operating system allows different privileges to be set for different users, this means that executable files can be limited to only trusted users, helping to keep the machines safe.
- Anti virus software is provided by the operating system to prevent viruses access the machine
- Firewalls are provided by the operating system to prevent unauthorized access to the machine

Utilities

The operating system also provides many utilities that help the computer to function correctly. For example:

Defragmentation software - software that takes the sectors of the hard disk that are empty as the file has been deleted and moves them together. This limits the read/write time and means bigger files can be stored consecutively.

Application Software

The Operating System is known as System software, it does not do any tasks for us. If we only had an operating system our computer would not be that useful. That is why we need application software. The following table shows the different application software available to the user and the jobs it can carry out.

Application	Example program	Purpose of the software	Specialist Tasks
Word Processing	Pages, MS Word	To allow people to create printed documents such as report, letters, essays etc.	Mail Merge, indexing, producing references
Spreadsheet	Excel	To allow people to create accounts, analyse data, produce graphical information from data	Graphing, formulas, what if analysis
Database	SQL Server, MS Access	Store data in an organized and logical manner which can be used to produce information	Produce reports from data, store information, query data

Desk top publishing	In Design	To produce magazines, books, professional promotion materials	Combining text and data in a non standard way
Web authoring	Dreamweaver	To allow people to build websites	Combine text, graphics, multimedia in html format.
Email software	Outlook Gmail	To communicate between machines.	Encode and decode emails (with attachments) for sending across the network
Film Editing Software	IMovie Final Cut	To create films	The ability to manipulate sounds and images and add effects to make movies
Sound editing software	Garageband Audacity	To create songs	The ability to record into the software, or add recordings to the software in order to manipulate them into songs
Photo Editing software	Photoshop, Lightroom	To create professional looking images	To be able to add effects to photographs to make them enhanced.

There are many similarities in the menu structures across software packages. This is to help people learn the software quickly and become competent within a reasonable timescale. There are many common metaphors used across platforms such as scissors for cut, a floppy disk for save. By having these common metaphors, people feel less alienated by new software and can adapt to the software quickly.

Binary Representation

All computers use binary to represent data, the computer is filled with millions of transistors that are either on (1) or off (0). Depending on the state of the transistors depends on the data that is processed. All data must be represented as 1 and 0, nothing in a computer system is represented in any other form.

Computer Science – History

Computers used to be very simple devices they performed calculations depending on how vacuum tubes were set. Using programs where the user could write a program, print out punch cards and manually feed the punch cards into their machines in order to run their program. Following on from this we had procedural programming where computers had to follow steps in order. Now we have object orientated multiple input programs. These programs required much more memory and more sophisticated data representation.

Expressions of Size

The smallest information that can be stored in a computer is a bit a bit is a 1 or 0, the next measurement is a byte which is 8 bits.

Bit	1 or 0
Byte	8 bits
Kilobyte	1024 bytes
Megabyte	1024 kilobytes
Gigabyte	1024 megabytes
Terabyte	1024 gigabytes
Petabyte	1024 terabytes

RAM is measured in gigabytes (usually). Hard drives in gigabytes, although terabytes are becoming more common. As technology has progressed the size required to store data has decreased. What once required a whole room of storage now only required a postage stamp sized device.

Computer Science – History

The 2014 MacBook pro had 8Gb RAM and 512Gb memory

The IBM Mainframe (which sent man to the moon) has 8 – 64 kb RAM and 1mb memory (not even enough space to store a word document containing a picture)

Binary Conversion

One thing you need to be able to do on this course is to convert binary numbers into decimal and vice versa. To do this it is important to think about how numbers work.

Decimal - the decimal/denary number system is based on 10 units (0 - 9), once we pass the number nine we have ten + the number of units. In primary school you may have learned numbers as hundred, tens and units.

If we have a closer look at the number 8495 this becomes more clear. We know that this is eight thousand four hundred and ninety 5 but how? it is all to do with place holders.

Each space in our number represents the number of each value that needs to be added for example:

$10^3 = 1000$	$10^2 = 100$	$10^1 = 10$	$10^0 = 1$
8	4	9	5
this means we have $8 * 1000$	we need to add $4 * 100$	we need to add $9 * 10$	we need to add 5 units

We know that in a decimal number system once we get to 9 we need to add one in the next column. The same principle works for binary. Binary is base 2 therefore it can be a 1 or 0 if we have 1 and need to add one we move along the number line.

$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
1	0	1	1
this means we have $1 * 8 +$	we need to add $0 * 4$	we need to add $1 * 2$	we need to add 1
This binary number 1011 is therefore the same as the decimal			11

Using this principle we can work out any number in any base as long as we have the correct number line

Binary number line

2048	1024	512	256	128	64	32	16	8	4	2	0
0	0	1	1	0	1	0	0	1	1	1	1
$= 512 + 256 + 64 + 8 + 4 + 2 + 1 = 847$											
1	0	0	1	0	0	0	0	0	0	0	1
$= 2048 + 256 + 1 = 2305$											
0	0	0	1	1	1	1	1	1	1	1	1

$$= 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 511$$

Conversion from Denary (Decimal) to Binary

- 1: First have a copy of the Binary number line
- 2: Start with the largest number that will subtract from the target number without a negative result
- 3: subtract place a one in the column and move the remainder along the number line

For example the number 147

2048	1024	512	256	128	64	32	16	8	4	2	0
				147 - 128	19 - 64	19 - 32	19 - 16	3 - 8	3 - 4	3 - 2	1 - 0
				1	0	0	1	0	0	1	1

check $128 + 16 + 2 + 1 = 147$

Another example the number 1450

2048	1024	512	256	128	64	32	16	8	4	2	0
	1450 - 1024	426 - 512	426 - 256	170 - 128	42 - 64	42 - 32	10 - 16	10 - 8	2 - 4	2 - 2	0 - 0
	1	0	1	1	0	1	0	1	0	1	0

check $1024 + 256 + 128 + 32 + 8 + 2 = 1450$

Binary Addition

In order to understand the importance of variable types it is necessary to understand binary addition.

for example if I wanted to add the following

00001001
00001111
= 00011000

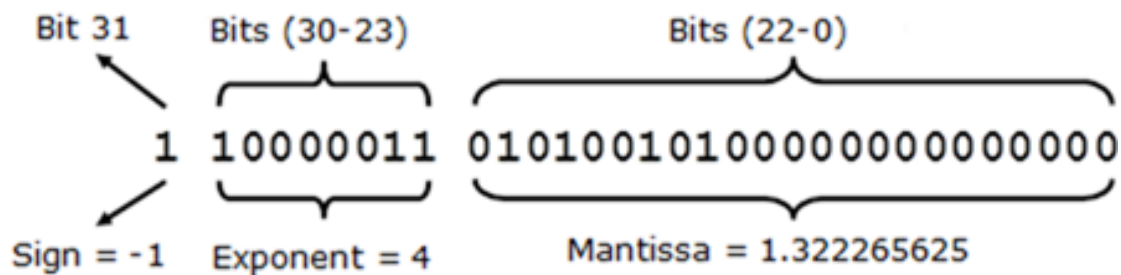
1 + 1 = 10
1 + 1 + 1 = 11

This works however if we run out of bits to move along to then an overflow error occurs.

This means that the number of bits allocated to the variable has been exceeded and an error has happened. Therefore in Java programming it is important to use the correct variable remembering int has an upper limit of 2,147,483,647.

Decimal Numbers

Decimal numbers are stored in a different way. It is similar to scientific notation. We store the number without the decimal point (mantissa) and then the number showing where the decimal point should be (exponent).



What is the purpose of hexadecimal?

With all the talk of colours and storing colours it is important that there are many options available to people producing digital images. It would be difficult for users to remember binary numbers for colours this is why hexadecimal has been used as a kind of shorthand to remember binary, instead of writing down the 16 bit colour code you break each 16 bit colour code into 4 bit chunks and use the hexadecimal short code.

Bin	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

To convert 0110111000001000 separate 0110 1110 0000 1000 and add the correct hex 6E08

The same method is used to remember any binary numbers. Hexadecimal numbers are also used to refer to memory locations on the machine.

File Extensions

Remembering that all data is stored as binary file types are important, the file type tells the computer what encoding has been used so that the computer knows how to read it. For example 00000000 01000001 could be the letter A (in Unicode), it could be the number 65, it could be a sound in an MP3 file or a colour in a picture. It depends on how the computer needs to decode the data. Common data types are as follows:

File Type	Extensions
Text	.txt .docx .ascii .unicode
Music	.mp3 .wav . aac .ps4
Pictures	.jpeg .gif .bmp .png .koalaglausen
Video	.mp4 .mov .wmv .YouTube

Text

In the past ASCII was used to store text. ASCII is the American standard code for information interchange. It is a list of

USASCII code chart

Row\Col				0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0	0	0	0	0
63	0	0	0	0	0	0	0	0	0	0	0
64	0	0	0	0	0	0	0	0	0	0	0
65	0	0	0	0	0	0	0	0	0	0	0
66	0	0	0	0	0	0	0	0	0	0	0
67	0	0	0	0	0	0	0	0	0	0	0
68	0	0	0	0	0	0	0	0	0	0	0
69	0	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0	0
71	0	0	0	0	0	0	0	0	0	0	0
72	0	0	0	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0	0	0	0	0
74	0	0	0	0	0	0	0	0	0	0	0
75	0	0	0	0	0	0	0	0	0	0	0
76	0	0	0	0	0	0	0	0	0	0	0
77	0	0	0	0	0	0	0	0	0	0	0
78	0	0	0	0	0	0	0	0	0	0	0
79	0	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	0
81	0	0	0	0	0	0	0	0	0	0	0
82	0	0	0	0	0	0	0	0	0	0	0
83	0	0	0	0	0	0	0	0	0	0	0
84	0	0	0	0	0	0	0	0	0	0	0
85	0	0	0	0	0	0	0	0	0	0	0
86	0	0	0	0	0	0	0	0	0	0	0
87	0	0	0	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	0	0	0	0	0
89	0	0	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0	0	0
91	0	0	0	0	0	0	0	0	0	0	0
92	0	0	0	0	0	0	0	0	0	0	0
93	0	0	0	0	0	0	0	0	0	0	0
94	0	0	0	0	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0	0	0	0
97	0	0	0	0	0	0	0	0	0	0	0
98	0	0	0	0	0	0	0	0	0	0	0
99	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0	0	0	0
102	0	0	0	0	0	0	0	0	0	0	0
103	0	0	0	0	0	0	0	0	0	0	0
104	0	0	0	0	0	0	0	0	0	0	0
105	0	0	0	0	0	0	0	0	0	0	0
106	0	0	0	0	0	0	0	0	0	0	0
107	0	0	0	0	0	0	0	0	0	0	0
108	0	0	0	0	0	0	0	0	0	0	0
109	0	0	0	0	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0	0	0	0
112	0	0	0	0	0	0	0	0	0	0	0
113	0	0	0	0	0	0	0	0	0	0	0
114	0	0	0	0	0	0	0	0	0	0	0
115	0	0	0	0	0	0	0	0	0	0	0
116	0	0	0	0	0	0	0	0	0	0	0
117	0	0	0	0	0	0	0	0	0	0	0
118	0	0	0	0	0	0	0	0	0	0	0
119	0	0	0	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0	0	0	0
121	0	0	0	0	0	0	0	0	0	0	0
122	0	0	0	0	0	0	0	0	0	0	0
123	0	0	0	0	0	0	0	0	0	0	0
124	0	0	0	0	0	0	0	0	0	0	0
125	0	0	0	0	0	0	0	0	0	0	0
126	0	0	0	0	0	0	0	0	0	0	0
127	0	0	0	0	0	0	0	0	0	0	0
128	0	0	0	0	0	0	0	0	0	0	0
129	0	0	0	0	0	0	0	0	0	0	0
130	0	0	0	0	0	0	0	0	0	0	0
131	0	0	0	0	0	0	0	0	0	0	0
132	0	0	0	0	0	0	0	0	0	0	0
133	0	0	0	0	0	0	0	0	0	0	0
134	0	0	0	0	0	0	0	0	0	0	0
135	0	0	0	0	0	0	0	0	0	0	0
136	0	0	0	0	0	0	0	0	0	0	0
137	0	0	0	0	0	0	0	0	0	0	0
138	0	0	0	0	0	0	0	0	0	0	0
139	0	0	0	0	0	0	0	0	0	0	0
140	0	0	0	0	0	0	0	0	0	0	0
141	0	0	0	0	0	0	0	0	0	0	0
142	0	0	0	0	0	0	0	0	0	0	0
143	0	0	0	0	0	0	0	0	0	0	0
144	0	0	0	0	0	0	0	0	0	0	0
145	0	0	0	0	0	0	0	0	0	0	0
146	0	0	0	0	0	0	0	0	0	0	0
147	0	0	0	0	0	0	0	0	0	0	0
148	0	0	0	0	0	0	0	0	0	0	0
149	0	0	0	0	0	0	0	0	0	0	0
150	0	0	0	0	0	0	0	0	0	0	0
151	0	0	0	0	0	0	0	0	0	0	0
152	0	0	0	0	0	0	0	0	0	0	0
153	0	0	0	0	0	0	0	0	0	0	0
154	0	0	0	0	0	0	0	0	0	0	0
155	0	0	0	0	0	0	0	0	0	0	0
156	0	0	0	0	0	0	0	0	0	0	0
157	0	0	0	0	0	0	0	0	0	0	0
158	0	0	0	0	0	0	0	0	0	0	0
159	0										

binary numbers and the character they represent on the keyboard. ASCII uses 7 bits, therefore there are only (2^7) or 128 characters that are available. What are the problems with this? iCloud

128 characters is not enough to represent anything that is not the Latin alphabet. Internationalisation is a problem without more than 128 characters, therefore global use of software would be a problem.

To overcome this problem Unicode was used. Unicode uses 16 bit to 32 to store characters. This allows 65 536 - 4294967296 available characters. Allowing all possible (known) alphabets.

Render Mode:	Standards compliance mode
Encoding:	UTF-8
Size:	20.93 KB (21,433 bytes)
Referring URL:	http://www.bbc.co.uk/education/guides/zp73wmn/revision/5
Modified:	12 Jan 2016 15:18:03

To find out the encoding being used on the browser you are using check view > page info > encoding

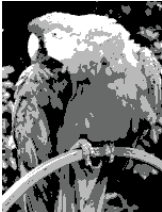
If you were creating a program that would only be used in one specific language then ASCII encoding would be best as it saves space. However, if you are unsure of where the product would be used Unicode should be used.

Pictures/Graphics

A bitmap is like the name suggests a map of the picture showing where each colour should go. The picture is made up of pixels and the colour information for each pixel that is then reproduced by the computer onto the screen.



The image on the left is the original image and on the right the zoomed image. Each pixel requires 24bits to store the colour information.
The number of bits required to store the picture is known as "colour or bit depth" the number of bits available for colour information the less bits available the less colours available for use:

Number of Bits	Number of Colours	Example
1	2 (2^0)	
2	4 (2^2)	
8	256 (2^8)	
16	65536 (2^{16})	
24	16777216 (2^{24})	

It is pointless storing colour information in more than 24bits as the average human can only distinguish between 16 million colours.

Calculating the file size of a bitmap (uncompressed):



the picture is 600dpi
it is 12 inches by 16 inches
24 bit colour depth

1: get the area of the picture ($12 * 16$) = 192
2: multiply by dpi, get the total pixels ($192 * 300$) = 57600
3: multiply by colour depth ($57600 * 24$) = 1382400 bits
/ 8 = 172800 bytes
/ 1024 = 168.75 Kb

Calculating memory requirements



You need to be able to calculate the storage requirements of any given file. For example if a picture is 300 dpi and it is 5.5 * 4.5 with a 16 bit colour depth what size will the (uncompressed) file be.

- 1: get the area of the picture: $(5.5 * 4.5) = 24.75$
- 2: multiply by resolution (300) = 37 125
- 3: next multiply by the number of bits in each pixel (16) = 594 000 bits
- 4: divide until appropriate unit.
 $594\ 000 \text{ bits} / 8 = 74\ 250 \text{ bytes}$
 $74\ 250 / 1024 = 73 \text{ Kb}$
 (always divide is answer is still over 1024)

A **vector graphic** is different, it stores the information in layers. Each layer stores the coordinates of where it should be on the page and reproduces itself on the screen. As this coordinates are stored relative to the size of the page, the image can be scaled up and down easily without loss of detail.

Metadata

Metadata is data about the data object for example when you take a photograph with your camera on your mobile phone it stores additional information such as the place the photo was taken, the data and time the photograph was taken and where it is stored on the phone. This metadata is stored alongside the photograph to help with searching, and to help applications track where the photos come from.

Compression

Compression of files is required so they can be sent across networks easier. It also allows more to be stored in memory as uncompressed pictures, videos and audio files use a lot of space.

There are two types of compression:

Lossless compression – compression techniques that compress the data without losing any of the original information.

Lossy compression – compression techniques that compress data but lose original information.

Techniques:

To decrease the size of music files you can do the following:

- Reduce the sample rate (the number of times the sound is produced per second)
- You can exploit the limitations of human hearing and get rid of the sounds humans won't hear
- To decrease the size of movie files you can:
 - Reduce the number of frames per second
 - You can reduce the size of the video window

All of the above compression techniques are lossy.

TOK Link

To what extent are the ways in which meaning is encoded in computer languages similar to the ways in which humans convey knowledge?

If we don't understand the language used to store our data how do we know what we see is a true representation?

Do we place too much faith in the electrical pulses?

Boolean Logic

Boolean logic is used extensively in programming, we need to complete comparisons for IF statements and to end conditional loops and as well as being able to write Boolean expressions in Java you need to be able to display Boolean expressions using Boolean logic and circuit diagrams.

Boolean Operators

There are several Boolean operators you need to know about.

AND	NAND
OR	NOR
NOT	XOR

The operators for these are as follows:

Operator	Symbol	Explanation
AND	$A \cdot B$	Both inputs must be true for the outcome to be true
OR	$A \oplus B$	One of the inputs or both of the inputs have to be true for the output to be true
NOT	$\neg A$	If you have a false input the output will be true
NAND	$\overline{A \cdot B}$	This is the inverse of AND, if both outputs are true the output is false
NOR	$\overline{A \oplus B}$	This is the opposite of OR, if both inputs are true or one of the inputs is true the output is false.
XOR	$A \oplus B$	This is an exclusive or meaning that is one of the outputs is true the output is true. However if both inputs are true the output is false.

Using Boolean Expressions

“If it is warm and **not** raining **then** we will go for a picnic”

This statement can be represented using the Boolean statement. When A is the temperature and B is the rain.

$$\neg A \cdot B$$

It has to be warm (input A) AND NOT raining (input B) for conditions to be right for a picnic

Another example would be “if there is no seatbelt of the door is open sound an alarm”, can be represented as the following where A is the seatbelt and B is the door.

$$\neg A + B$$

Truth Tables

Truth tables allow us to investigate all possibilities of a Boolean statement, to decide whether the output would be true or false.

Operator Truth Tables

AND

A	B	OUTCOME
0	0	0
1	0	0
0	1	0
1	1	1

OR

A	B	OUTCOME
0	0	0
1	0	1
0	1	1
1	1	1

NOT

A	OUTCOME	
0	1	
1	0	

NAND

A	B	OUTCOME
0	0	1
1	0	1
0	1	1
1	1	0

NOR

A	B	OUTCOME
0	0	1
1	0	0
0	1	0
1	1	0

XOR

A	B	OUTCOME
0	0	0
0	1	1
1	0	1
1	1	0

Important note – If there is a line over the operator it is negated

Using the same examples as above, the truth tables show when the output will be true.

“If it is warm and **not** raining **then** we will go for a picnic”

A	B	Out
0	0	0
0	1	0
1	0	1
1	1	0

Boolean Statement:

Truth Table:

$$\overline{A \cdot B}$$

“if there is no seatbelt of the door is open sound an alarm”

A	B	Out
0	0	1
0	1	0

1	0	1
1	1	1

Boolean Statement:

Truth Table:

$$\overline{A} + B$$

You can also use truth tables to determine if two Boolean expressions are equal or not.

$$\begin{array}{l} A \cdot \overline{B} \\ \overline{A} + \overline{B} \cdot A \end{array}$$

A and Not B		
A	B	Out
0	0	0
0	1	0
1	0	1
1	1	0

Not A or Not B and A		
A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

The truth table shows that these two expressions are not equal.

Circuit Diagrams

It is also important that you can represent Boolean logic statements as circuit diagrams.

Circuit diagrams consist of inputs, logic gate and outputs. For this course you have to be able to:

Draw a logic diagram for a given expression

State the expression that is being represented by the diagram

Draw a truth table for the logic circuit shown

The symbols used to represent each of the operators is shown below:

Operator	Symbol (algebra)	Picture	Function
AND	$\cdot$		If both are true put it through

OR	+		If 1 or both is true put it through
NOT	– A		False input = True output
XOR	+ inside °		1 but not both
NAND	– °		Inverse of AND
NOR	– +		Inverse of OR

Chapter Two – Review Questions

1: One line in a program reads

$$\text{salesTax} = a - b + 0.1$$

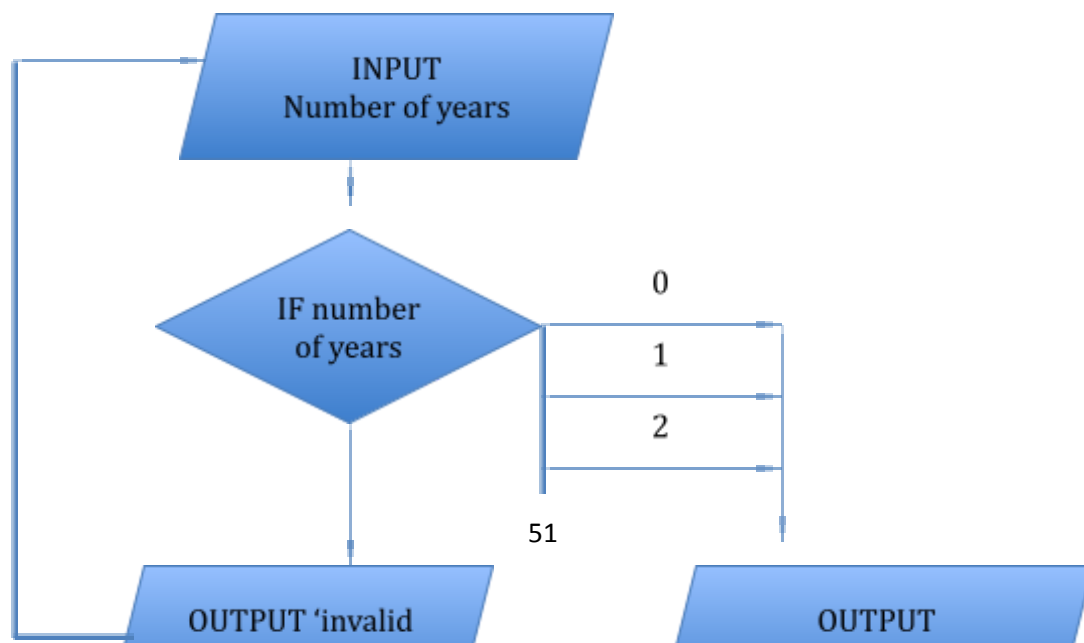
Work out the value of this expression when a is 600 and b is 200.

The programmer has made an error in the code. When a is 600 and b is 200 the expected answer is 40.

Rewrite the line of code with added parenthesis to give the expected result

2: Ahmed is writing a program to record the data of members of the school football squad. The input data will be validated. One input is the number of years a member has played for the team. This will be 0, 1 or 2.

The flow chart for the validation of number of years is shown below:



What is the output when the input is 2?

What is the output when the input is 3?

In a high level language, write the code that will produce the validation process shown in the flowchart.

The squad has 30 members. Ahmed stores the records in an array called Squad. To calculate how many females there are he designs this Pseudocode.

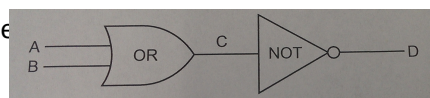
```
NoOfFemales = 0
Index = 1
While Index < 30
    If Squad[Index].sex = "f"
        Then
            NoOfFemales = NoOfFemales + 1
        End if
        Index = index + 1
    ENDWHILE
```

What is the problem with this code?

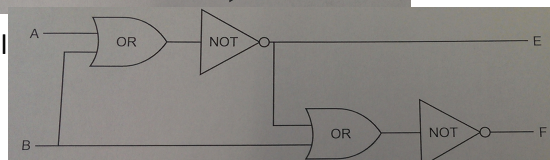
Write the updated Pseudocode to solve this problem.

3: The operating system controlling the student's home computer is a single-user, multi-tasking operating system.

i) Describe a single user



4: Complete a truth table for this circuit.



a)

b)

c) Construct a *logic diagram* for the Boolean expression

A and B OR NOT B

5: State **one** example of application software

6: State **one** function of the *operating system* in managing memory

7: Amalia Rogriguez is a student and is completing her homework. This requires her to view webpages, edit documents and print out a draft copy.

However, she is also surfing the web, keeping up to date on her social networking sites as well as downloading apps and music from a P2P site.

a) For one of the above applications, outline **one** task that is carried out by the application program itself.

Within the application the GUI elements are reliant on the operating system.

b) Identify **two** GUI components that are common to all of the above and carried out by the operating system.

Outline how the use of abstract GUI components simplifies the application programming.

8: The operating system in the latest mobile phones allows the user to open more than one application at the same times.

a) State **three** possible applications that might be open at the same time.

9: State the purpose of buses in a modern computer system.

10: Explain how the use of cache memory can speed up a computer system.

11: Convert the following to binary:

23
194
3940
482

12: Convert the following to decimal

0111100
1011011
1100011
1010101

Chapter Three - Networks

Essential Understandings: “How data actually sent across networks”

Key Questions:

How do I get information from networks?

What networks are there available to use?

What security issues are there to consider when using networks?

Have networks fundamentally changed the way we work and learn?

How is hardware used to send data around the world?

ATL Skills:

Social: Take responsibility and plan for ones own actions.

Self Management: Consider ethical, cultural and environmental implications.

So far in this book we have covered the use of networks and how networks have changed peoples lives. However, we have not looked at what a network is, what types of networks there are and how networks function. This unit will cover these topics.

Network Fundamentals

Types of Networks:

There are many different types of networks, each with their own unique purpose. The following is not an exhaustive list of networks but are the main networks to be aware of.

Local Area Network - A network for one organization that covers one building or campus. It holds the data and storage for that organization on that campus. To run this network you will need a server and machines to connect to the server. You may also have printer, scanner and other devices connected to the network. An example would be any network in an office building.

Virtual Local Area Network – Virtual area networks have the same characteristics as a local area network however, instead of using their own wiring etc. they will make use of wiring currently in place but because they are virtually a local area network they will not be able to see other devices on the same physical connections. An example of this may be if several companies share a building they may use the same network cabling but each company has its own separate network.

Wireless Local Area Network – A Wireless local area network functions exactly the same as a LAN without the wires. The advantage being people can connect to the network with tablets and mobile phones.

Wide Area Network – A wide area network is a network that spans larger geographical areas for example they may span countries or continents. The purpose of the network is to share information and store data. It is likely that a WAN makes use of both private hardware within buildings but public hardware to connect across countries. An example of this would be if a company has offices across several locations and they are all connected together.

Internet – The Internet is the most common WAN it is a connection of networks all joined together to cover the world. Most people connect through a telephone socket in their home/office, which is connected to a local exchange that is connected to a national exchange, which then connects to another national exchange.

Storage Area Network – Due to the limitations of storage on individual machines, storage area networks can be used to store data. The SAN will most likely consist of many machines connected to a network that contains many storage devices. Instead of storing data locally on a machine it is stored on the network and accessed when required. An example of this would be in the film industry, film is very memory intensive, when editing movies it is likely that footage is stored on the network and only moved to machines when it is being worked on.

Extranet – An extranet is a network that has a public section and a private section. The public section is used to show public information such as events, information about the company etc. however it will also have a private area people are required to log into to get to more private information. The extranet usually uses a website to gain access. An example of this would be a school website that has information about the school but a log in area containing materials for study etc.

Personal Area Network – A PAN is usually within one household and often uses wireless capabilities. In a PAN many personal devices are connected together and can share information if necessary. An example would be a household that uses laptops connected to the Wi-Fi, as well as tablets, games consoles and televisions, there may also be storage on the network where data can be moved to.

Peer to Peer Network – A P2P network allows users to share information however the information is stored on each person's machine and accessed from each person's machine rather than a server. Although they can be used for legitimate reasons they are often used to share copyrighted material. Users download the peer-to-peer software, they become a seed for each of the materials they wish to share and when people put out a request for the material their computers start to upload the data to others.

Virtual Private Networks (VPN)

Virtual Private Networks allow people to use public network connections to access private networks. For example connecting to a work private local area network from home or another location. Using VPN technology allows companies to keep their local area network safe but provide a secure method of connecting from outside.

The VPN acts as a trusted mid point between the computers outside the company network and the company network. First of all a request is made from the computer to join the network. To do this it contacts the VPN and sets up a secure connection using authentication. Once the secure connection is established the VPN uses tunnel software to set up a link to the Local Area Network, because the request comes from a trusted IP (the VPN) the authentication occurs and the computer can connect through the VPN. In addition the VPN model provides encryption, all data travelling along the connection is encrypted so even if attacked only encrypted data would be seen. It provides message integrity so it is clear no tampering has occurred.

Another use for VPNs is to provide people with access to an IP address with a different geographical location. This can be useful in many situations such as accessing data where it may be censored, being able to access information from another country if away from home. As with peer-to-peer networking people have exploited the use of VPN's to participate in illegal activities such as accessing television shows that would otherwise be unavailable in their current country.

Area of Knowledge: Ethics

Research has shown that due to Network technology improving people work longer. Should businesses make a conscious effort to stop people working longer or now much of the work is less manual has it become the social norm?

Data Transmission

If I developed a system independent of other people chances are with the right talent I could build a perfectly good system.



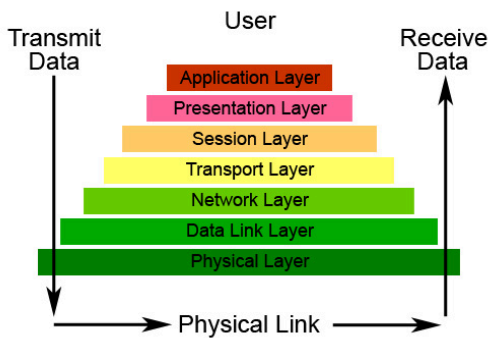
Here is my system, it functions just as I wanted and it has all my information on it. However simultaneously somebody else built a system that suits their needs and stores all their information. What happens if we want to share information?



We can't, the way my machine functions is completely different to share information we have to share in the old fashioned way. How do we overcome this problem?

Standards: Standards have helped to allow communication over a network. By developing standards that are common across systems we have allowed communication between systems developed in different ways. This means that even if the machines do not understand each other, they understand how to send each other information to be read.

The Seven Layers of OSI



Open Systems Interconnection (OSI Model)

The OSI model is a conceptual model that shows how the data is transferred across a network. The OSI model has **seven** layers each responsible for a different action relating to sending information.

The top levels are the levels that deal with the user and the bottom levels deal with actual physical transmission.

The function of each layer is briefly outlined below:

Application layer - closest to user, provides the support to the application that needs to the network resources. We interact with this layer

Presentation layer – The layer responsible for the transfer of data between formats. So the data can be read by receiving computer o/s level

Session layer – Responsible for creating a session between the two computers

Transport layer –makes sure the packets get there correctly

Network layer – Adds the IP address of sender and destination IP address router level

Data link layer – responsible for transferring the data, takes the binary and makes it readable by structuring it

Physical layer – actual connecting devices,

To what extent would the Internet be pointless without standards?

Standards are a means of communicating between different servers, so local servers and networks can be connected in order to share information and data. Without standards no networks could communicate, thus the Internet as a whole would not work, because nobody could access anything.

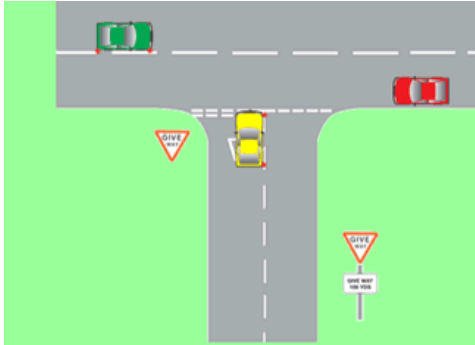
The whole point of the Internet is to give a medium for sharing data, without standards, which give the ability to share data between systems, it would have no purpose.

This would affect the majority of the world, a specific example that would be affected would be multinational corporations. Without a medium for sharing data from one office in one country to another, their business would be less efficient. Additionally, the clients would be affected because of their inability to interact with the business.

The Internet would be totally useless because systems are programs within your computer that are made in a language. The Internet gives us access to other people's systems and gives us the ability to download and use them. Without standards there is a very high risk of systems being in different languages not allowing others to use systems available to them on the line.

Protocols

When discussing standards it is also essential to think carefully about protocols. Protocols are the list of instructions devices agree on to determine how to communicate. It is an agreed behaviour between devices. Without them devices would not be able to communicate as: data integrity could be compromised, the flow of data could be interrupted, deadlocks could occur, congestion may happen and error may occur. An example of this can be demonstrated when we think about driving.



In this situation there are two cars turning right and a car turning left. Who has the right of way? If we did not have standard, accepted protocols in driving then this could become a problem. The drivers would sit in the deadlock until somebody made the decision to move. Protocols in network traffic also **prevent deadlocks** in this way.

Protocols also cover the following:

Data integrity: data integrity is concerned with the validity of the data, data can be changed in transmission in a number of ways but data integrity checking aims to ensure the data received is the same as the data sent. This is completed using Hash functions, Hash message authentication codes perform a calculation on the data and then sends the code with a secret key to the destination. If the hash codes match it is assumed the data has not been tampered with.

Error Checking: A check digit is sent with the packet, it is recalculated at the destination, and if this matches it is assumed that the data has not been tampered with.

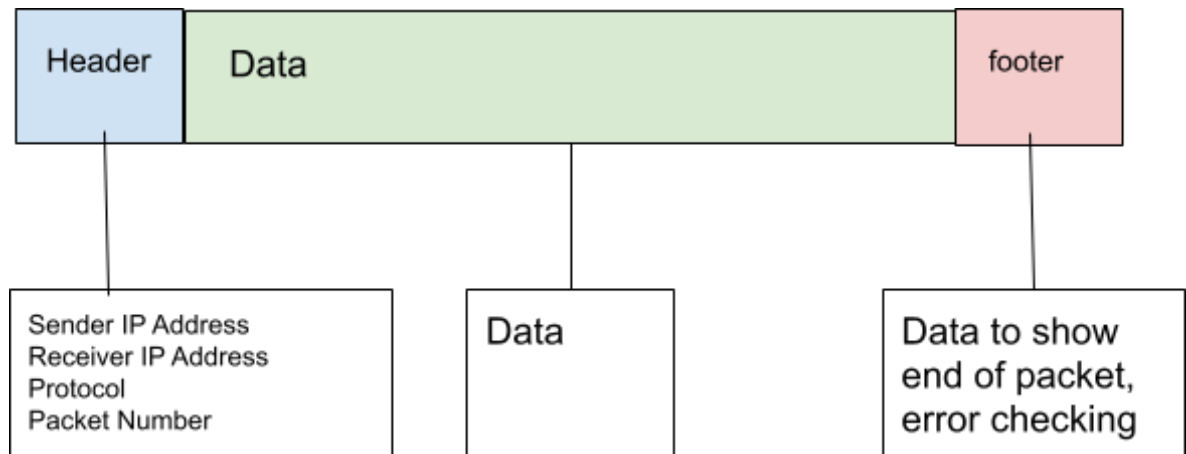
Sending Data

Data Packets:

On your machine if you look closely at activity monitor you may have seen this following:

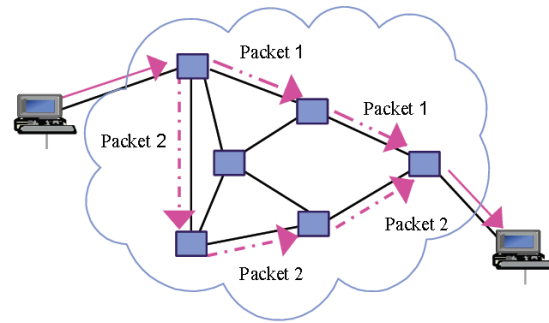
Packets in:	2,435,052	PACKETS ↕	Data received:	1.44 GB
Packets out:	7,422,942		Data sent:	8.49 GB
Packets in/sec:	484		Data received/sec:	39.7 KB
Packets out/sec:	3,413		Data sent/sec:	4.69 MB

This shows the number of packets my machine has received and sent. I sent a lot more data than I received. Packets are how data is sent across networks. The data to be sent is broken down into packets, numbered and sent on their way. At the other end the packets are collected put into order and unpacked. Data is sent in packets as it is more efficient, sending a big file as one packet would cause unnecessary congestion on networks.



How is data transferred?

Packet Switching



Data is split into packets and sent along different paths around the network to get to its destination. The advantage of this is they are harder to intercept as they are all following different routes. However, they can arrive in non sequential order at different times so it could take longer to receive the data and the data could get corrupted.

Circuit Switching

Data is sent along the same route. The advantage to this is the packets are sent in order and will arrive in order so they are easier to piece together and less likely to get lost. The issue is they are easier to eavesdrop upon and could be intercepted.

Like all networks, computer networks have flaws. Even if I have the fastest connection in the world it is limited by the connections it is connected to.

An example being the rail network shown. Traffic can move very fast in the Davos valley and very fast in the St Moritz valley however between the two valleys is very slow because the connections are much slower.

Another example would be the school network, inside the school the network is very fast, as is the connection to the outside networks however when we reach the local hub traffic is slowed which when many people are trying to watch an offsite video on the network can slow the network down.

One way to help prevent the slowing of traffic on the network would be to compress the data, we looked at data compression in the computer organisation unit. However, data needs to be compressed so it fits into less packets and is easily transported across the networks. The less packets on the network the less congested and therefore faster the network.

Transmission Media:

When creating a network there are many different transmission mediums you can use:

Description	Speed	Reliability	Security	Cost
Coaxial Cable	Much slower than fibre optic	Very reliable but can have interference	Reasonable although they can be listened to	Relatively cheap
Fibre Optic	Fast	Very reliable	High light does not make a noise	expensive
Wireless Technology	Depend on further connection	Can be blocked easy and has limited range	Can be intercepted relatively easily	Not too expensive
Twisted Pair	Faster than Coaxial but slower than fibre optic	Very reliable but can have interference	Reasonable although they can be listened to	Relatively cheap

Wireless Networks

Wireless networking has made connectivity much easier, instead of having wires connecting devices they are connected to each other using wave technology.

Advantages:

They are cheap to set up
 They can be set up anywhere as no need for wiring
 Multiple devices can be added with ease

Disadvantages:

Interference can occur
 The connection is not as stable as a wired network
 Quality drops if going through walls etc.
 Can be hacked easier than wired
 Slower than wired networks

Components of a WiFi Network:

In order to have a Wifi network you need the following components:

A device with a wireless network interface card, a wireless router with a wired connection to a physical network.

Wifi networks can operate on the following frequencies(radio bands):

Name	Description	Features	Pitfalls
WiFi	Using waves to connect devices using a wireless router	Up to 54mps in theory, and range of 200 meters	These speeds are rarely achieved and Wifi signals can be interfered with
WiMAX	Works the same as wifi but uses a different / faster standard	In theory 1gbps and a range of 30 miles	the technology is still in its infancy and is still likely to be interfered with
3G	Uses telecommunication towers to provide internet access	For a network to be 3g it has to have a minimum speed of 144kbps but it is more likely to be around 400kbps	You have to be in an area with 3G coverage, in some remote areas this is not achieved yet
4G	Uses telecommunication towers to provide internet access	For a network to be 4g it has to have a minimum operating speed of 1.5 gbps	You have to be in an area with 4g coverage although it is becoming more prevalent it does not have full coverage yet.

Future developments:

One development that might possibly replace WiFi is LiFi, using light emitting diodes, LiFi, uses light to transmit signals to devices. It is extremely fast with speeds up to 224 gigabits per second. The downfall of this technology is that it requires the light to be seen.

Network Security

Networks allow us to send data to many devices across the globe. They have enabled us to build global corporations that share the same data. They allow us to speak to people on the other side of the globe but they do all this through sharing data. There is rarely a week goes by where there is no news of a data breach in a system, people stealing data because of flaws in networks. In this lesson we will look at the different security methods employed to stop these breaches and their perceived effectiveness.

Physical

it is a good idea to have physical security, such as locks on doors etc. In the past people used to leave their routers either unprotected or in windows with the username and password on display, this led to people using the devices without authorization.

Coded

Security	Advantage	Disadvantage
Encryption	Encrypt all data passed on a network as this means that even if data is intercepted it can not be read.	Unless it is automatic people might forget to encrypt their data.
Trusted MAC	MAC addresses are the unique identifier assigned to items on a network. Having a "white list" of trusted MAC addresses allows the network prevent access to and from non trusted places.	Using only trusted MAC addresses could mean that genuinely good websites are blacklisted by accident.
Protected Access	Use a username and password to gain access to the router means that only authorized and trusted users can access the system.	If there is a high turn over of people in the company constantly providing user names and passwords is time consuming.

Chapter Three – Review Questions

1: Define the following terms:

- i) VPN
- ii) Denial of Service Attack
- iii) OSI Model
- iv) Data Packet
- v) Compression

2: Explain clearly the difference between Packet Switching and Circuit Switching when sending data across a network. Stating Advantages and disadvantages when appropriate.

3: Describe three security measures that could be taken to prevent hacking on a network.

4: Describe why VPN could be a disadvantage for company employees.

5: Explain 3 ways in which wireless networks could be created.

6: Explain 2 ways in which the speed of a network could be affected.

7: Explain the difference between a LAN and a WAN.

Chapter Four - Computational Thinking

Essential Understandings: “By thinking in a structured and organized manner using the tools of computational thinking, problem solving becomes easier and larger problems become more manageable” Key Questions: How can I use logical and procedural thinking in order to solve problems? What are the strands of computational thinking and what do they look like in real terms? Without the skills learned in computational thinking would problem solving be more difficult? Do computational skills make big problems less daunting?	ATL Skills: Communication: Use intercultural understanding to interpret the information. Use and interpret a range of discipline specific terms and symbols. Thinking: use models and simulations to explore complex systems. Self Management: Make connections between various sources of information.
--	---

What is Computational Thinking?

Computational thinking has recently emerged as one of the key skill sets to have in the 21st century. Essentially it is the ability to see a big problem and break it down into its components parts in order to manage it more easily.

The key concepts of Computational Thinking are as follows:

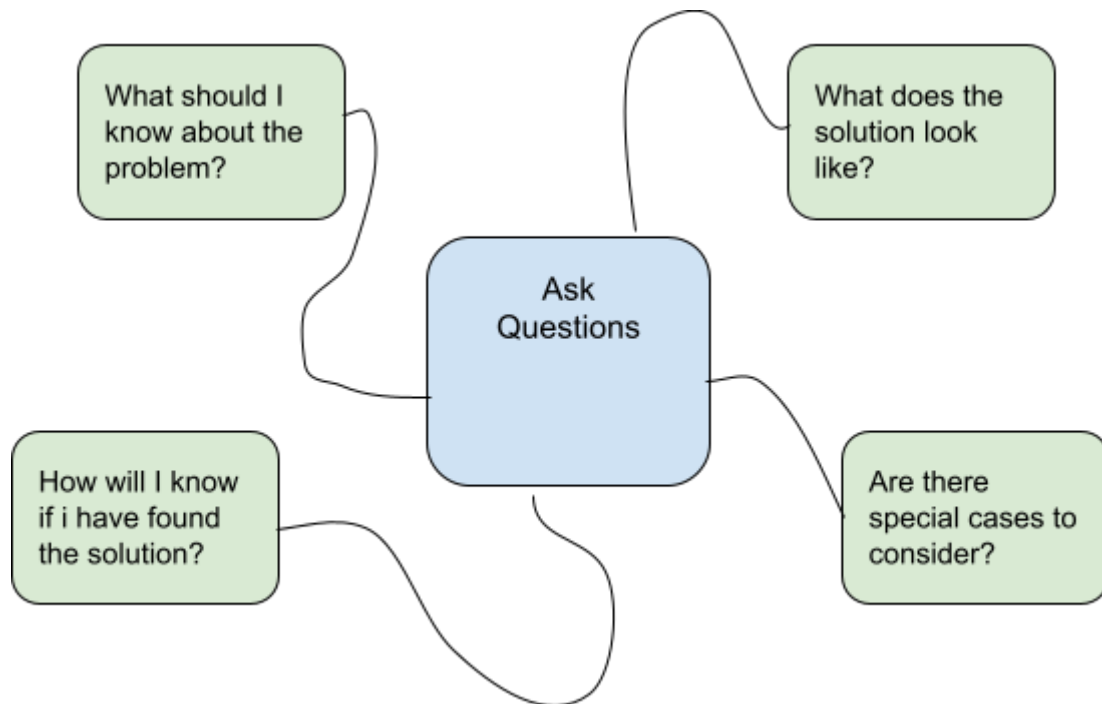
- Thinking Procedurally (or decomposition)
- Thinking Logically (also known as Thinking Algorithmically)
- Thinking Ahead (also known as Thinking Algorithmically)
- Thinking Concurrently (being able to organize tasks logically)
- Thinking Abstractly

Thinking Procedurally

Thinking Procedurally requires us to identify the task that needs to be completed, enables us to evaluate the correct order for the task and allows us to understand that tasks can be separated and completed by different people/methods.

Stage One:

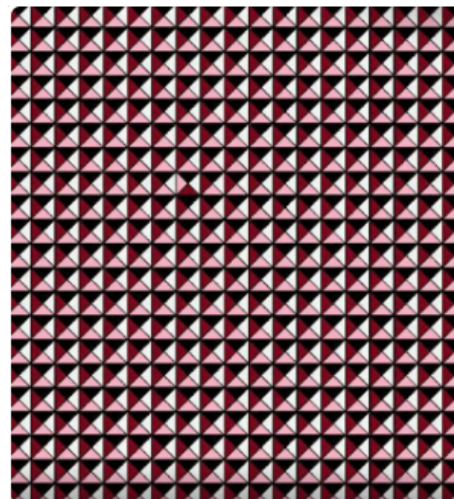
The first stage of thinking procedurally is to think “What is the problem?”



Stage Two

The next stage is to look for patterns.

Look for trends in the data you have been given, are there any generalizations you can make? If there are any outlying data items are they exceptions or a second emerging pattern?



Stage Three

```

Organise everything together;
Plug in kettle;
Put teabag in cup;
Put water into kettle;
Wait for kettle to boil;
Add water to cup;
Remove teabag with spoon/fork;
Add milk and/or sugar;
Serve;

```



Scones

Send, Share and Save your Favourites with just one click

Print Tweet Like Pin It

INGREDIENTS:

- 225g/8oz Odiums Self Raising Flour
- 150ml/¼ pint Milk (approx)
- Pinch Salt
- 25g/1oz Caster Sugar
- 25g/1oz Margarine

METHOD:

1. Sieve flour and salt into a bowl, stir in sugar, if used.
2. Rub in margarine.
3. Add sufficient milk to make a soft dough.
4. Turn onto a floured board and gently knead to remove any cracks.
5. Roll out lightly to 1" in thickness. Cut into scones with a cutter dipped in flour.
6. Place on a floured pre-heated baking sheet/tray, glaze if liked with beaten egg or milk.
7. Bake in a pre-heated oven 425°F/220°C/Cas 6 on upper shelf position for 10 mins. approx.
8. Cool on a wire tray.

Look for the steps that need to be completed in order to complete the task.

As well as identifying what needs to be done we need to be able to decide whether the order in which the task needs to be completed is relevant or not. Is it that some things need to be completed in order? can everything be done at every time? Can some things be done together?

Most of the time the stages need to follow a set order. There will be an order to follow sometimes we can deviate but on the whole it is set.

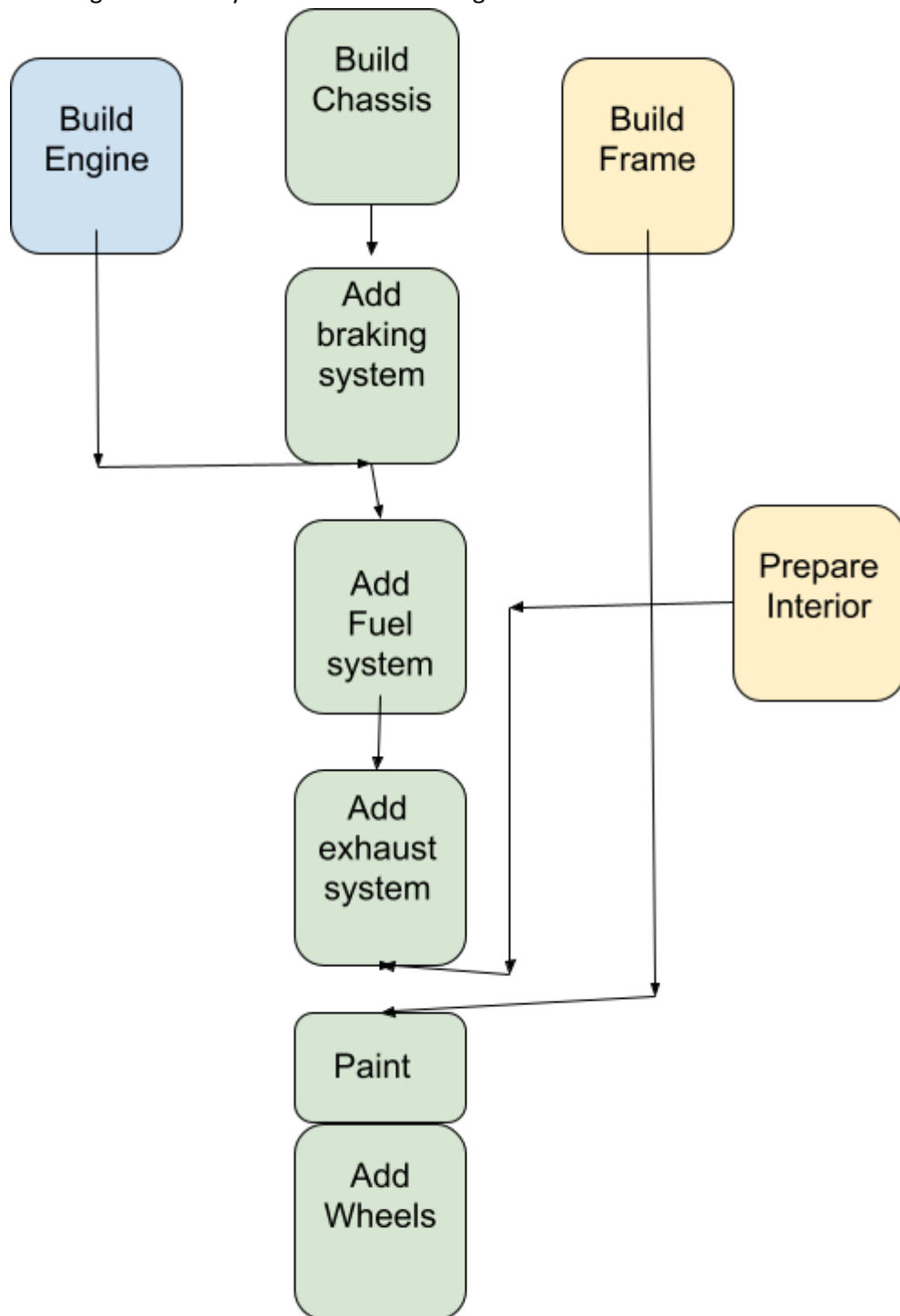
Sub Procedures

Sub procedures are a list of things that can happen independently within the main task. They can usually be reused in other programs or systems. For example if were to build a car you could say that the main method would be:

- 1: Build engine
- 2: Build chassis
- 3: Add braking system to chassis
- 4: Add fuel system to chassis
- 5: Add exhaust system
- 6: Add interior
- 7: Add frame
- 8: Paint
- 9: Add wheels

Some of these things can be carried out at the same time. For example an engine could be built at the same time as I build the chassis. Some of these things could be reused in different procedures.

For example: An Audi A3, Golf and Skoda Fabia (I think) are all the same chassis they just contain different frames, slightly different engine configurations and different interiors. To make these cars Volkswagen would not change the factory just the procedure. e.g. if i separate the building an engine component of the making the car, the adding a frame and the adding the interior component then I can create any version of the car I want. As a diagram this may look like the following:



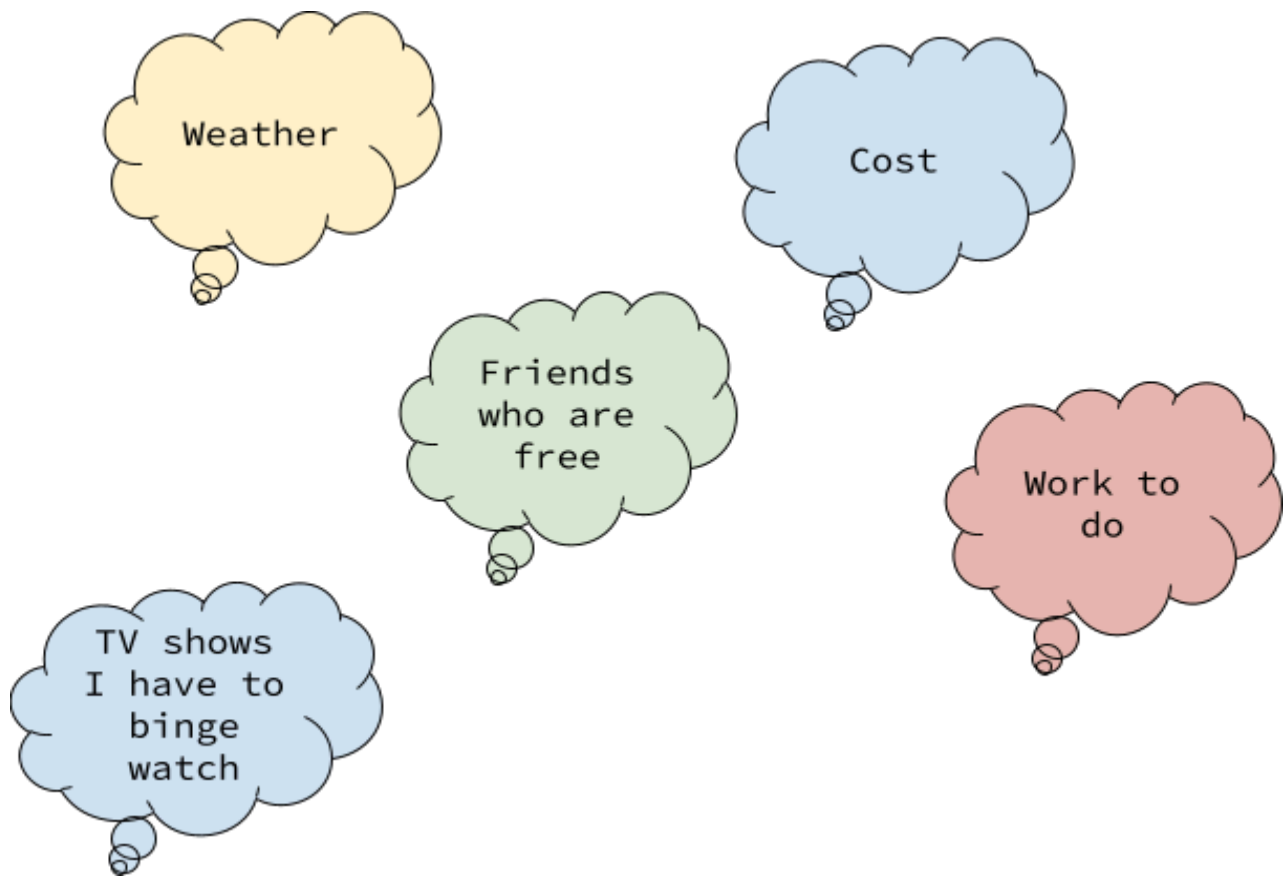
Thinking Logically

We make decisions everyday hundreds of them. Should I leave the house at 730 or push it to 735? should I catch the 8 or the 51 bus? Should I have canteen food or go to the salad bar? Many of these decisions won't make a big difference but nevertheless we make them all the time. Decisions that may be more pertinent to you at this stage of your schooling may be where to study and what to study. These things do have a knock on effect for your future.

Below is an example of some decisions and the effects that they may have.

Decision	Chosen Outcome	Effect
Whether to stay in bed 5 minutes longer or get up with the alarm	Stay in bed 5 minutes longer	Making your friend wait longer and possibly miss the bus
Whether to do homework or watch the next episode of a tv show	Watch the next episode.	Have less time to do my homework and possibly less time to sleep
Whether to take the short cut which is messy or the longer road	Take the short cut	Look a little bit muddy at school
Whether to be lazy or going running	go running	Feel happy because I have been healthy

It is important for programming to be able to identify the factors (or circumstances) around the decisions that need to be made. For example if I wanted to write a program that told you what to do at the weekend what factors would be involved?



IF AND OR NOT ELSE - these are important words that you will come to associate with both decision making and in the magical art of programming. IF is the word that accompanies most decisions made, AND OR NOT ELSE help us to make up the conditions associated with the decision.

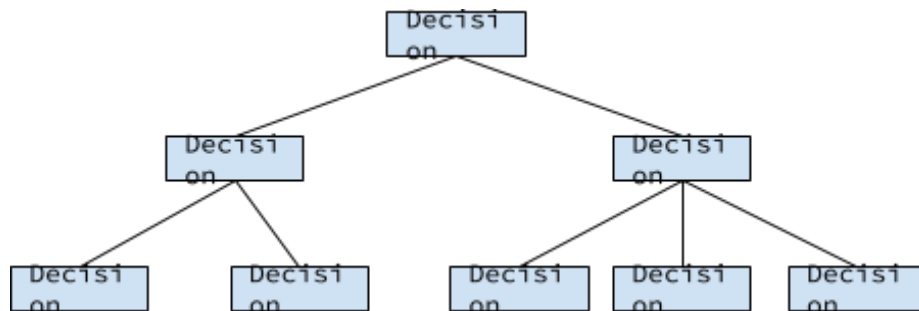
for example here is the thought process I had for choosing a Halloween costume:

```

IF I am at school AND there are young students present
    then I will be a witch
ELSE IF I am at school AND it is only older students
    then I will be my living dead doll
ELSE IF I am at school AND it is freezing outside
    then I will wear my cheetah onesie
ELSE IF I am NOT at school
    go as a TV character from the 80's 90's
ELSE don't bother
  
```

You can see from this that you can break most decisions into statements using the words IF AND OR NOT ELSE

Branching is the word that describes what happens as a result of the conditions taken. It is effectively the steps that you follow as a result of what has been said. It is the path that is followed. This could be modelled using the following.



Link to programming IF statements are the backbone of computer science. They allow us to create functioning programs that allow user interaction.

Way of Knowing: Reason

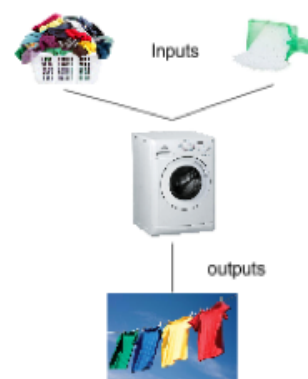
To what extent can reason be distilled into algorithms? When we are reasoning are we really just making algorithms in our head to help decision making? If this is the case can we create one big self help book for all life's situations in order to help people out?

Thinking Ahead

So far in the course we have thought about the process we would carry out to solve our problems however, we have not stopped to think about the tools we need in order to solve the problems. We need to think about the tools we need in order to complete the task and then outputs that we expect to receive from the system. If we have these ideas firmly in our head then we can begin to plan our systems easily.

Thinking ahead makes us think about what we need in order to do a task and what we want to achieve at the end of it.

An example could be TOK homework. You need to complete a presentation about ethics what do you need? And what do you want to achieve? It is likely you will need a real life situation and class knowledge to make the presentation and the outcome is a successful assignment.



Thinking about what we need is called **pre planning**. Why is pre planning required? if we pre plan then we can collect the things that we need in order to complete a task, this enables the task to go smoother as we know we have all the inputs we require. For example baking a birthday cake. The pre planning that is required would be to know what ingredients I need for the cake and the icing. This is important as if I don't have them then can't have the output that I want.

Algorithms and methods, the backbone of all computer programs. Very rarely in programming will a program follow instructions in a linear fashion. Where we can start at the beginning and follow each instruction one by one until we reach the end. Programs are designed to allow interaction between machine and user, in order to for this to happen we need to have different methods associated with different tasks. The simple way to view methods is in the following way:



Therefore when designing or using a pre existing methods you need to know the pre conditions (inputs) that the method needs in order to complete the task.

In programming these inputs become the parameters you have to pass into the methods. The outputs become the return types you expect from the parameters.

Exceptions:

Most problems contain exceptions that are outwith the norm. An example I can think of in a school that runs trip for every child there is a selection process usually all grade 9's go to the grade 9 trip. All grade 10's to the grade 10 trip, Grade 11's go to the European trips and the Grade 12's go outside Europe. However due to personal reasons some grade 12 students may choose to stay in Europe. This is an exception to the rule, it would be something we have to consider but only after we have dealt with the patterns.

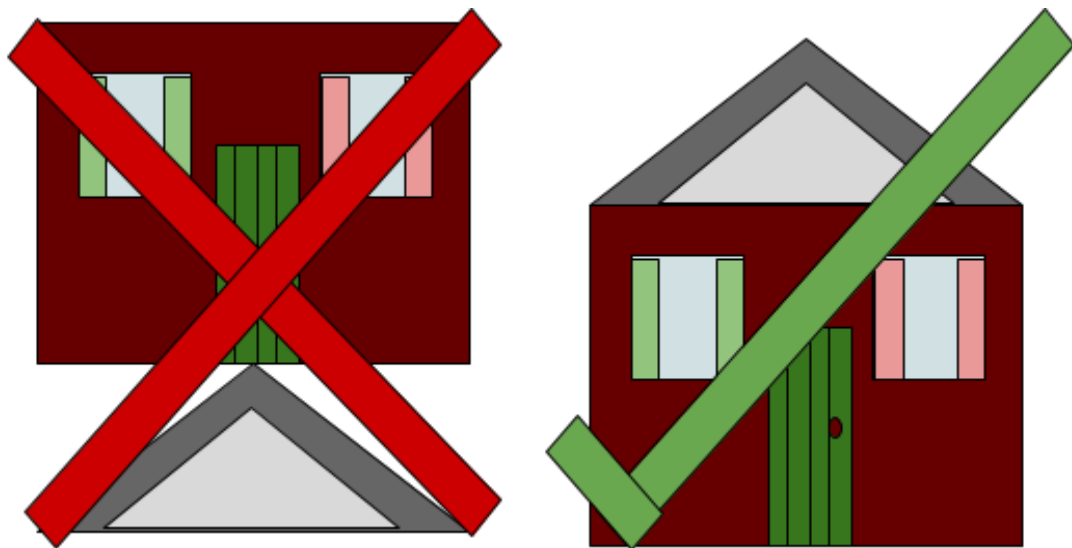
Thinking Concurrently

For the purpose of this section we are going to have to think about building a house. The stages involved in building a house can be summarised in the following way.

build the foundations
 build the walls
 add the roof
 floor the house
 add electricity
 add water

- add heating
 - add windows and doors
 - paint
 - landscape

some of things can only be done one after the other for example. You can't add a roof to a house without walls. That would not work.

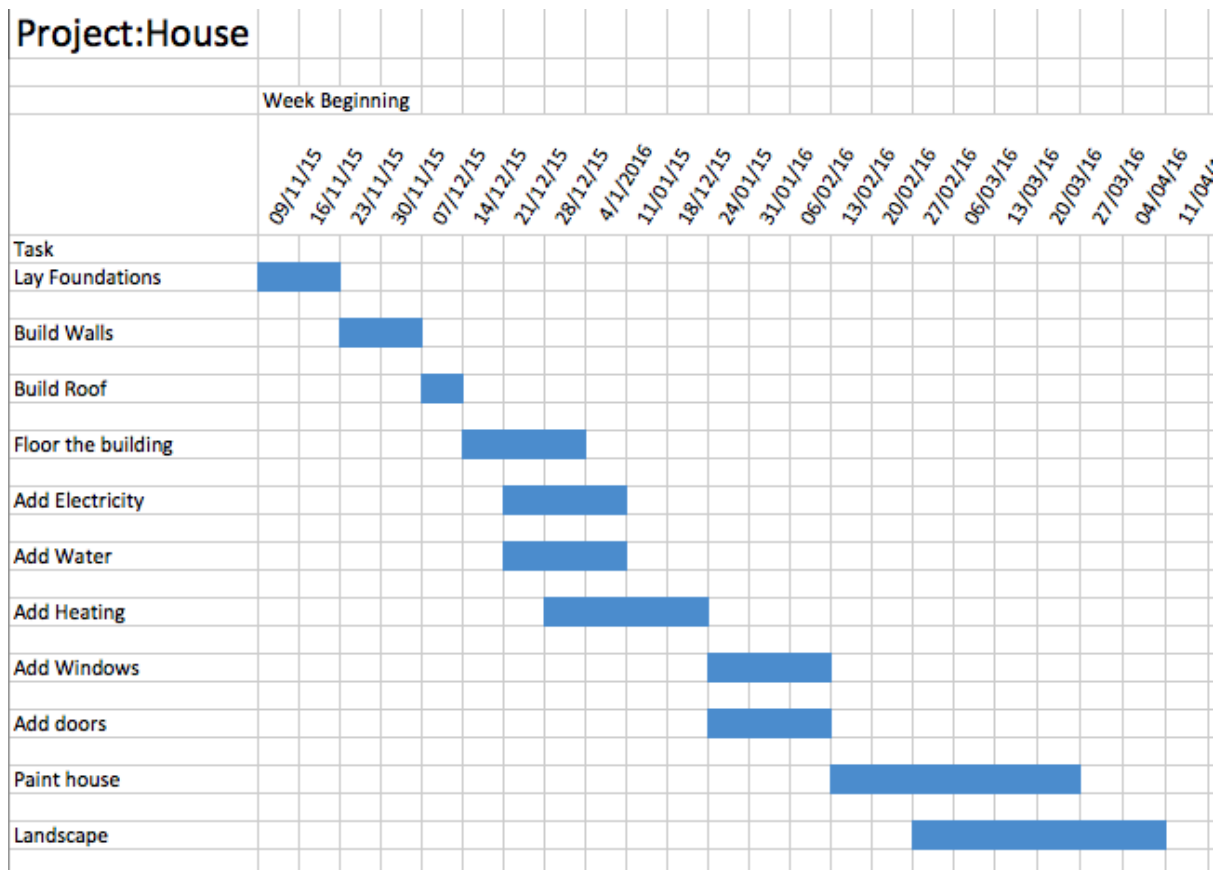


However some things can be completed at the same time.

For example:

Adding electricity and water
 Adding window and doors
 Painting and landscaping

This is usually represented in a Gantt chart.



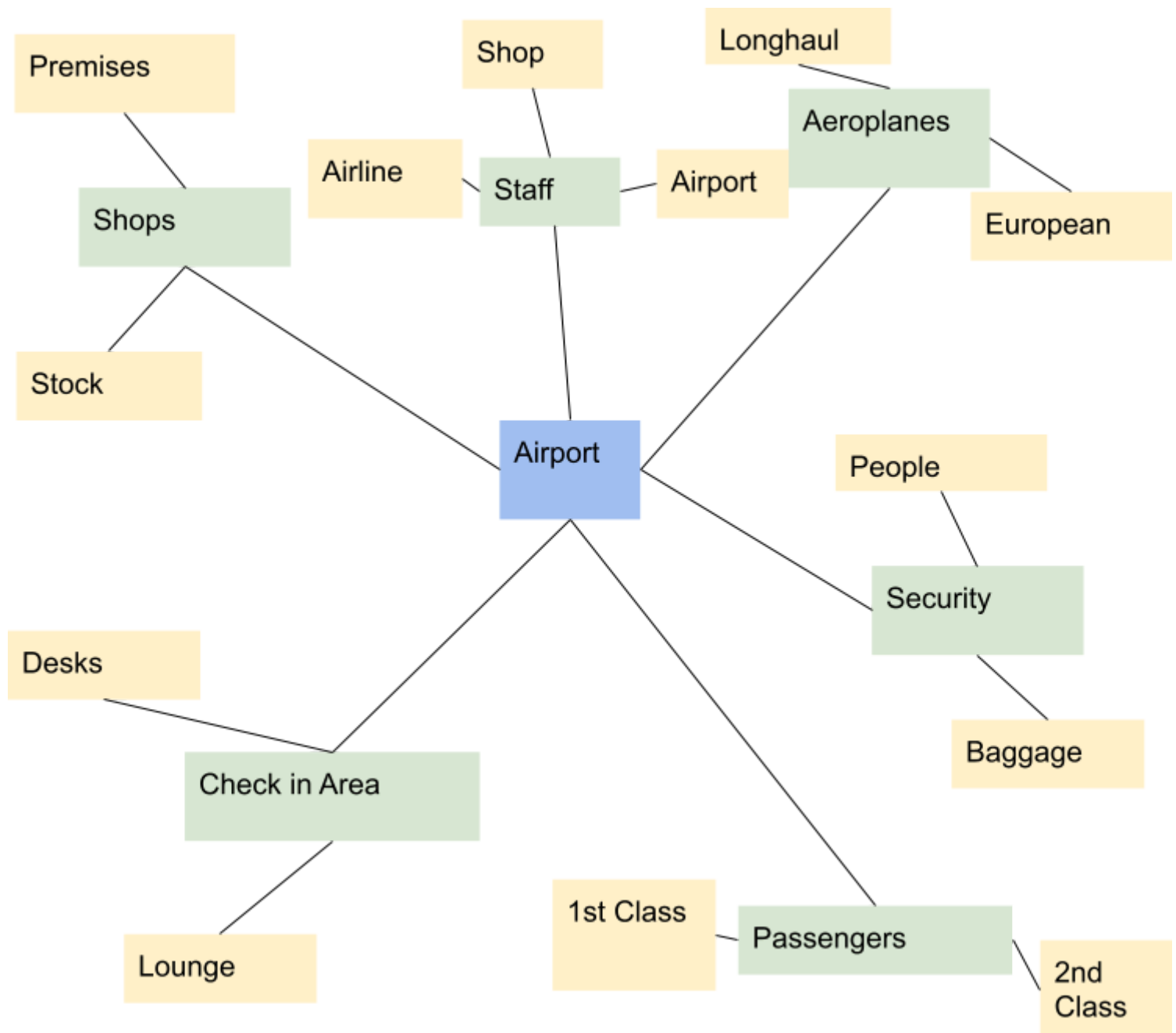
Bars that appear after one another means that the tasks are **non - concurrent** the previous task must be completed before the next one is started. Bars that appear in line with each other or overlap in some way are **concurrent** they can be completed at the same time.

Concurrent processing can be used to complete problems quicker. If you can divide the tasks up and then complete them together then the length of the project is shortened. There are both advantages and disadvantages to this as described below:

Advantages of using concurrent processing	Disadvantages of using concurrent processing
The overall time taken to complete the project is decreased	More people on site could lead to more complications
Less wait time on projects as things are being carried out simultaneously	Harder to manage more people on site.

Thinking Abstractly

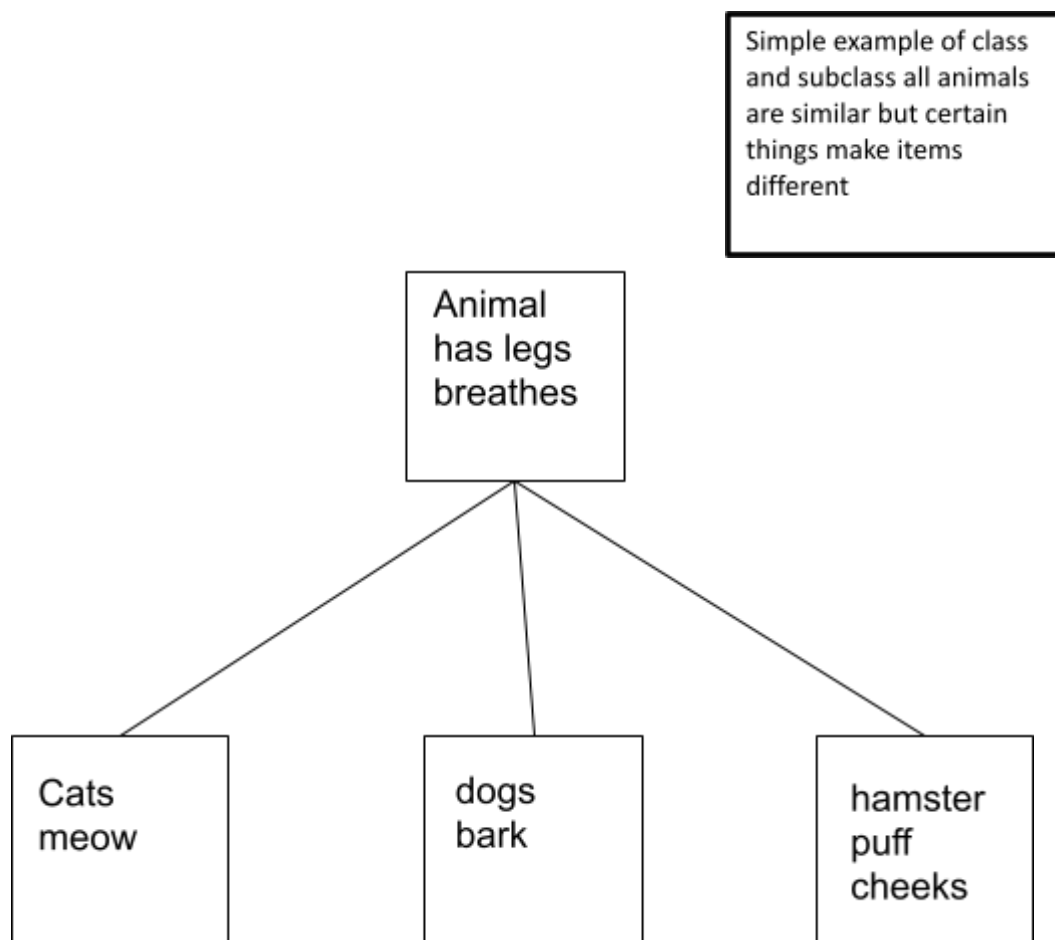
Abstraction in simple terms is the representation of the “real world” or real world concepts in the virtual worlds or programs. Abstraction requires us to start thinking about objects as separate parts rather than the sum of its parts. One example of this would be an airport. Airports are often just thought of places where we go in order to get somewhere different but what makes up an airport really?



If we were to implement the airport program then many of these boxes would become classes and subclasses of each other.

Another way to think about abstraction is looking for the commonalities within objects and then identifying the difference that make them unique. One example of this would be animals. What makes us different from animals? How is a fish different from a cat?

As all computer systems are essentially aiming to represent the real world virtually, abstraction is essential. We need to be able to show the interactions between objects in order to elicit information or perform calculations. Whether this is completed through a database, a program or a simulation model all objects need to be represented. In order for programs to work efficiently we need to have as little duplicated data as possible. In databases this is completed through a process called normalisation in coding it is by using superclasses and subclasses.



+

So how does this look in programming and why do we do it? You know that duplication of data is a bad thing because having to update data in many places is inefficient and leads to mistakes. Therefore we use super classes and sub classes. Using the animal example again we are going to create a program containing classes and sub classes.

Its possible to see abstraction everywhere. We use an abstraction in our daily lives all the time. For example a calendar is an abstraction of time, it is a visual representation of how we see time.

Perception

One of the problems with abstraction is that the way the item is abstracted depends on the perception of the person doing the abstraction. For example creating the classes and methods of a school from a teacher's point of view would be very different to a student's point of view. To what extent do you agree that abstraction depends on perception? Is there any way we can make it less so?

Further TOK Thoughts

To what extent is a computer model of reality analogous to the knowledge models built up by humans about the real world? Can we model things in the real world when many of the things we take for granted in the real world are arbitrary divisions?

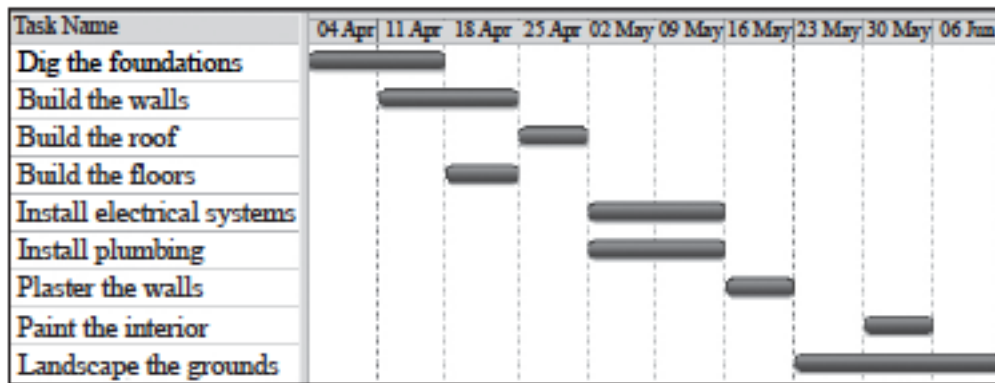
To what extent is intentional human involvement essential in the development of knowledge in the Arts? If we have algorithms to create Art do we need artists?

To what extent is faith essential in the use and development of computer and software systems?

Chapter Four – Review Questions

1: Mr Smith is having a new house built and will require local tradesmen to complete a number of tasks:

The Gantt chart below shows the tasks involved in the building of the house.



- Define the term concurrent processing.
 - Identify two tasks that are carried out concurrently.
 - Identify two tasks that are carried out sequentially.
 - Describe how the idea of abstraction applied to one of the tasks.
 - Explain **one** advantage and **one** disadvantage of carrying out a number of tasks concurrently.
- Train Stations have many different sections, represent a Train Stations as an abstract format.
 - What are pre conditions and why are they required when discussing algorithms?
 - Why is breaking a program into objects useful when working as part of a development team.
 - Using your knowledge of thinking procedurally, create a procedure for booking flights to New York.
 - Represent the following as a flow chart (thinking logically).
If it is raining outside you should stay inside and do some homework, if it is snowing you should consider going skiing, when it is sunny it is good to go to the lake. Otherwise it is nice to catch up with reading.
 - Why does the following java program not work?

```
//program to determine the cost of postage
//September 2013
//L Stephenson

public class postage {
    public static void main (String [] args){

        System.out.println("how heavy is the package?");
        weight = in.nextLine();

        if (weight < 500){
            System.out.println("cost is 5 chfs");
        }
        Else IF (weight > 500){
            System.out.println("Cost is 10 chf);
        }
        else {
            System.out.println("weight not recognized");
        }
    }
}
```

Chapter Five-Resource Management (HL Only)

Essential Understandings: “In order for computers to work effectively they need complex prioritization system”

Key Questions:

To extent do you agree with the statement “Without prioritization systems we would have to be very clever in the way we worked”?

Why do computer sometimes “crash”?

Is it true, the computers we currently have are actually inefficient?

ATL Skills:

Research: Understand and use technology systems.

The focus of this unit is to look more closely at the inner works of a computer system and to think about the limitations of some systems.

What is inside a machine?

Look carefully at this picture of a deconstructed MacBook.



When reconstructed all of these parts need to be connected in someway and they need to work together to do useful things. One way of getting them to work together towards a common goal is to have a “manager” of the machine. This is where the operating system comes in to play. In the Computer Organisation unit we looked at the role of the operating system in this unit we will look a little closer.

In this unit you will be required to think about the following products:

Mainframe Computers

Servers

Personal Computers

Tablets

Laptops

Mobiles Phones and Digital Cameras

For each of the devices we will think about, what needs to be managed, What the limitations of each device is and how we can overcome these limitations.

Examples are shown below, this however is in no way an exhaustive list.

Mainframe	
	Resources to be managed: multiple processors, RAM and the secondary memory. Issues with mainframes: the space requirements of a mainframe are large. You need specialized people to run a mainframe. Mainframes are only useful if storing analysing terabytes of data and above. Overcoming these limitations: Consider the space requirements and if you don't have enough room consider using a server instead. One specialized machine can do the job of many machines.
Server	
	Resources to be managed: connections to other devices, files stored on the server Limitations of Servers: If all data is stored on them then they can run out of space fairly quickly Overcoming this problem: To overcome this problem it is worth considering if only some data should be stored on the server and some on personal machines.

Personal Computers



Resources to be managed: processor, primary memory and secondary memory, peripheral devices

Limitations to Personal Computers: Single processor machines might have issues rendering large graphic files or high quality video.

Overcoming the limitations: To overcome the limitations you could invest in a computer with multiple core processors, increase the ram or improve the video card.

Laptops



Resources to be managed: processor, primary memory, secondary memory, wireless connections

Limitations: One limitation of the laptop is its limited memory and processing power, trying to render video on this machine would be a very slow process. Complex processing will drain the battery of the laptop quickly, therefore the type of work being carried out should be considered carefully. Wifi can be slow especially if transferring large amounts of data.

Overcoming the limitations: One way to overcome the limitation is to consider whether video rendering needs to happen on this machine and if so consider editing smaller sections of video at a time. To avoid draining the battery it might be good to close applications not in use, or remain connected to a power lead. To overcome the Wifi problem connect the laptop using a wired connection.

Tablets



Resources to be managed: Processor, primary memory, secondary memory, touch screen.

Limitations to the device: trying to run a full application on this device is difficult as it does not have the power. The memory is not big if storing lots of files.

You can consider whether you need full functionality for the tasks you are completing and if not use 'light' versions of web apps. You could also consider buying a tablet with expandable memory.

Mobile Phones



Resources to be managed: Processor, primary memory, secondary memory, touch screen, network connections.

Limitations to the device: the battery life is short on phones when in high use therefore if relying on a phone to complete many tasks you should leave it plugged in or ensure there is a charging option nearby. Many functions of a phone rely upon good connectivity, if completing a video call or downloading a large file try to stay within in a high connected area.

Digital Camera



Resources to be managed: processor, SD card, battery, focus

Limitations to this device can include video recording, it will only shoot for 7 minutes at a time and the microphone is not great. One way to overcome this is to get an external microphone that you can plug in.

The Operating System

In this section of the unit we will look in more detail at the operating system, the operating system we will focus on would be one that you find on a desktop or laptop computer. It has limited network capabilities (it will connect to a network) and also supports multiple user accounts.

Inside the computer is organized chaos, I like to think about each of the areas as annoying toddlers:



There is the peripheral who just wants to be seen



Then there is the hardware who wants to be noticed



Then we have the memory who can't put anything away properly

Process Name	% CPU	CPU Time	Threads	Idk
Adobe Acrobat Updater	7.6	1:11.86	8	
WindowServer	7.2	21:26.36	4	
hidd	3.7	7:02.24	6	
launchservicesd	3.2	5:58.02	4	
kernel_task	3.2	19:33.58	98	
screencapture	1.8	0.27	2	
Spotify	1.3	5:25.23	34	
coreaudiod	1.3	5:13.60	4	
Microsoft Word	1.1	30:47.62	16	
SMARTBoardService	1.0	5:33.12	7	
Activity Monitor	0.9	1.32	6	
SMART.lnk	0.6	3:06.31	15	

The Operating System is the parent that then needs to take care of the toddlers vying for attention in a calm manner to contain the chaos. It deals with the **peripherals** by constantly monitoring

them. If they need attention, it stops all none critical processes attends their needs then moves on. The problem with **hardware** is not all hardware speaks the same language. When a device is loaded onto a system it has an associated driver, this driver enables the operating system to decode what it wants and through the translator that is the operating system all devices can talk to each other. Then there is **secondary memory** when we store files the files are stored in fragments. These fragments are stored all over the hard drive as there might not be room to store all the fragments next to each other. Each fragment has a pointer telling the OS where to go to retrieve the next fragment of the file. The operating system is responsible for finding the starting point of the file and then the next section of the file. It can also compress the data on the disk so there are more blocks available to store data next to each other. Finally the operating system needs to manage **primary memory** it does this by keeping a track of what programs are open and which program needs the processor time at any given point. If you open the activity monitor on a Mac O/S computer or Task Manager on a windows machine you can see all of the tasks the operating system is trying to manage.

Lets look at how the operating system manages resources in more detail.

Managing Memory

Computers need to be able to do more than one thing at a time for example, they should be able to have a word processing package open while allowing us to listen to music and be distracted by the internet and emails. However, unless they have more than one processor computers can not truly multi task instead they take advantage of how fast they run to pretend to multi task. To do this they can employ a couple of techniques:

Time slicing is the first technique – the processing power is split between different operations. A fixed amount of time is set for each operation once the time limit is reached the current state is saved, added to the list of tasks in its current state and the next task is tackled. This means that every operation is given the same amount of processor time but if the task is not finished within the time it has to be placed back into the queue.

Priority – the program that is “in focus” occupies most of the processors time but when waiting upon input or it is sitting idles the resources are transferred else where. The main program takes up a bulk of the time but the processor makes use of the downtime. If you open activity monitor you can see the different processes being switched between.

The main thing to remember about managing memory is because processors run so fast we will never notice that the tasks are not happening simultaneously (until something goes wrong).

Scheduling

When using the time slicing or priority systems the processor has to have some way of deciding what processes fit where in the schedule. The following techniques are options:

First in first out: This operates the same as a queue. The first job into the system is the one that is completed first. However, if there is a big job at the front of the queue this could slow the system down.

Shortest Job First: in this system the shortest jobs (as the name suggests) are always completed first. The main issue with the system is that if there are many short jobs that continue to be added the longer jobs will never be reached.

Round Robin Scheduling: each process is given equal time once that time is finished the current state is saved and the processor moves onto the next task, the unfinished task must then wait until the processor time returns to it before continuing. The Mac operating system uses this type of scheduling.

Multi Level Queue Scheduling: processes are marked as either batch or interactive. The processor schedule different processes at different times based on this. Windows uses this type of scheduling.

The amount of Random Access Memory is something that should be considered carefully when choosing a computer system. The more RAM available the more memory intensive programs you can run without issues and you can have more programs open at the same time. Sometimes we over load the memory and the operating system has to compensate for this through virtual memory and paging.

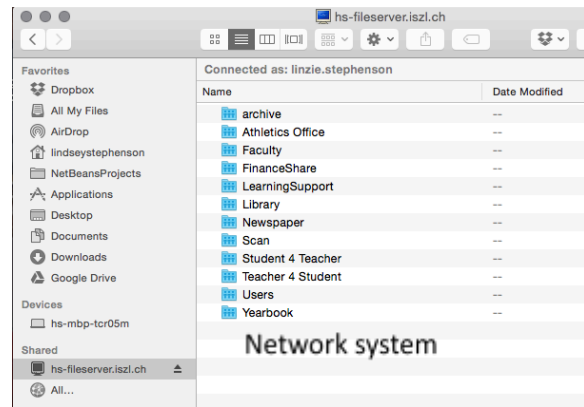
Managing Virtual Memory

We can not use secondary storage as RAM as it does not have a fast enough read / write time. Similarly we can not use RAM as storage because it is lost when the computer is switched off. However, if we overload our RAM the computer exploits the two forms of memory to make more RAM available for the user but how?

A section of the RAM is sectioned off as virtual RAM, programs that the user currently has open which are not being used are stored in this section of the hard drive. The operating system leaves a "page" in the actual RAM and if the program is selected the program is brought into the actual RAM and another program is moved to virtual RAM. The advantage of this is the user can have many more programs open than they should be able to.

Managing Peripherals

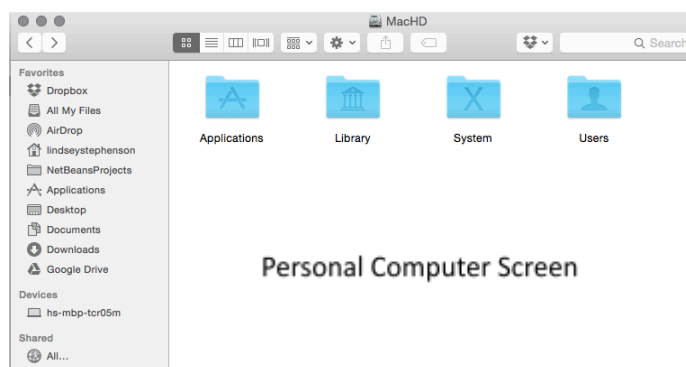
Managing Memory and Managing Peripherals are ways in which the operating system manages internal tasks but sometimes peripherals need attention too. The operating system deals with peripherals in a slightly different way. As usual there is more than one option it is system dependent.



Interrupts – inside the machine there is a bus known as an interrupt bus. The interrupt bus is linked to the processor. When the peripheral has an input or an issue it sends a message on the interrupt bus. The processor saves its current state and then attends to the peripheral. Once the peripheral has been dealt with the previous tasks are resumed. Examples of interrupts include: input from the keyboard, a read/write signal from an SD card or an issue with a printer.

Polling – polling is another way to check for peripheral issues, the processor intermittently checks each device for a signal. If there is not issue it moves on to the next task. Again as processors work so fast you would not notice the delay.

Virtualization of the System



one of the jobs of the Operating System is to make things easier to understand for the user. One way in which this happens is through virtualization. Each of the folders shows a section on the server. Although it is not literally divided the OS allows us to think that there is literal divide between sections and that data we have stored in each section will remain safe. 3

TOK Links

To what extent does the use of abstraction in computer system promote or restrict the exchange of knowledge?

How can we take the complexities of human languages and model them using graphics?

When things go wrong!

In computer systems things often can and will go wrong. In this situation the computer receives a non maskable interrupt signal on the interrupt bus. The processor does its best to save the current states and deal with the crisis but sometimes it can't and this is when your computer "crashes" sometimes it just needs to reset itself or sometimes we have to "force an unexpected reboot" (shameless IT crowd joke).

Dedicated Operating System

Many machines use operating systems some are general purpose such as window and mac and some others are specialized for example operating systems inside a car or air traffic control systems. Why would a dedicated operating system be used? In some cases it would be for safety reasons. I am not sure anyone would feel safe landing at an airport if the computers being used were not solely dedicated to checking that no planes are crashing into each other. Similarly if you needed to be on a machine in hospital you would want the system to be focused solely on you feeling better. In other cases it is more likely to be for legal reasons. It is likely most cars have similar operating systems but each has their own version of it. This is probably for legal reasons you could not have an Audi operating system making a Mercedes work, it would be difficult to then decide who owns the operating system, who is responsible for updates, who is responsible for something going wrong. Therefore legally it makes more sense for them to develop their own system.

Chapter Five – Review Questions

1: Use the Internet to compare the OS on apple mac book pro to the OS on the Apple iPhone 7

Similarities	Differences

2: Why does it make sense that washing machines, microwaves and iPods have dedicated operating system?

3: What is a reduced instruction set?

4: Why is it sensible for a washing machine / mobile phone etc. to have a reduced instruction set?

5: Why are interfaces required in a Computer System? What consequences would there be should interfaces not exist?

6: Why are single processor machines not “multi tasking”?

7: explain what scheduling in an operating system is. Give two examples of scheduling policies that exist in operating systems.

8: Explain how virtual memory makes use of paging to increase the size of the RAM.

9: What are interrupts? Why do you think they are more efficient than polling?

Chapter Six - Control (Higher Level Only)

Essential Understanding: “We are immersed in Computer Systems, whether we realize it or not they control many of the everyday items we interact with”

Key Questions:

How can I write pseudocode that models the control of sensors and physical outputs?
 How do sensors make everything easier?
 Have we become too reliant on sensors?
 How can I identify the sensors and their purpose?
 Without computer systems and sensors, everything would take longer and be more expensive.
 What are the ethics behind us using sensors to monitor people and their behaviours?

ATL Skills:

Self Management – consider ethical, cultural and environmental implications

Thinking – Draw reasonable conclusions and generalizations.

This unit concentrates on sensors and how they interact with systems to produce autonomous machines that we interact with on a daily basis.

Sensors

There are many different systems with which we interact on a daily basis including washing machines, traffic lights, automatic doors, GPS systems. For this course you need to be able to identify the sensors and processing that may taking place in any given system. However the first thing that we need to think about it what sensors exist. Below is a list (by no means exhaustive) on sensors, what they do and possible uses.

Sensor	Purpose	Usage
Proximity	A proximity sensor is a sensor able to detect the presence of nearby objects without any physical contact.	Can be used in cars, in order to inform the driver how close they are to a nearby object.
Temperature	Makes sure that objects are not over/under heated.	Used in order to regulate the temperature of a pool.
Sound	Converts a sound into an electrical signal	Could be used in an alarm system, if there is sound detected and alarm would sound
Tilt	Makes sure that a certain angle is not passed, if so, the object will automatically adjust.	Can be used in mobile technology for games, or to create electronic spirit levels.
Altimeter	Instrument used to measure the altitude of a given object over a certain level	Airplane or helicopter cockpit to get the high of the object
Oxygen	Measures the amount of oxygen in the given liquid or gas.	In Oxygen sensitive environments such as space stations
Pressure	Pressure sensor cause a reaction when a certain amount of weight or pressure is applied to them	Can be used to detect if someone is waiting for a door to open.
Speed	Displays the speed of a given object	In speed cameras to detect the speed of a car.

Once you have a knowledge of sensors you can apply them to real situations to try and understand what is happening in a given system. Examples of this are given below:



Example One – Train Doors

The sensor on the train door will stop the door from closing on the people entering the door. If there was no sensor people would be hurt if they tried to enter the train when the doors are closing. Or they would get stuck and it could cause an accident. It is likely that the door uses a proximity sensor. If people are within the proximity of the open door it will not close. Pressure sensors would not work as it is likely people will be on the inside of the train.

Example Two – Cars

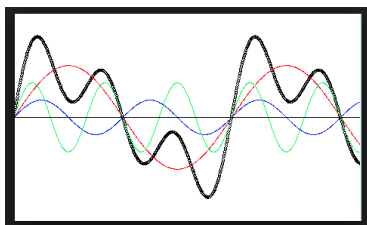
The proximity sensor in a car is a very useful tool which allows for the driver to estimate how far they are from an opposing object. This will mainly be done through infrared signals which notify the sound system to produce beeps complementing the distance the car is from the opposing object. For example if the car is closing the distance to the opposing object, the proximity sensor will send an infrared signal which will repel faster towards the proximity sensor, meaning that the sound system will be notified of how close the car is to the opposing object. This will cause the sound system to play repetitive beeps at a fast speed.



Audi A1
www.pexel.com

Making a System

When we talk the sounds we produce look like the following:



However computers only understand 1's and 0's how do we translate analogue (wave) signals into digital ones?

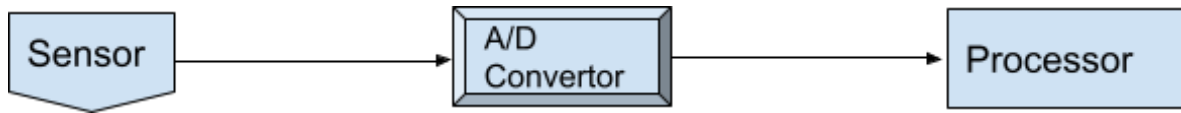
firstly there is lots of physics involved which looks a little like this:

$$SQNR = 20 \log_{10}(2^Q) \approx 6.02 \cdot Q \text{ dB}$$

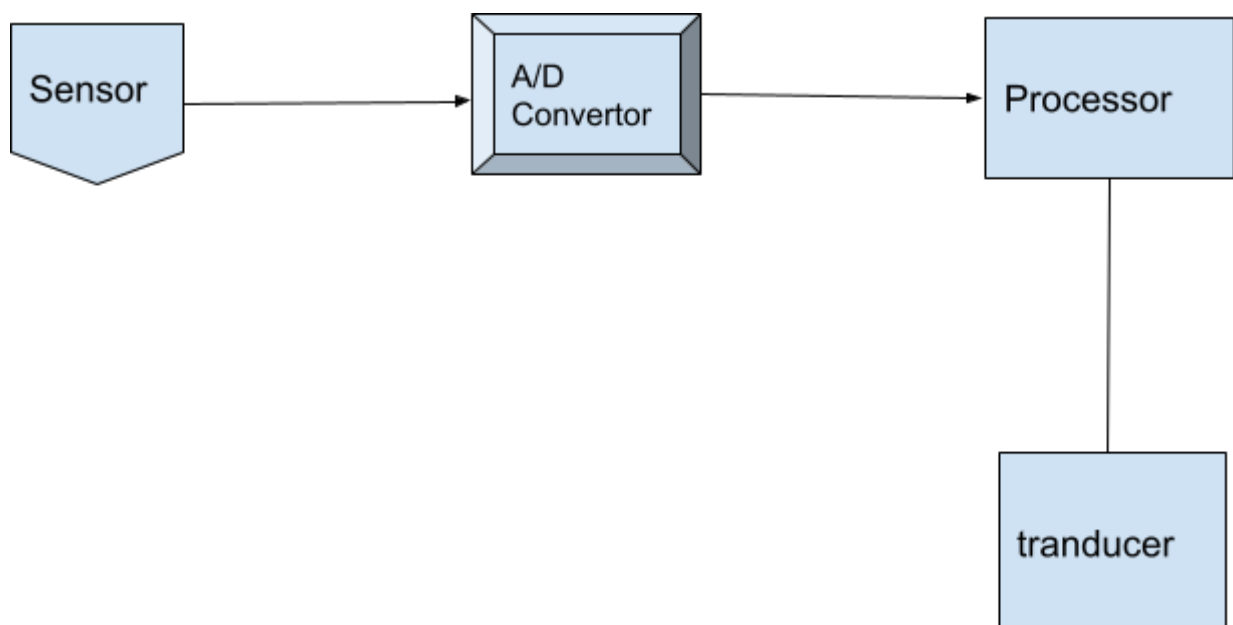
Thankfully we don't need to know that we just need to know about sampling and what happens with the data.

Sampling

Sampling refers to how we gather data from analogue signals. The computer takes snapshots of the data several thousand times per second using an analogue to digital convertor. These snapshots are then fed into the processor as digital data.

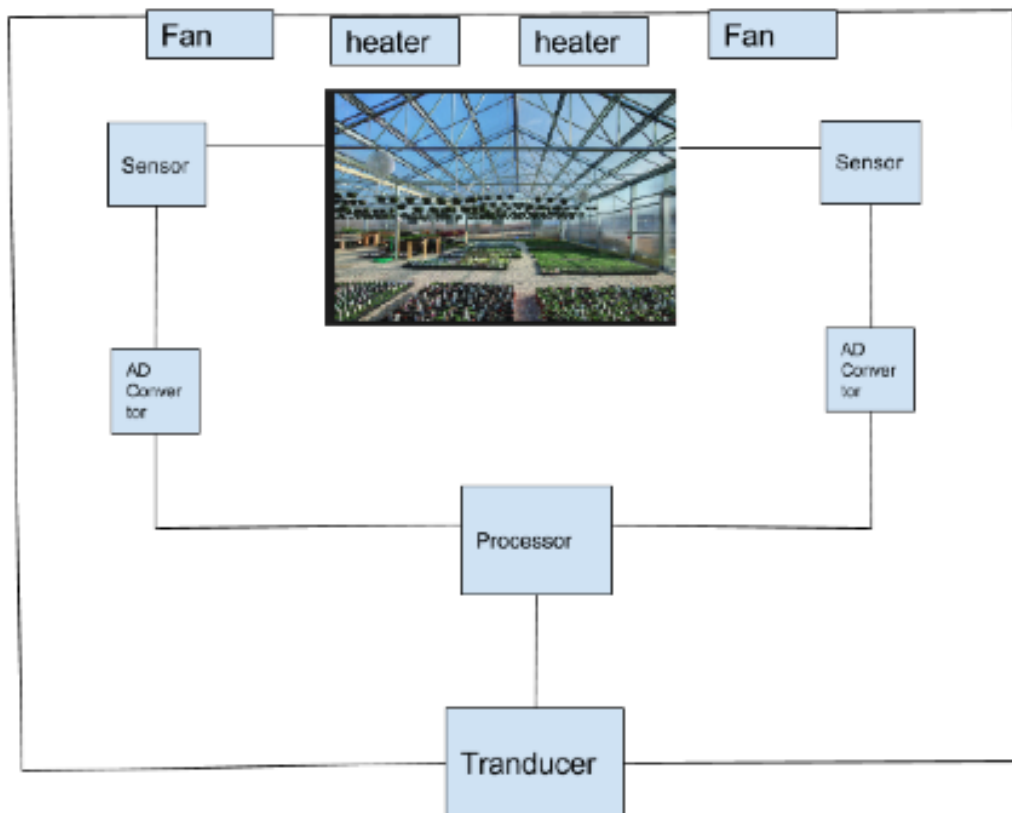


Once the digital signal reaches the processor an algorithm is followed to make decisions based on the data given. For example if a temperature of 20 degrees is read into the system the processor then has to decide whether to take action or not. If the algorithm decides that an action needs to happen then this is passed onto the transducer. The transducer activates the correct mechanical system and the process is completed.



A worked example of a green house is shown below. If the greenhouse gets too hot then the windows will be opened and the heaters switched off. If the greenhouse gets too cold the heaters will be turned on and the windows closed. Otherwise nothing happens.

A Greenhouse



Embedded Systems



The above items all have one thing in common. They all contain embedded systems. These are dedicated operating systems with specialized instruction sets that help the systems use sensors to make decisions.

In this course you need to be able to discuss embedded systems within items. Some examples are shown below:

In car - satellite navigation system



Inputs: Satellite Signal

Transformation: Satellite signals from different satellites are sent to the system in the car. Calculations are then carried out to plot the position on the on screen map.

Processing: The A*Algorithm - which uses a lot of charts to determine the current position of the car and the intended position of the car to determine the best route. It must also keep secondary route information in cache just in case the driver goes off course.

transducing: the best route loaded into cache

output: map on the screen and voice output

Dishwasher:

Inputs: Multiple inputs; hatch

Transformation:

the system can read it

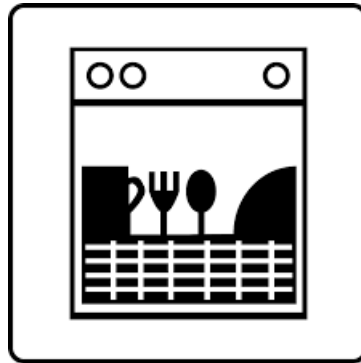
Processing: Takes in the

orders the system what

Transducing: Activates

time for this cycle

Output: The system starts to run once the hatch is shut



Cycle, time, sensor for shutting of

Transformed into 1s and 0s so that and apply the best fit.

1s and 0s and then applies and to do based on what the input is. the appropriate cycle and sets the

www.pixabay.com



Car

Inputs: Press the acceleration pedal

Transformation: Transforms the amount of pressure put onto the pedals into binary code (1 and 0).

Processing: The system reads the binary and detects how much the car needs to accelerate by.

Transducing: More fuel is put into the engine and all

the required settings are adjusted.

Output: The engine goes faster which makes the car accelerate.

Vacuum:

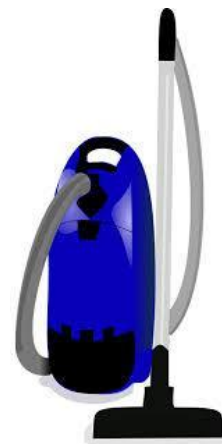
Inputs: time, power, sensor feedback

Transformation: Data is transformed to binary so that the computer can understand it and perform the task

Processing: The 1s and 0s are transformed into commands that are sent to the system

Transducing: The power is set with a certain input time, when it is near a wall, it turns

Output: The vacuum starts going around as soon as it is activated by the user

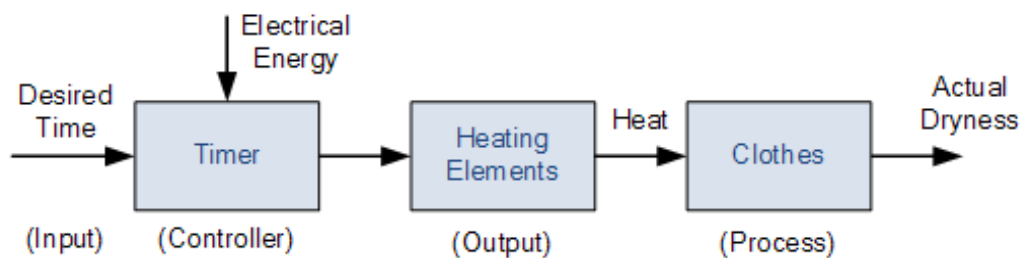


www.pixabay.com

Loop Systems

Open Loop Systems

An open loop system is a system that does not make use of feedback. For example tumble dryers. If you put clothes into a tumble dryer it will continue to dry until the timer stops. There will be no feedback about the dryness of the clothes at any point.



In Pseudocode the above diagram could be shown as the following:

INPUT time

Time on controller set

Timer started

Element heated

OUTPUT clothes

The key points of an open loop system are:

- There is no comparison between the values that come out of the system and what we want to come out for the system.
- There is no self regulation of the system
- If there is a problem / disturbance in the system then the control can not process this, it must be dealt with manually

Therefore user of open looped system must be very skilled as they have to understand how to optimize the system manually to get the best results.

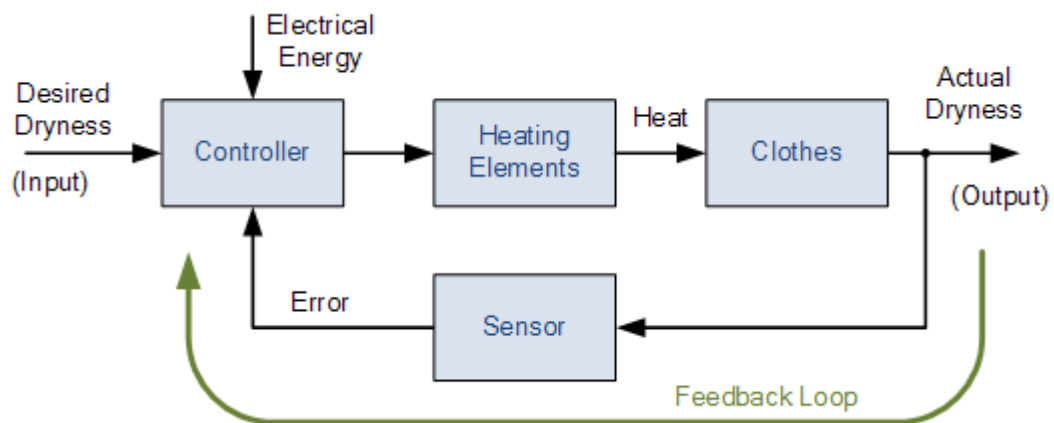
Open Loop Systems are best for the follow systems:

- light switches
- toasters
- sinks
- hobs
- remote controls
- clothes dryers
- volume on stereos



Closed Loop System

A closed loop system is a system which makes use of feedback. If we use the same example before, that of the tumble dryer. If you select the dryness level that you want and there is a sensor in the system then the system will keep drying until the clothes reach the desired level of dryness.



In Pseudocode

INPUT dryness

Program on controller set

Heating element started

Humidity measured

If humidity ? min

OUTPUT dry clothes

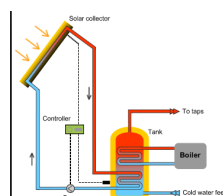
The key points of a closed system are:

- the errors are reduced as there is an automatic adjustment of the system
- the system is enhanced against external disturbances
- the system produces a reliable and repeatable performance

The users of the system have no requirement to be as skilled as the sensors and feedback do most of the work. It is the programmers of the system who are responsible for optimizing performance not the operators of the system.

Closed Loop Systems are good for the following:

- Heating systems
- Refrigerators
- Cookers
- Air conditioning systems
- ABS systems in cars
- Auto pilot



Distributed vs. Centralised Systems



All systems that use sensors require control panels that can be used to control the outputs of the system. The control panels are used to set the variables, the actions that should occur when certain conditions are met and changes to the system.

There are two different types of systems:

Centralized - in a centralized system all of the rooms/areas are controlled from one panel. The advantages of this are that you only have to input the variables once, the costs are lower as you only have one control panel to set up. A disadvantage to having a centralized system is that if the system breaks you lose functionality to all rooms/areas and each room has to have the same variables you can not set different variables for each item.

Distributed - in distributed systems all of the rooms/areas have their own control panels. The advantages of this are that you can set different variables for each area. If a system breaks you still have several other systems to call upon. The disadvantages are that it is more expensive to set up as you have more equipment to buy and it takes longer to set up the different areas.

Autonomous Agents

Autonomous agents are :

“Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires”
(Wikipedia)

Autonomous agents in control systems are entities that carry out actions based on the feedback of the sensors. They do this without interference of the end user. The only action the end user has is to set the initial variables.

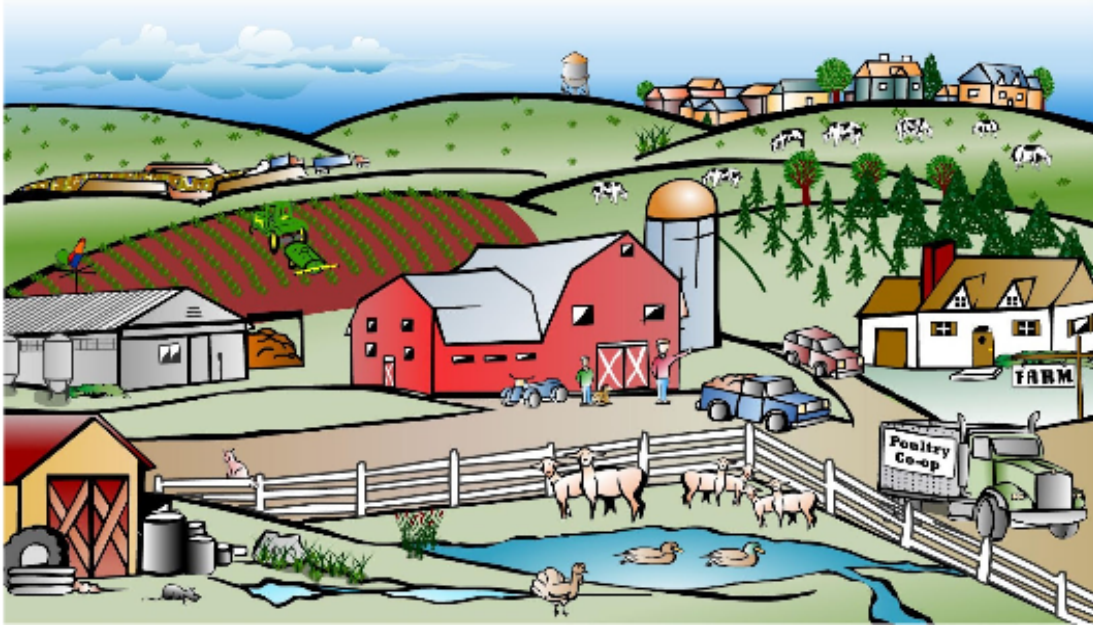
In this course you need to be able to for any given scenario discuss the advantages and disadvantages of using autonomous agents as well as the connection media required. One example that was used in the past was that of a farm.

Farm Example:

As you can see the farm has many different animals and many different buildings. There will be a system in place to heat / cool the barns as necessary and also to identify when animals need attention.

The questions to consider are:

- Evaluate the use of both a centralized and a distributed system and make a decision about which would be best for the farm
- Discuss the welfare of the animals with regard to using sensors to make decisions
- How could a mobile control panel be useful to the farmer?



In the farm it is clear there are several types of animals who need looking after. If a centralized system was used the farmer could control all the barns from one system, however, it would mean that all animals would need the same conditions. In a distributed system each barn would be able to have different conditions, as this is a farm and it is likely the animals would need different conditions even though it is more expensive and more tricky to manage I would recommend a distributed system

Using sensors to manage the animals could be useful, sensors would quickly be able to tell if the barn was too hot or too cold and adjust the machinery in the barn accordingly. If there were sensors alerts could be sent to the farmer if food or drink needed replenished. The mobile control panel could be useful to the farmer as the if they are away from the main area they still have access to the controls. This means the are able to respond to any alerts immediately.

Utilitarianism

"It's too dangerous. You can't have a person driving a two ton death machine."

- Elon Musk (March 2015)

It is important as technology is progressing to think about what Musk claims. He claims that cars will become so safe that it will be difficult to get insurance if you drive your car but this means that algorithms will have to be written to drive the car and will have to be written to mimic the instincts of drivers so where does that lead us?

Cornell university have been conducting a survey of the general public with the vision that cars should be programmed to act in the manner that people expect them to. Which leads to the following question:

"In the following situations who should the car kill?"

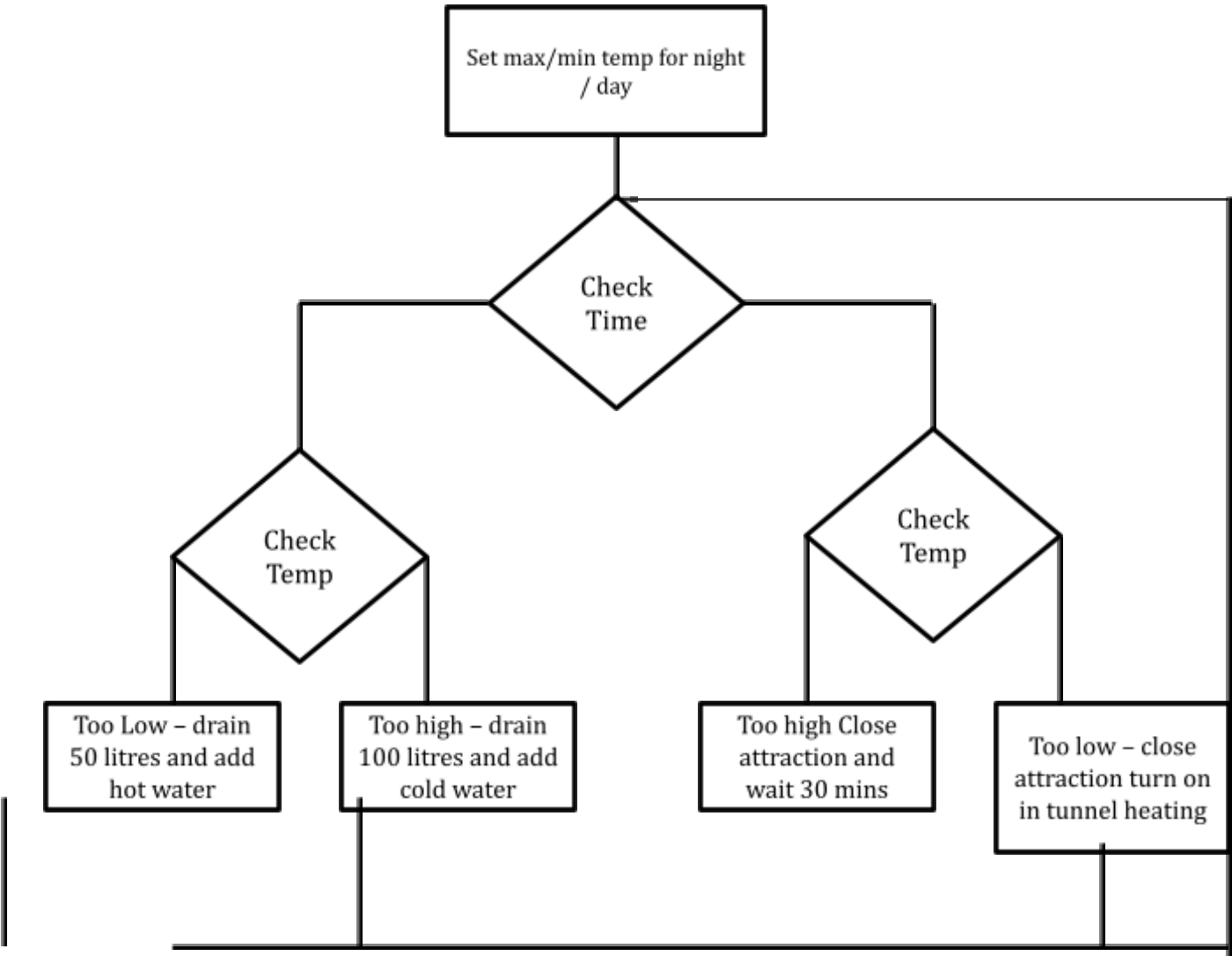
Specifically in the following situations do we injure / possibly kill one person over the group of people, how do we decide which life is more important?

Chapter Six - Review Questions

In Konstanz there is a sea life center containing over 3000 animals in 30 different tanks. The largest being the ocean tank containing nurse sharks and turtles. Each aquarium has to be kept at a regulated temperature in order to keep the animals healthy and happy.



The flowchart for the aquarium is shown below:



1a: Identify one situation in which the system should respond to:

Input from the user
Temperature

b: State three hardware devices that are needed to process and capture the data.

c: Construct Pseudocode for the algorithm outlined on the previous page.

d: The company currently uses a manual system for the heating system. Employees take the temperature of each tank using a thermometer and then manually carry out the processes required to regulate the temperature.

Evaluate the economic impact of using a computerized system
Discuss the social impact of the computerized system for the employees

e: The company decides to move to a computerized system. This will allow them to control the temperature on a more frequent basis as staff are often too busy to be effective. They are unsure whether to have all 30 tanks on one system or have separate systems. Evaluate the decision of having all 30 tanks on separate systems and make a recommendation to the company.

2: In all areas of the home you can find systems that are controlled by technology.

Identify two systems in the home that make use of technology

For one of the systems identified above identify two sensors that are required.

Explain why feedback is required in any given system.

Explain what would happen in one of your identified systems if there were no feedback.

Chapter Seven - Programming - Standard Level

This section of the book incorporates **Computational Thinking**
(the connection to programming)

Essential Understanding: “Computer despite their perceived intelligence are in fact just millions of lines of short statements that collect, compare and save data.”

Key Questions:

Can I create a program on a machine that people will find entertaining?

Where do the programs I run on my machine come from?

Are high level languages an example of a true global language?

Can all actions on a computer really be dissected into small calculations?

ATL Skills:

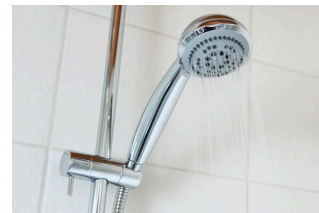
Self Management –
demonstrate persistence
and perseverance.

Thinking – Create novel
solutions to complex
problems.

What is Programming?

Computers are essential dumb machines, they have to be told very carefully and very simply what to do. An example of this would be when trying to teach a small child how to shower, the conversation might be similar to the following:

- 1: walk into the bathroom
- 2: turn the shower on
- 3: adjust the temperature of the shower
- 4: stand under the shower
- 5: shampoo your hair etc...



But what happened? The child still has their clothes on! That's because we didn't say remove your clothes at any point in the instructions. As with small children you can not make any assumptions with computers you must make everything clear.

Fundamentally computers can only carry out four functions. They can carry out simple calculations (add, subtract, multiply and divide), they can compare data, they can retrieve data and they can store data. Combining these fundamental operations allow us to complete more complex tasks. A simple example of a compound operation is find the largest. To complete this task the processor inside the computer needs to perform the following tasks:

- 1: Retrieve the data values from memory
- 2: Store the data in a linear fashion
- 3: Store a variable called largest and set data item one as the largest
- 4: Compare the next value in the list to data item one
 - 5: If the current item is largest than the value in largest
 - 5.1 :Store data current data item as largest
 - 5.2: Else
 - 5.3: Leave current largest value as largest
- 6: repeat steps 4 and 5 for the rest of the data items
- 7: retrieve largest

Through the use of compound operations we can create complex programs for use. In order to create complex programs we use high level languages. Software instructions that allow us to create programs. There are two types of programming language types. General purpose programming languages and specialist programming languages.

General purpose programming languages allows us to create programs suitable for a number of tasks. Examples of General purpose programming languages include: Java, C++, Basic and Python. Examples of specialist programming languages include: BANCStar (a financial application), HTML (for authoring webpages), FORTRAN (for scientific data) and SQL (for manipulating databases).

High Level Languages

Computers only understand 1's and 0's and as humans although we could if we tried incredibly hard could learn how to program like this, it would take us many many years and we would not be able to create complex code. This is why high level languages were invented. This allows us to create complex code in a relatively easy fashion. Common features of high level languages include fixed vocabulary, unambiguous meaning, consistent grammar and syntax. Unambiguous meaning is especially important in programming as it allows us to ensure the program will do what we want them to do. However if we understand high level languages and computers understand low level languages how do we write the programs?

Translation

There are three levels of computer languages and these are shown in the following table:

Name	Description	Portability	Difficulty to learn
Machine Code	Only understood by the hardware it is compiled for.	None – you can not put a PS3 disk into an Xbox and expect it to work	Very difficult
Assembly	Low level language that can be compiled into the machine code for each hardware	Portable between similar platforms	Difficult
High Level	Understandable by humans but not computers.	Highly portable as used in the programming software	Relatively easy

High level languages need to be translated into programs the computer understands. This is done in two ways.

Stage One: During development a translator is used to translate the program line by line into machine code. Similar to when speeches are being translated in real time between two languages. This can take a long time however the advantage of this process is errors can be pinpointed and fixed easily by the programmer. To run a program using a translator you need the source code and the debugging program.

Stage Two: After development a compiler is used, it changes the program from a high level language and into machine code. This is done all at the same time, the same as if a person translated a whole document using an online translate tool. It can work without using the translator however it is very difficult to find mistakes. The compiled program is then an executable file that can be used stand alone on a machine.

This is quite a difficult concept to understand so here is an analogy using a car:



Object Orientation

What is it?

In the past programs were written procedurally the computer viewed and executed the code line by line, as programs became more complex this led to very long and complicated programs. The result is now we have object orientated programming, each object is a separate item which can be manipulated with its own behaviours. This allows us to make complex programs easier.

An easy way to think about object orientation is a class is a noun and the method is the verb. An object is like a living creature it has knowledge inside of them telling them how to do things, they can remember things. Instead of interacting with them on a low level we interact at a high / abstract level.

Example: If I am your air ticket object and you send me the message “buy me return tickets to Rome this weekend”. I know where to look for cheap tickets, I know where Rome is and how to get them. I buy your tickets, I print them off and I hand them to you. You have no idea how I did it, however I have your tickets.

This is how objects and methods work, I can ask my object to do things, I don't know how it does these things but it does and the result is I get what I want.

What is a class?

We start any program with a set of classes. Classes are the blue prints for the objects we create. Our program will always have a main class which is the control class for the program and will most likely have at least one other class which will be used to create an object we can manipulate.

One example of a class is a bicycle class. Imagine we wanted to store details about all the bikes stored in a school. We would need a bike class that stores our knowledge of what a bike should have and do. We know that they speed up and slow down, this would be represented in Java using the following code:

```
public class bicycle{

    // the bicycle has two instance variables
    private int gear;
    private int speed;

    // there is a constructor class which allows us to make
    // a bicycle object we can manipulate

    public bicycle (int startGear, int startSpeed){

        gear = startGear;
        speed = startSpeed;
    }
}
```

```
// the bicycle class has three methods

public void setGear (int newGear){

    gear = newGear;
}

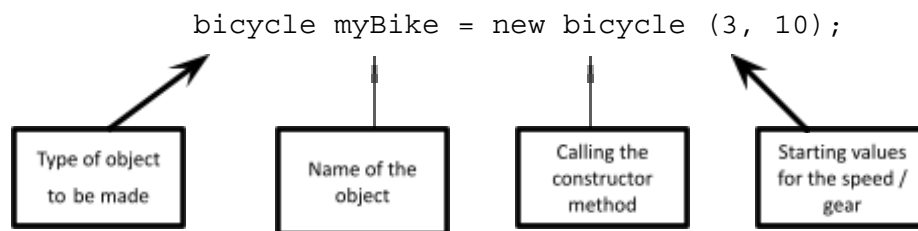
public void applyBreak (){

    speed = speed - 1;
}

public void speedUp(){

    speed = speed + 1;
}
}
```

At the moment our bike will do nothing as it is only an idea. Its only when we create an instance of our bike with a start speed and a starting gear do we have something tangible we can manipulate. To do this we have to call the special constructor method from the method that is running our program. The constructor method is the methods that “constructs” the object. In Java it looks similar to this:



Check your understanding (theory):

- 1: What is the difference between a class and an object?
- 2: What is a constructor method?
- 3: Using the example above explain what the numbers between the () are.
- 4: Think about a dog, what instance variables and methods would you need to create a dog?

Check your understanding (practical):

Critters are little creatures that can be adopted and grown into pets. They are brightly coloured creatures and each has a special power. They eat and drink like regular animals but they exercise by flying. An example critter is shown below:

Name: Katie
Colour: Blue
Special Power: Invisibility
Eats: popcorn
Drinks: Mountain Water



Implement the Critter class in java and use a main method test class to check it works.

Inheritance

It is clear from the above example that although classes are very useful they do not allow us to specialise items in the way that we would like. An example would be there is no such thing as a bicycle, there are several different types of bicycle you could have a road bike, a mountain bike and a racing bike. Although they have similar properties they also have very different properties. Java allows us to deal with this using a concept called inheritance. We can put all of the common variables and methods for each class in a special class called a super class. Variables and methods specific to each type of item can be placed in the subclasses. The advantages of this are:

- If one of the common methods needs to be changed it is only required to be changed once
- There is less duplication of code as the common methods are only programmed once
- Reusability, the classes can be used in many programs

The disadvantages are:

- you have to use both the super class and sub classes in any code
- it take additional time and effort to use inheritance
- maintenance is harder as you need to change the sub class and the superclass
- if you delete the superclass the subclass will fail.

How would this look in java?

First of all we would have our superclass with the common methods for bikes (speed up, slow down and change gear)

```
public class bicycle{
    // the bicycle has two instance variables
    private int gear;
    private int speed;

    // there is a constructor class which allows us to make
    // a bicycle object we can manipulate

    public bicycle (int startGear, int startSpeed){
        gear = startGear;
        speed = startSpeed;
    }

    // the bicycle class has three methods
    public void setGear (int newGear){
        gear = newGear;
    }

    public void applyBreak (){
        speed = speed - 1;
    }
    public void speedUp(){
        speed = speed + 1;
    }
}
```

```

    }
}

```

We would then need to have additional classes for each of the specialized bikes, these are called sub classes.

For example we have the mountain bike subclass

```

public class mountainBike extends bicycle {

    public int suspension;

    public mountainBike (int startSuspension, int
startGear, int startSpeed) {

        super (startGear, startSpeed);
        suspension = startSuspension;
    }

    public int setSuspension (int susValue) {

        suspension = susValue;

    }

}

```

The variable that is
unique to the
mountain bike

The constructor method
has the super and sub
class start values

The variables for
the superclass are
sent to that
constructor

This method is
only for the sub
class

For example we have the racing bike subclass

```

public class racingBike extends bicycle {

    public int tyreStyle;

    public racingBike (int startStyle, int
startGear, int startSpeed) {

        super (startGear, startSpeed);
        tyreStyle =startStyle;
    }

    public int setTyre (int tyreValue) {

        tyreStyle = tyreValue;

    }

}

```

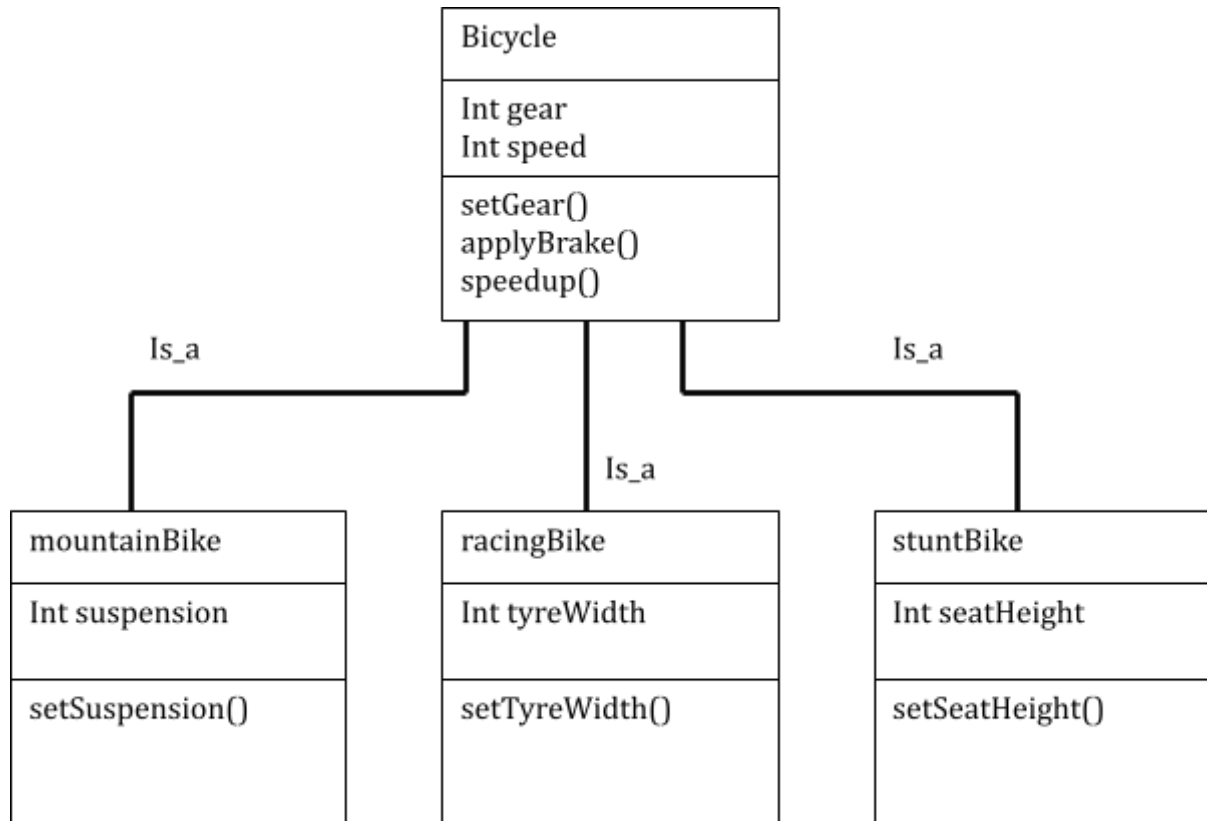
The variable that is
unique to the
racing bike

The constructor method
has the super and sub
class start values

The variables for
the superclass are
sent to that
constructor

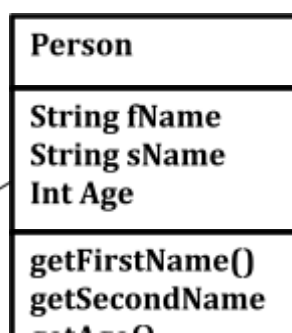
This method is
only for the sub
class

When planning our bike program we would create a class diagram, this would look similar to the following:



Check your understanding - Practical

Using your knowledge of inheritance implement the following program in Java, use a main method to check that it works.



Parameters

Parameters are used to pass information between the classes used in a program. Without the use of parameters our programs would all have to be in the one class which defeats the purpose of object orientated programming. Parameters allow us to pass information between the methods in our classes. We use parameters in several ways throughout the programming process.

There are two types of parameters:

Formal Parameters – the information and type of information the method expects. These are declared in the method signature e.g.:

```
public void speedUp(int increment)
```

This method when called is expecting an integer value in the brackets

Actual Parameters – when you call the method, the actual parameters are the information you send into the method to make it work e.g.:

```
myBike.speedUp(10)
```

This is the data being sent in the method. The method will then do something with this data. It could also be a variable

One of the first ways we use parameters is with constructor methods. The parameters in between the brackets are the initial values we are using to create our object. It is important in all cases we send the information in the correct order, otherwise it will not work correctly.

Using the example above we would call our constructor (to make our bike) with the following declaration:

```
mountainBike myMountainBike = new mountainBike (2,1,10);
```

This line of code will make a new mountain bike object that we can manipulate with a start suspension of 2, a start gear of 1 and a start speed of 10. We can adjust these numbers using methods, sometimes methods use parameters for example we can speed up the mountain bike using the following method:

```
myMountainBike.speedUp(10);
```

we are passing in the actual parameter of 10, we know therefore that when we invoke this message the speed of the bike is increased by 10.

Check your understanding:

- 1: What is the difference between a formal parameter and an actual parameter?
Use screen shots of previous code to explain.
- 2: Why are parameters required in constructor methods?
- 3: Why is the order in which the parameters are passed to the class important?

Check your understanding (practical):

Write a simple calculator program – when run it should add, subtract, divide and multiply two numbers together. It should have a calculator class where all the hard work is completed and a main method class to run the program.

Ext. use IF statements and user input to make this a useful calculator

Variables

Variables are used to store data while the program is running, for example it might store user input which is then used in an IF statement. It could be used to store the result of a calculation which is then displayed to the user. It is important to remember that the value in our variables will only be held while the program is running.

Variable Types

There are two types of variables primitives and objects. Primitive data types can only store data there are no methods associated with them e.g. integer, double, character. Object data types have their own methods associated with them e.g. the string object has a length, charAt etc. methods. When you create your own classes you are creating types of object variables that can be used by programs.

Primitive Data Types	
Integer	A whole number the is between -2^{31} and 2^{31}
Double	A floating point number that used 64-bits
Boolean	A data type that stores true or false (1 or 0)
Char	A single 16 – bit Unicode character
Long	A whole number that has a value between -2^{63} and 2^{63}
Short	A whole number that has a value between -32 768 and 32 767

Object Data Types	
String	A collection of characters that has a number of methods associated with it such at .length() .charAt(1) etc.
Array	An array is a container that holds a set number of a single type of variable (primitive or object)

Declaring Variables

Before a variable can be used it has to be declared, this means telling the program that you would like to set aside an area in memory that you will refer to with a set name, and the data type you are going to store in it. Sometimes you can set an initial starting value to the variable.

```
String username;
Double cost = 5.99;
Int age = 15;
Boolean answer;
Double newBalance = cost * 0.1;
Cow myCow = new Cow(12,1)
```

Strings

Strings are important variable types for any program. They are a collection of alphanumeric characters. As Strings are object variable types they have several methods available to use with them. The Methods available are:

Accessor Methods	Mutator Methods
length() charAt() getBytes() getChars(l,j, target[], xstart) substring(l,j)	split(string, delim) concat(g) replace(x,y) toLowerCase() toUpperCase() trim()

Scope of Variables

Depending on where the variable is declared, depends where the value can be seen and used in the program.

```
Public Account () {
```

```
    Private double balance;
    Private double interest_rate;
```

These are instance variables required throughout the class

```
    Public account (double startBalance, double startInterest_rate)
    {
```

```
        balance = startBalance;
        interest_rate =
        startInterest_rate;
```

These are parameter variables, values sent from other class to be used here

```
    }
```

```
    protected double getInterestBalance (double startBalance) {
```

```
        private double interestCalc;

        interestCalc = interest_rate * balance;
        balance = balance + interestCalc;

        return balance;
```

This is a local variable SEEN in this METHOD ONLY

```
}
```

Account, balance, interest_rate, getInterestBalance() etc. are all identifiers. **Identifiers** are the names of the classes, methods, variables in java. It is how they identify themselves in the program.

Constant Variables

Constants are variables that do not change their value throughout the program. Their value should be set at the time of creation. For example if you were creating a program that was calculating the VAT of an item you know that it is unlikely the rate of VAT will change. Therefore you would set it as a constant value. Naming conventions in Java show constant variables in capital letters. E.g.

```
Double VAT_VALUE = 20.0;
```

Casting

Sometimes we have to force our variable into a different variable type this is achieved through casting. For example people can not take 0.25 out of their bank account therefore the amount they are withdrawing would be rounded up to an integer. This is achieved using the following.

```
(int) (balance * 100) – this forces the result to be an integer  
rather than a double
```

Relational operators for primitive variables

Relational operators are usually used in IF statements to help make a decision. These operators only work with primitive data types:

Operator Java	Description
>	Greater than, testing to check the value on the left is greater than the value on the right.
>=	Greater than or equal to, testing to check whether the value on the left is greater than or equal to the value on the right
<	Less than, testing to check whether the value on the left is less than the value on the right.
<=	Less than or equal to, to check whether the value on left is less than or equal to the value on the right
==	Equal to, to check whether two values are equal to each other
!=	Not equal to, to check whether two values are not equal

Relational operators for String variables

The following relational operators are very useful for comparing Strings.

Operator Java	Example Use	Description
<code>.equals()</code>	<code>If(string1.equals(string2))</code>	To check if two strings are equal however unless they are exactly the same with upper and lower case letters this will not work.
<code>.equalsIgnoreCase()</code>	<code>If(string1.equalsIgnoreCase(string2))</code>	To check if two strings are equal but this will ignore the case so even if they have different upper and lower case letters it will work
<code>.compareTo()>0</code>	<code>String1.compareTo(string2)>0</code>	To check if string 1 comes before string 2 in the alphabet.
<code>.compareTo()<0</code>	<code>String1.compareTo(string2)<0</code>	To check if string 1 comes after string 2 in the alphabet.
<code>.compareTo()==0</code>	<code>String1.compareTo(string2) ==0</code>	To check if string 1 and string 2 are the same alphabetically.

Boolean Operators

These are useful for ending loops && means and so both conditions have to be met. || means of, either condition must be met to end the loop. Examples of use are shown below:

Until (amount == 35 || userInput.equals("g"))

One condition has to be met

Until (amount ==35 && cycles <=100)

Both conditions have to be met

Check your understanding:

1: Print a copy of your last program.

- highlight in one colour all the local variables
- highlight in another colour global variables
- identify different variable types used
- highlight where Boolean comparisons (true / false) have been made

2: Create a simple program that will ask for my first name and second name. It should then

- take the first 3 letters of my first name and concatenate them into the last 3 letters of my surname and store and print this in a new username variable.

Methods

Methods are the pieces of code that will allow are programs to function in a useful manner. Without methods our programs will perform nothing of interest. The syntax of methods is very strict, if you do not follow the syntax correctly then you will experience problems with your programming.

Method Signatures

Methods signatures are very useful as they tell us what the method expects as parameters and what the return value will be.

e.g.

```
Public double calculateInterest(double interest_rate, double
                                amount_saved)
```

If we break down the signature it tells us the following:

Public – it can be accessed outside the class in which it was written

Double – it will return a double value to the class which calls it

calculateInterest – is the name of the method, this is how it will be identified

(double interest_rate, double amount_saved) – shows that it expects two doubles, they must be sent in this order otherwise the method will not function correctly

it would be called from another method using a line of code similar to the following:

```
myAccount.calculateInterest(interest, savings);
```

Return values

The return value is the value that is sent back to the method that called the method. The type of data being returned must be declared in the method declaration. Examples are shown below:

```
private double calculateWage (int hours, double
                                hourly_wage)
```

in this example a double will be returned

```
Public int getArea (int height, int width)
```

In this an integer value returned

```
Public string getName()
```

In this example string value is returned

```
Public void setInterest ()
```

In this example no values are returned

Constructor Methods

When we create a class we have a “blueprint” for an object but the object does not actually exist. To construct the object we need to have a constructor method within the class.

For example in our main method we will have:

```
Account myAccount = new Account(1000);    // this line asks
the account class to make a new object
```

The method that construct the Account object my account is the constructor method it looks like:

```
Public account (double startBalance){

    Balance = startBalance
```

Accessor Methods

When using encapsulation all instance variables should be set as private, this means they cannot be “seen” by other objects. However not being able to see the contents of the variables would render most of our programs useless. Therefore accessor methods are create which allow private variables to be seen by other objects. Below is some examples of accessor methods.

In the person class the code would look like this:

```
Public String getFirstName (){
    Return firstName
}
```

in the object using the person class the code would look like this:

```
username = person.getFirstName()
```

Mutator Methods

When using encapsulation all instance variables are private therefore we can not access them outside of the object. However in most cases we need to be able to set or change the contents of the variables inside the object. This is when a mutator method is used.

Inside the person class the code would look like this:

```
Public string setFirstName (String iniFirstName) {
    firstName = iniFirstName;
}
```

In the object using the person class the code would look like this:

```
Person.setFirstName (iniFirstName);
```

Reserved Words Associated with Methods

Public/private and protected modifiers determine which object can see the methods of a class.

This is shown in the following table:

Modifier	Class	Package	Subclass	World
Public	Y	Y	Y	Y
Protected	Y	Y	Y	N
private	Y	N	N	N

Public

When you declare a method as public it can be seen by all objects everywhere and can be used by all objects everywhere.

Private

Private methods can only be seen inside the class in which they are declared.

Protected

Protected methods can be seen by their class, the package they are declared in and their subclasses.

Extends

Extends is used with inheritance to show that the object inherits the methods from the super class as well as having its own methods.

Static

Using static means you do not have to create an instance of something as it is available for use in the class without having to explicitly declaring it.

Example:

```
Public class Account [
    Public static void getAccNumber () {
        Return accountNumber;
    }
}
```

instead of saying

```
Account a = new Account ();
a.getAccNumber();
```

you can say

```
a.getAccNumber()
```


Check your Understanding:

1: Print one of your programs:

- highlight the constructor method
- highlight accessor methods
- highlight mutator methods

2: What information can you gain from a method signature? How will this help you in your programming?

3: What is a return value and why does it help with programming?

Encapsulation

Encapsulation is an essential concept of object orientated program. It means that all of the components of the class are kept together in one place. This allows the class to become a single entity that can be used in other programs.

The advantages of this would be:

- the class can be created and pretested
- different classes can be programmed by different members of the programming team concurrently therefore saving time
- the complexity of the class can be hidden from the user, the user can make use of the class without the need to understand it
- the class can be used in other classes without the need for reprogramming

For programming it is also very useful as you can use the premade classes such as String, Math, Scanner etc. without understanding how they work. We can import them make use of their methods by providing the correct parameter values and understanding the return value we will get but we don't understand the inner works.

A non programming example of this would be a car:



When I put my foot on the accelerator I am invoking the speed up method.



What happens under the bonnet is a mystery to me!

However the car goes faster which is the return I wanted

from my action.

So how do we get true encapsulation?

Instance variables should be declared as private as this ensures they can not be accidentally changed from outside the class. They can only be purposefully changed using the correct mutator method.

The correct method declarations should be used, only methods that can be used outside the class should be public e.g. mutator and accessor methods, the rest should be private.

```
Public Account () {  
  
    Private double balance;  
    Private double interest_rate;  
  
    Public Account (double startBalance){  
  
        Balance = startBalance()  
  
    }  
  
    public double addFunds (double inAmount) {  
  
        balance = balance + inAmount;  
        return balance;  
    }  
}
```

Check your Understanding:

Pixabay.com

We are going to implement a car program using encapsulation. I would like you to design the UML diagram (what variables and methods a car should have. Stick to around 3 or 4 variables otherwise it will get complex). You should then create a main method to test your car class.

Libraries

Libraries make use of encapsulation and are collections of classes that can be reused by other classes throughout the program. They have several advantages:

- They save time as they are preprogrammed and pretested
- They can be used by people who are not as experienced at coding
- You do not need to understand them

The Math class is one of the most useful libraries as it has lots of preprogrammed maths functions that can be used. The only thing you need to know in order to use it is the method name and the parameter required.

The Java API page has a list of all of the library classes that are available for use in Java, this can save a lot of time and energy. An example of a program that uses libraries is shown below. In this program it will roll the dice until snake eyes (two ones) are thrown. It will then report the number of rolls taken to reach this.

```
public class PairOfDice {

    public int die1;    // initialize first dice
    public int die2;    // initialize second dice

    public void PairOfDice() {
        roll(); // Call the roll() method to roll the dice.
    }
    public void roll() {
        // Roll the dice by setting each of the dice to
        // a random number between 1 and 6.
        die1 = (int)(Math.random()*6) + 1;
        die2 = (int)(Math.random()*6) + 1;
    }
}

public class Main {

    public static void main(String[] args) {

        PairOfDice dice = new PairOfDice();
        int rollCount = 0;

        do {
            dice.roll();
            System.out.println("The dice come up " + dice.getDie1()

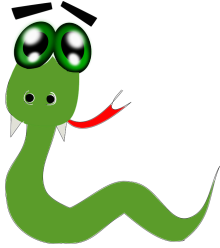
                                + " and " + dice.getDie2());

            rollCount++;
        } while (dice.getTotal() != 2);

        /* Report the number of rolls. */

        System.out.println("\nIt took " + rollCount + " rolls to get a
                                2.");
    }
}
```

}

Check your Understanding

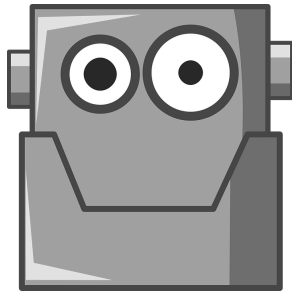
1: Implement the above snakes eyes program in Java. How does this program make use of both libraries and encapsulation?

Pixabay.org

2: Using the techniques learned in the paper, scissors game.



above develop a rock,



3: Use the internet to find the methods available in the math class. Use this to produce a complex calculator.

Pixabay.org

IF Statements

IF statements are very important in programming all programs need to be able to make decisions.

The statements follow a similar pattern

```
IF (condition) then
    Do this instruction
IF (condition) then
    Do this instruction
ELSE
    Do this instruction or an error message
```

An example of an IF statement would be to modify the main method to ask the user which bike they would like to purchase / use and then create a bike based on their choice.

The main method would then look like this:

This code
get the info
from the
user

The if statement
what to do is in the
{

```
package bicycletester;
import java.util.Scanner;

/**
 *
 * @author lindseystephenson
 */
public class BicycleTester {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here

        // create a variable to store the user input
        String userInput = new String();
        // create new scanner to ask for user input
        Scanner userChoice = new Scanner(System.in);
        //use the pane to ask for user input
        System.out.println("which bike would you like? Racer, Mountain or Stunt");
        // store what the user types in into the variable user input
        userInput = userChoice.nextLine();

        if(userInput.equals("Mountain")){

            mountainBike myMountain = new mountainBike(10, 5, 20);

            System.out.println("I am a mountain bike and ");
            myMountain.setGear(8);
            System.out.println("I am a mountain bike and ");
            myMountain.applyBrake(1);
            System.out.println("I am a mountain bike and my suspension is "+myMountain.setSuspension(4));
        }

        else if (userInput.equals("Racer")){

            racingBike myRacer = new racingBike(4,4, 40);

            System.out.println("I am a racing bike and ");
            myRacer.setGear(8);
            System.out.println("I am a racing bike and ");
            myRacer.applyBrake(1);
            System.out.println("I am a racing bike and my start tire is "+myRacer.addtireWidth(4));
        }
    }
}
```

Else statement what
to do if all else fails

```

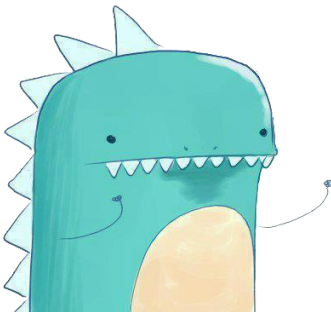
else if (userInput.equals("Stunt")){
    stuntBike myStunt = new stuntBike (20, 2, 10);

    System.out.println("i am a stunt bike and ");
    myStunt.setGear(2);
    System.out.println("i am a stunt bike and ");
    myStunt.applyBrake(2);
    System.out.println("i am a stunt bike and my seat height is"+myStunt.setSeatheight(-3));
}

else
{
    System.out.println("you have not made a valid choice chicken");
}
}
}

```

Check your Understanding:



1: Open up buzzfeed.com and you will find a number of quizzes designed to tell you what character you should be in a tv show, who your spiritual artist is or what animal you were in a past life. This works by counting up points assigned to answers. Depending on the number of points you have depends on the outcome of the quiz.

Make a quiz that will help me to determine what dinosaur I was in a past life (or if you have an alternative idea feel free to use it).

2: Another game that can be made with IF statements is a text based adventure game. In the 1990's they tried to make a virtual reality show called 'Knightmare', we are going to make an even worse version.

How it works:

1: Give the user a question and 2 or 3 options. E.g. "You walk into the banquet hall and you are faced with a talking wall. What do you do 1 – run the other way, 2 – talk back with an attitude, or 3 – answer his riddles in a suave style."

You then use the answer and use if statements to decide what happens next.

e.g. IF input == 1

print "you ran the other way what are you a chicken? What do ye now? 1 – the door is locked scream and hope someone rescues you, 2 – pick up the cement you see in the corner?"



```

if (choice == 1)
    print "no one came its over"
else if (choice == 2)
    print "ok walk to the wall, what happens next? 1 – the wall
    charms you? 2 – you cement up his face
    if.....
Else if (choice == 2) // this refers back to the original question

```

As you can see there are a lot of nested ifs but used well could create good game.

Loops

Loops as the name suggests repeat sections of code over and over until a condition has been met. There are two types of that exist in java.

Conditional Loops:

Conditional loops repeat sections until a condition is met. For example it might repeat until someone enters the correct password or the person has had three attempts. They can be used to repeat a program until the user presses quit. Conditional loops will repeat an unknown number times. The following is an example of a conditional loop:

The do is the start of the conditional loop

```

- */
-
3 public static void main(String[] args) {

    // TODO code application logic here
    String userInput = new String();
    Scanner in = new Scanner(System.in);
    boolean correct = false;
    int counter = 1;

    do {
        System.out.println("enter your password");
        userInput = in.nextLine();

        if (userInput.equals("HOLA")){
            correct = true;
        }
        counter ++;

    }while (counter <= 3 && correct == false);

    if (correct == true ){
        System.out.println("you cracked the code");
    }
    else {
        System.out.println("no entry");
    }
}
-
}

```

This is the condition that must be met to end the loop. The code between the do and while will repeat

It is important when using conditional loops that you get the end condition correct otherwise you will enter an infinite loop that would eventually crash the computer!

And example of someone getting in an infinite loop is demonstrated quite nicely in the big bang theory here:

<https://www.youtube.com/watch?v=k0xgjUhEG3U>

Fixed Loops:

Fixed loops will repeat a section of code a set number of times. For example if we need ten words we would use a fixed loop to ask the user to enter a word ten times. Fixed loops will always repeat a set number times.

The for loop will repeat the code in between the two brackets 10 times

```

public class Loops {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        String userInput = new String();
        Scanner in = new Scanner(System.in);
        boolean correct = false;

        for (int counter = 0; counter <=10; counter ++){

            System.out.println("enter a random word");
            userInput = in.nextLine();
            System.out.println(userInput);
        }
    }
}

```

Check your Understanding:

The 21 sticks game. In this game the pile starts with 21 sticks. Each person removes one or two sticks. The loser is the person who runs out of sticks.

The method would look similar to the following:

- 1: Declare variables
 - no_of_sticks = 21
 - userOneInput
 - userTwoInput
 - sticksLeft = true



```
2: Start loop
  2.1 prompt user one for entry
  2.2. IF input == 1
    no_of_sticks = no_of_sticks - 1
    check no_of_sticks
  ELSE
    No_of_sticks = no_of_sticks - 2
    Check no_of_sticks
  2.3 repeat above of user two

3: end loop when not sticksLeft
4: display winner
```

Pixabay.com

Arrays

If we were to store the grade of everyone in a grade 9 maths class we would required 15 – 25 variables based on the number of students in the class. E.g.

```
int classMember1;  
int classMember2;  
int classMember3;
```

This is not very efficient, it would mean hours of extra coding and the variables could get complicated.

To solve this problem there are variables called arrays. Arrays allow many variables to be stored under the same name. To think about it in a non programming way I would think of it as a named shelf that you can store information on. Arrays allow us to store data in a streamlined way and because they are objects they have associated methods we can use to make things easier.

classMarks Array



classMarks[0]

classMarks[1]

classMarks[2]

classMarks[3]

classMarks[4]

classMarks[5]

Declaring Arrays

You can declare an array in two different ways:

The first way to declare the array is to name the array and then tell the program what data you will be storing in the array straight away.

```
int [] myArray = {10,20,30,40,50,33,22,33,1,12};
```

This is useful if you know the data at the time of programming however, it does not allow for any interactivity from the user. The second and more common way to declare an array which allows for interaction from the user is to use the size declaration. We declare an array and say how many spaces we want to keep free to use, we can then fill these at a later date.

```
int [] myArray = new int [10];
```

The above code creates an integer array with 10 spaces we can then use a for loop to fill these spaces when we have the data. It is important to note that **once declared an array can not change size**.

All data can be stored in arrays, Arrays can be set to store primitive values such as integers, doubles, Booleans etc. or they can be used to store any class that you create or currently exists in the java API e.g. Strings.

An example of a program using an array is shown below. It allows the user to enter the grades for the students in a class and then calculate the average grade.

```
public static void main(String[] args) {

    int [] classGrades = new int [15];
    Scanner input = new Scanner (System.in);
    int total = 0;

    for(int i = 0; i < classGrades.length; i++){

        int myvalue;
        myvalue = input.nextInt();
        classGrades[i]=myvalue;
    }
    for (int x = 0; x < classGrades.length;x++){

        total = total + classGrades[x];

    }

    double average = total/classGrades.length;
    System.out.println ("the average grade is : " +
                        average);
}
```

Check your Understanding

1: Arrays that only contain one data type are not the most useful things ever but I would like you to implement the code above and understand what it does. You also should understand the use of the fixed loop. Explain the fixed loop here.

2: Using an array store the course choice for the class of students in our school. Count and print how many IB students there are and how many AP students there are.



PNG AND JPEG FILES 300 DPI
PERSONAL AND COMMERCIAL USE

GIFTSEASONSTORE
Classroom OWLS DIGITAL CLIP ART

3: Random haiku generator – using two arrays one with 3 syllable sentences and one with 7 syllable sentences generate haikus.

Arrays of Classes

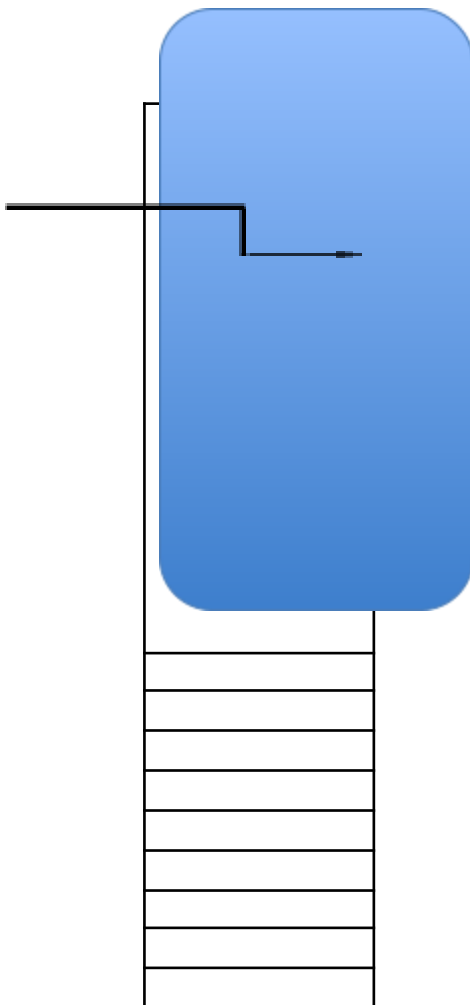
Arrays of primitive data types although useful in certain cases, such as calculating grade averages, can be quite limiting in a real world program. Similarly when we have been using classes we have only created one instance of a class which is also incredibly limiting. Wouldn't it be good if we could combine several instances of a class and store them in an array? Luckily Java allows us to create arrays of classes.

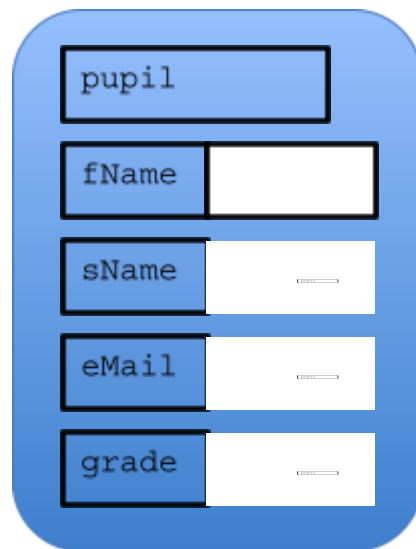
An example would be a group of students. It might be useful to store their first name, last name, email, grade and then have an array of all the students in the class. In this case a pupil class would be created and then an array of classes used to store the instances of the classes.

The array declaration would be as follows:

```
pupil [] class9J = new pupil [15];
```

Visually it might look similar to the following:





The diagram shows a blue rounded rectangle representing a class. Inside, there are five fields: 'pupil' (a single-line text box), 'fName' (a label next to a single-line text box), 'sName' (a label next to a single-line text box), 'eMail' (a label next to a single-line text box), and 'grade' (a label next to a single-line text box). Each text box has a small horizontal line at the end, indicating it is a text input field.

You can access each individual element in the classes using the code similar to the following:

```
system.out.println("the email address of "
+class9J[i].getfName() + " is " + class9J[i].getemail());
```

if you used an array of classes to store the data of the students in the class as described in the arrays section, you could use the following code to find the average grade. You could also add additional methods to find the student with the highest mark, students who are getting low grades etc. This is shown in the program below.

```

*/
public static void main(String[] args) {

    pupil [] theClass = new pupil [10];
    pupilFill myClass = new pupilFill(theClass);

    int total = 0;

    for (int i = 0; i < theClass.length; i++){
        total = total + theClass[i].getGrade();
    }
    double average = total/theClass.length;

    System.out.print("the average is: " + average);

    int lowest = theClass[0].getGrade();
    int lowestperson = 0;

    for (int i = 0; i < theClass.length; i++){
        if (theClass[i].getGrade() < lowest){
            lowest = theClass[i].getGrade();
            lowestperson = i;
        }
    }
    System.out.println("the lowest grade in the class is: " + lowest);
    System.out.println("the person with the lowest grade is: " + theClass[lowestperson].getfName() +
        " " + theClass[lowestperson].getsName());
}
}

```

The output from the program is shown below:

```

the average is: 5.0the lowest grade in the class is: 3
the person with the lowest grade is: Linzie Stephenson

```

Check your Understanding:

- 1: Open your tamagotchi cow program. Amend this program so it now stores a herd of cows. What code did you have to change?
- 2: Create a "database" of pupils in a class including their personal information and their subjects. In an ideal world this will allow teachers to view their grades and calculate their average grade.



Standard Algorithms

As you have seen in the Array of classes program in the previous section, storing and gathering information from an array is very useful. For some operations in the array there are standard algorithms that you need to know. These are the accepted best practice ways to carry out a set task.

The pseudocode and examples of the code are as follows:

Filling an Array

This code will fill an entire array at the same time. However this could be modified to add your own code to get user input etc.

```

FOR each item in array
    VALUE array [current element] = VALUE

For (int I =0; i <array.length(); i++){
    System.out.println ("what is your name?");
    String input = s.nextLine(); // s is the scanner
    array[i] = input;
}

```

Computing Sum and AverageValues

This will add the values of the array together and then calculate the average.

```

INITIALIZE total = 0
FOR each element in array
    Total = total + array [current element]
Return total

Average = total / array.length

double total = 0;
for (int i =0; i < array.length; i++){
    total = total + array[i];
}
double average = total / array.length;

```

Count Occurrences

Count the number of times a certain value appears in an array. The example below uses the word "tree".

```
INITIALIZE total = 0;
FOR each element in array
    If (array [current value] = tree)
        Total = total + 1
Return total
```

```
int total = 0;
for (int i = 0; i < array.length; i++){
    if (array[i].equals("tree")){
        total = total + 1;
    }
}
```

Finding the minimum or maximum value

This algorithm will find the highest or lowest value in an array depending on which operator you use. The example below finds the highest value

```
INITIALIZE max = Array[first item]
For each item in array
    If array[current value] > max
        Max = Array[current value]
Return max
```

```
double max = array[0];

for (int i = 0; i < array.length; i++){

    if (array[i] > max){
        max = array[i];
    }
}
return max;
```

Searching for a value

This algorithm will search an array for a value, if it finds the value it will return a true value.

```

Search term = user input
Boolean found = false
  For each element in the array
    If (current value = search term)
      Found = true
    End if
  End for
Return found

```

```

String searchTerm = in.next(); //where in is the scanner
Boolean found = false;

  For (int i = 0; i<array.length; i++){
    if (array[i].equals(searchTerm)){
      found = true;
    }
  }
return found;

```

Locate the position of an element

This algorithm will search for a value in an array and return the position of the value in the array if it is found in the array. The one problem with this code is if there is more than one instance of the item then it will only report the position of the last time it appears.

```

Search term = user input
Int position = 0
  For each element in the array
    If (current value = search term)
      Position = current value position
    End if
  End loop
Return position

```

```

String searchTerm = in.next(); // where in is the
scanner
int position = 0;

  for (int i = 0; i<array.length; i++){
    if (array[i].equals(searchTerm)){
      position = i;
    }
  }
return position;

```

Check your understanding

In the programming folder there are partially completed java programs. Open them and enter the code for the standard algorithm identified. There is pseudocode inside the class.

Two Dimensional Arrays

While one dimensional arrays allow us to store a list of items under one variable name. Two dimensional arrays allow us to store data in a matrix. One way to think about this is a table filled with data.

So far what we know about arrays is that they allow us to store information under the same variable name. We need to use loops in order to use array efficiently and they can be useful for storing data in programs. For example an array of animals could be drawn as shown below:

ANIMALS

[0]	[1]	[2]	[3]	[4]	[5]	[6]
						

The arrays we know could be considered as lines of data or lines of classes. However, there is another type of array which is known as a two dimensional (2D) array. A 2D array allows us to create a matrix or table that we can use to store data. The best way to show this is using a game of noughts and crosses.

A noughts and crosses board looks like the following:

			Or a 2D array	[0][0]]	[0][1]]	[0][2]]
				[1][0]]	[1][1]]	[1][2]]
				[2][0]]	[2][1]]	[2][2]]

Other uses for 2D arrays could be keeping track of seats that are available on trains, planes, concert halls etc. When data needs to be stored in a table form a 2D array is very useful.

Check Your Understanding

1: Create the following Noughts and Crosses program. This shows how Columns and Rows are set out.

```
Public class TicTacToe{

    Private String [][] board;
    Private static final int ROWS = 3;
    Private static final int COLUMNS = 3;

    Public tictactow()
    {
        board = new String[ROWS][COLUMNS];
        for (int i = 0; i < ROWS; i ++)
            for (int j = 0; j < COLUMNS; j++)
                board [i][j] = " " ;

    }
    public void set(int i, int j, string player)
    {
        if (board[i][j].equals(" ")){
            board[i][j] = player;
        }
    }
    public String toString(){

        String r = " ";
        For (int i = 0; i ++)
        {
            r = r + "|";
            for (int j = 0; j < COLUMNS; <j++)
                r = r + board[i][j];
            r = r + "|\n";
        }

        return r;
    }
}
```

THIS IS THE MAIN CLASS

```
Import java.util.Scanner;

Public tictactoeRunner
{
    public static void main (String[] args){

        Scanner in = new Scanner(System.in);
        String player = "x";
        TicTacToe game = new TicTacToe();
        Boolean done = false;

        While(!done)
        {
            System.out.print(game.toString());
```



```

System.out.print("Row for " + player + " (-1 to exit)");
Int row = in.nextInt();
If (row <0) done = true;
Else
{
    System.out.print("column for " + player );
    Int column = in.nextInt();
    game.set(row, column, player);
    if (player.equals("x")){
        player = "o";
    }
    else{
        player = "x";
    }
}
}
}
}

```

(note some of the capitalization maybe off in the above example).

2: Airline Seating Chart

Imagine that the two dimensional array could be used to allocate seats on an airplane. It would become quite a useful tool. Devise a program that will represent an aircraft as a two dimensional array of classes, you can then use this two dimensional array to store passenger details.

Sorting Algorithms

Algorithm Efficiency

Algorithm Efficiency is determined in the following way:

In sorting it is the number of comparisons/moves that need to be made in order to complete the task and when data is added to the data set what effect that has on the number of comparisons / moves that need to be made. The effect of the data is measured using Big O notation.

You need to know how the following algorithms operate and what their efficiency is, however to start with we need to understand how the Big O notation quantifies algorithm efficiency.

O(1) The algorithm uses the same amount of operations regardless of the size of the data structure or input.

O(log n) The time taken to complete the algorithm increases slightly each time, however it is not exponential or linear to the size of the input. If you imagine trees, half of the tree gets removed each iteration of the algorithm.

O(n) the number of operations required to complete the task increases in direct proportion to the size of the input

O(n²) the time taken to complete the algorithm grows exponentially in proportion to the size of the input

Efficiency	Example of Algorithms
O(1)	Non Programming examples: Cooking, it takes a constant amount of time no matter how many people cook as the dish will always take the same length of time to cook Programming Examples: Accessing an index in an Array (<code>int a = arr[1]</code>), inserting a node in a linked list, pushing or popping on a stack, insertion and removal from a queue
O(log n)	Non Programming Example: looking up a word in the dictionary, looking up a contact in a phone book Programming Examples: Binary search, finding the largest/smallest number in a binary tree, Calculating Fibonacci numbers (using the most efficient algorithm)
O(n)	Non programming example: Reading a book the bigger the book the longer it takes. Programming Example: Traversing an array, traversing a linked list, linear search, comparing two strings, checking for a palindrome
O(n ²)	Bubble sort, selection sort, insertion sort, traversing a simple 2D arrays

The algorithms you need to know for this course are: sequential search, binary search, bubble sort and selection sort. The following section describes each algorithm with its pseudocode, how it operates and its efficiency.

The following array will be used to demonstrate how the algorithms work:

TEST_SCORES [10, 34, 11, 45, 67, 54, 40, 32, 9, 15]

Algorithm: Sequential Search	
Pseudocode	Description
Set found = false Set Index to 0 Input search item While not found and not at the end of the array If list at index = search item Found = true Location = index Endif End while loop If found = false output that item was not found If found = true output location item was found at.	A Boolean variable is set to false indicating the item has not been found. The user is asked for the search term. While the item is not found each element of the array is inspected. If the search term matches the current value in the array then the Boolean is set to true and the loop ends. The location is set to the current index in the array. If not found then a message states not found. If found a message stating the item has been found and the index of where in the array is printed.
Walk through of the algorithm Found = false Search term = 67 First iteration of loop 10 != 67 found remains false second iteration of loop 34 != 67 found remains false third iteration of loop 11 != 67 found remains false fourth iteration of loop 45 != 67 found remains false fifth iteration of loop 67 = 67 found = true loop ends output = location of item in array 4	
Algorithm: Binary Search	
Pseudocode (iterative)	Description
BinarySearch(A[0..N-1], value) { low = 0 high = N - 1 while (low <= high) { // invariants: value > A[i] for all i < low value < A[i] for all i > high mid = (low + high) / 2 if (A[mid] > value) high = mid - 1 else if (A[mid] < value) low = mid + 1 else	Binary tree created and value input from user. Search term compared to root of tree. If search term is > root then right subtree is removed. If search term < root then left subtree is removed. Next node visited. If search term > node the right subtree is removed else left subtree is removed. This continues until the item is found or all possible nodes visited.

<pre> return mid } return not_found // value would be inserted at index "low" } </pre>	
Algorithm: Bubble Sort	
Pseudocode	Description
<pre> For I = 1 to N - 1 For J = 1 to N - 1 If (A(J) > A(J + 1)) Temp = A(J) A(J) = A(J + 1) A(J + 1) = Temp End-If End-For End-For </pre>	The first and second item in the array are compared, if the first is higher than the second then they swap. The second and third are then compared if the second is higher than the third they are swapped. This continues until a lower value is found and it can not be swapped. The start of the array is then revisited until the array is in order.
Walk through of the algorithm 1st iteration of loop 10, 34, 11, 45, 67, 54, 40, 32, 9, 15 2nd iteration of loop 10, 11, 34, 45, 67, 54, 40, 32, 9, 15 3rd iteration of loop 10, 11, 34, 45, 54, 67, 40, 32, 9, 15 4th iteration of loop 10, 11, 34, 45, 54, 40, 67, 32, 9, 15 5th iteration of loop 10, 11, 34, 45, 54, 40, 32, 67, 9, 15 6th iteration of loop 10, 11, 34, 45, 54, 40, 32, 9, 67, 15 7th iteration of loop 10, 11, 34, 45, 54, 40, 32, 9, 15, 67 8th iteration of loop 10, 11, 34, 45, 40, 54, 32, 9, 15, 67 etc. until the array is sorted	
Algorithm: Selection Sort	
Pseudocode	Description
<pre> For I = 1 to N-1 do: Smallsb = I For J = I + 1 to N-1 do: If A(J) < A(Smallsb) Smallsb = J End-If End-For Temp = A(I) A(I) = A(Smallsb) A(Smallsb) = Temp End-For </pre>	The smallest value is set to the first item of the array. The array is then searched to find the smallest value. Once the smallest value is found the values are swapped. The second value is then set as the smallest value, the array is then searched to find a smaller value if found they are swapped. This continues until the array is sorted.
Walk through of the algorithm 1st iteration of the loop 9, 34, 11, 45, 67, 54, 40, 32, 10, 15	

2nd iteration of the loop

9, 10, 11, 45, 67, 54, 40, 32, 34, 15

3rd iteration of the loop

9, 10, 11, 45, 67, 54, 40, 32, 34, 15

4th iteration of the loop

9, 10, 11, 15, 67, 54, 40, 32, 34, 45

5th iteration of the loop

9, 10, 11, 15, 32, 54, 40, 67, 34, 45

6th iteration of the loop

9, 10, 11, 15, 32, 34, 40, 67, 54, 45

7th iteration of the loop

9, 10, 11, 15, 32, 34, 40, 67, 54, 45

8th iteration of the loop

9, 10, 11, 15, 32, 34, 40, 45, 54, 67

9th iteration of the loop

9, 10, 11, 15, 32, 34, 40, 45, 54, 67

TOK Thoughts

In Computer Organisation we will look more closely at how data is stored however in the past ASCII code was used to store characters. ASCII allowed 128 characters including special characters to be stored in the machine. As the use of computers became more global it was understood that 127 characters is not nearly enough characters to store the character sets required, Unicode is now used which uses 16 or 32 bits to store characters.

One thing to think about is the statement:

“Without Unicode we would have no global computer system”

To what extent do you agree with this statement? Can you find evidence to support or disprove this statement.

Legally programmers must abide by the intellectual property laws, which states that you may not use another’s program without their permission. You must also when using open source code fully acknowledge the original author of the code. However when producing a piece of code it is your responsibility to ensure it works correctly.

With this in mind think about the following statement, do you agree or disagree with the statement. In the commercial world to what extent is the statement true.

“Programmers should take full responsibility for their code, if there is a flaw in the code that is exploited by malicious code/hackers, it is the programmer who is at fault”



Additional Projects - SL

Project One – The SuperHero Decider

IF Statements

Instructions:

It is a well known fact that all super heroes have unique characteristics. The following flow chart shows the unique character traits of each super hero. What you have to do is create a program that will ask the user questions and depending on their answers show them which super hero they should be and their characteristics.

Project Two – Password Protector

Loops

Instructions:

While loop (conditional loop)

The while loop allows us to repeat until a condition is met. Here is the pseudocode for a password protection program. Implement this program in Java.

Create password variable

Create userInput variable

Create found variable and set to false

Create tries variable

Start loop

 Get user input

 If userInput = password

 Display enter message

 Found = true

 End if

End loop when tries > 3 or found = true

 Display failed message



Extension Task (higher level): create a password list in a text file. Compare the user input to the text file. If the password appears in the field then allow access.

For Loop (unconditional loop)

The for loop allows us to repeat a section of code as many times as we tell it too. Here is the pseudocode for entering data into an array.

Create integer [10] testMarks

for each item in array

 create temp integer

 temp = user input

 current item in array = temp

end loop

Extension Task (HL and SL): Find the highest test result and display it on screen, find the lowest test result and display it on screen. Calculate the average test result and display it on screen.

Project Three – The Alias Generator

Arrays

Instructions:

Sometimes people like to remain anonymous and its harder than you think to come up with an alias. Therefore you should create an alias generator for creating anonymous names.



The outline pseudocode for this task would be:

Declare firstname array [10]

Declare surname array [10]

Add 10 names to firstname array

Add 10 names to surname array

Loop

 Generate random firstname

 Generate random surname

 Display name

Until

 User presses x

* note this program does make use of the random number generator which is part of the maths class *

Project Four – Amazon WishList

Arrays of classes

Instructions

Amazon allows you to make a wish list of items that you would like to buy. This acts as a reminder for when you feel like buying something or you can send it to people for Christmas / Birthdays etc.

You should create a similar idea. Create an “item” class, see the object diagram below, then allow people to enter the items they would like to purchase at a later date.

Once they have entered their items they should be printed out in a list on screen so people can view them easily.

Item
Int id
String desc
String dept
Double cost
String message
Int getID()
String getDesc()
String getDept()
Double getCost()
String getMessage()

Main Method:

Create array

For each item in the array

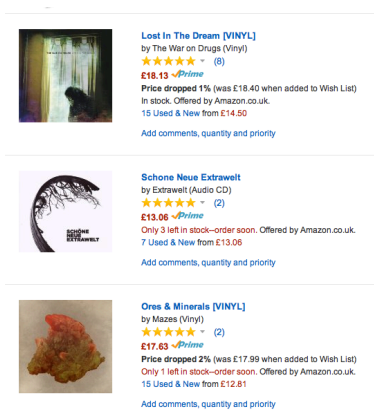
Enter data

Create a new instance of the item class

End loop

For each item in the array

Print the name and description of each item



Extension task – if you are HL you could use an array list and let the user store as many items as they like. This could also be saved to file for use at a later date.

Chapter Eight - Programming – Higher Level

(this unit incorporates HL Advanced Data Structures)

Essential Understanding: “The way data is stored in a program does affect the quality of a program”

Key Questions:

- Why is data storage important?
- What is the difference between abstract data structures and static data structures?
- Would programs be less efficient without abstract data structures?
- Are abstract data structures interchangeable or are they designed for specific purposes?

ATL Skills:

Self Management –
Practice analyzing and attributing causes of failure. Practice failing well. Demonstrate persistence and perseverance.

Thinking: Create novel solutions to complex problems.

Recursion

“I like broccoli its recursive” – possibly the most geeky thing ever said about a vegetable but I stand by my statement. A broccoli is an example of recursion occurring in nature. When you have a broccoli it and break is down, it holds the same shape getting smaller and smaller until it gets too small to see.

			
This is a larger form of broccoli,	When broken down it looks the same	Broken down further still the same	In a smaller form it is still the same.



Another form of natural recursion you may be familiar with is when you look in the mirror with another mirror and the image seems to never end. Or there are a few examples with art and album covers.

As you can see the image is repeated until we can no longer see the image, this is know as the “base case” you might have also used recursion if looking for somewhere in a strange city. You ask for directions based on your current location, head in the location told until you find yourself unsure again then you use your current location to restart the same procedure. This can be shown in the following pseudocode:

Finding the Eiffel Tower

- 1: if at the Eiffel tower stop
- 2: otherwise ask for directions
follow directions until unsure
- 3: repeat method using new location



Commons.wi
kimedia.org

Very simply put recursive methods are methods that call **themselves** with **modified** parameters until they reach their base case.

Advantages of using recursion are that the solutions tend to be shorter and less code intensive than iterative solutions however they are very memory intensive and can take a lot of processing power. As you will find out when we complete the following tasks.

Recursion in Computer Science

Fibonacci numbers are one of the most common algorithms for demonstrating recursion. Fibonacci numbers are as follows:

$$\begin{aligned} F(1) &= 1 \\ F(2) &= 1 \\ F(n) &= F(n-1) + F(n-2) \end{aligned}$$

If you are mathematically challenge like me, in Fibonacci numbers the current number is the sum of the previous two numbers. Therefore the Fibonacci sequence is as follows:

1 1 2 3 5 8 13 21 34 55 89 144 377 610

The algorithm for producing Fibonacci sequence is:

- 1: get user input
- 2: print Fibonacci number
 - 2.1 = if $n \leq 2$ Fibonacci number = 1
 - 2.2 = else Fibonacci number = Fibonacci number – 1 + Fibo



Towers of Hanoi

One of the most famous recursive problems is the “Towers Of Hanoi”. The Mayans had a belief that if this game was solved with 64 discs the world would end. Why did they think this? And how did technological advancement start to spoil indigenous beliefs? (TOK)

You may have played it before or studied it in Maths. In case you haven’t you will find it at the following link:

<http://www.mathsisfun.com/games/towerofhanoi.html>

Solving the problem

input: integer n , such that $n \geq 1$

1. **if** n is 1 **then return** 1
2. **return** $2 * [\text{call hanoi}(n-1)] + 1$

end Hanoi

```
public class TowersOfHanoi {

    private int size;
    private String fromPeg;
    private String toPeg;
    private String usingPeg;

    public TowersOfHanoi(int size, String fromPeg,
                        String toPeg, String usingPeg) {
        this.size = size;
        this.fromPeg = fromPeg;
        this.toPeg = toPeg;
        this.usingPeg = usingPeg;
    }

    public void solveTower() {
        // Print out some header information.
        System.out.println("Steps for solving " + size +
                           " disk tower of hanoi puzzle, ");
        System.out.println("starting with discs on " + fromPeg +
                           " and moving discs to " + toPeg + ":");
        System.out.println();

        // Solve the problem by moving size disks from the
        // fromPeg to the toPeg using the usingPeg.
        solveTower(size, fromPeg, toPeg, usingPeg);
    }

    // Recursive helper method that prints the instructions
    for
    // moving numDisks from the fromPeg to the toPeg using
    the
    // usingPeg.
    private void solveTower(int num, String from,
                            String to, String using) {

        if (num == 1) {
            // Base Case: Move 1 disk...
            System.out.println("Move disk from " + from +
                               " to " + to + ".");
        }
        else {
            // Recursive Case:

```

```

        //    Move num-1 disks from the from peg to
        //    the usingPeg using the toPeg.
        solveTower(num-1, from, using, to);    // RC1

        //    Move 1 disk from the fromPeg to the toPeg using
        //    the usingPeg.
        solveTower(1, from, to, using);        // RC2

        //    Move num-1 disks from the usingPeg to
        //    the toPeg using the fromPeg.
        solveTower(num-1, using, to, from);    // RC3
    }
}

public static void main(String[] args) {
    TowersOfHanoi tower = new TowersOfHanoi(64,"A", "B",
"C");
    tower.solveTower();
// C1
}
}

```

TOK Links – indigenous knowledge

Mayans believed that if this problem could be solved with 64 discs the world would end. Using technology we have solved the problem it would take 18,446,744,073,709,551,615 moves. With this in mind is technology ruining indigenous knowledge?

Check your Understanding

- 1: Implement the Fibonacci number sequence in java
- 2: trace the tower of Hanoi algorithm with the input 3
- 3: implement towers of Hanoi in java

Abstract Data Structures (ADT's)

So far in this course you have used arrays of classes to store the data in use. Although they have been effective for the programs you have written so far, there are several limitations to arrays. One of the main disadvantages is that you have to declare the size of the array at the time of running the program. This is a problem because how do you know how much data someone wants to add? Also this means that this section of memory will be permanently blocked off for the array, what happens if you only use 10% of it? Searching algorithms and sorting algorithms would be hideously inefficient and you would be wasting lots of memory for no reason.

The alternative – the abstract data structure grows and shrinks to meet the demands of the program. They also have libraries of methods associated with them so we can write more effective programs very quickly.

Compare the following:

Array of Classes (cow classes)		Dynamic structure of classes (cows)
	Memory ineffective as Storing more space Than necessary	
	Can only store 7 cows The list is limited by The RAM available	
	Sorting in this array Will always take the Same time a smaller list = less time	
	Searching in the array Will always take the same time Smaller list = less time	
	Both are referenced in a Similar way	
	Lists have their own Methods that make Them very useful	
	You can read straight Into an ADT from a File	

ADT – Array List

One of the Advanced Data Structures you need to know about is an Array list. An array list has similar properties to arrays however, their size does not have to be declared at runtime. Also Array lists only store objects. If you want to store primitive data types in Java using an array list you need to use the following “cheats”. Byte – Byte, Boolean – Boolean, char – Character, double – Double, float – Float, long – Long, short – Short.

Methods that you will find useful when using Array Lists:

Method	Action
<code>ArrayList<pupils> myClass = new ArrayList <pupils>();</code>	This code constructs an empty ArrayList that will hold instances of the pupil class.
<code>myClass.add(new pupil(parameters));</code>	Adds a new instance of pupil to the myClass array list (this will add the pupil to the end of the list)
<code>myClass.add(1, jess);</code>	Adds a new instance of the pupils class to the array list at the index 1
<code>myClass.remove(2);</code>	Will remove the instance of the object at index 2
<code>myClass.set(5, maya);</code>	Will replace the instance of the pupil object at index number 5 with the pupil object maya.
<code>Pupil temp = myClass.get(myClass.size()-1);</code>	Gets the last instance in the array list and stores it in a temporary pupil object.

Example program using an array list:

This program sets up an array list of animals called theFarm then prints out all the cows on the farm (simple but shows some useful methods).

```

public class TheFarmMain {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        ArrayList <animal> theFarm = new ArrayList <>();

        theFarm.add(new animal("Cow","George"));
        theFarm.add(new animal("Pig", "Jackie"));
        theFarm.add(new animal("Cow", "Patsy"));
        theFarm.add(new animal("Sheep","Alice"));

        for (animal temp : theFarm) {
            if (temp.getType().equals("Cow")){
                System.out.println ("the name of the animal is " + temp.getName());
            }
        }
    }
}

```

Check Your Understanding

1: Implement the Farm program above and ensure you can get the addition function and getting information from the array list working. You should also try to remove an item from the array list.

2: Criminal Database:

Here are some pretty dodgy looking people.



They have committed some fairly innocuous crimes because look at them they have parrot pets. However, they must still be charged and serve the time for their crime. It is your job to create a pirate class to store the details of any pirates and then use a main class to store the information about them in an array list. You should be able to add and remove pirates as necessary. As an extension use the date/time function any pirate with a crime over 6 months old should be pardoned.

Information

Pirate A – Name: Sandy McDonald Home Port: Lerwick Theft: Diamonds Date: 15th September

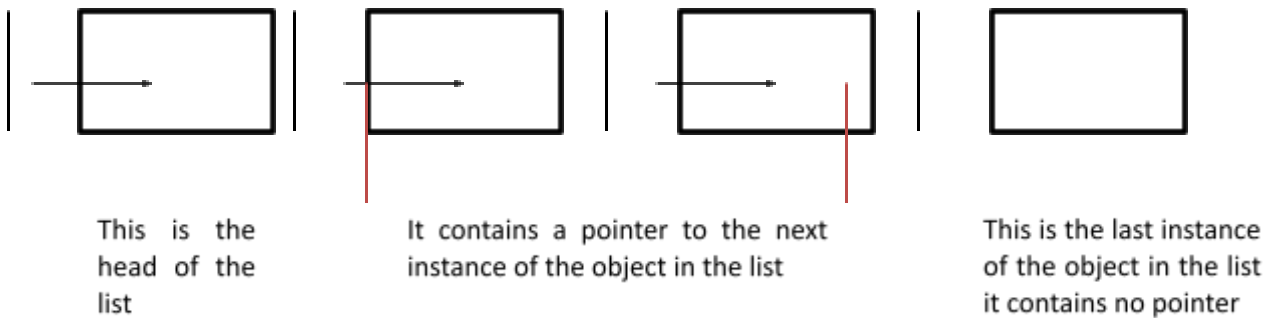
Pirate B – Name: Bob Walker Home Port: Invergordon Theft: Rum Date: 20th January

Pirate C – Name: Archie McDonald Home Port: Arran Theft: Rubies Date: 23rd May

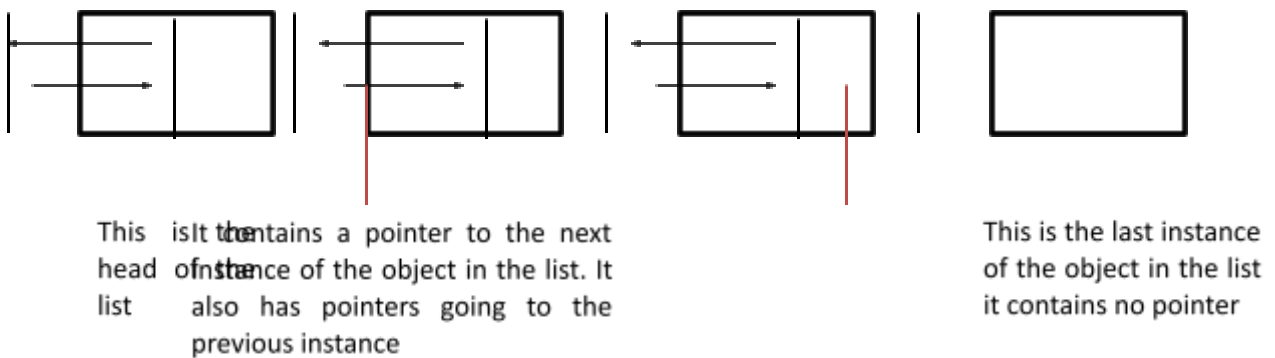
Pirate D - Name: Steve Smith Home Port: South Shields Theft: Coal Date: 5th June

ADT – Linked List

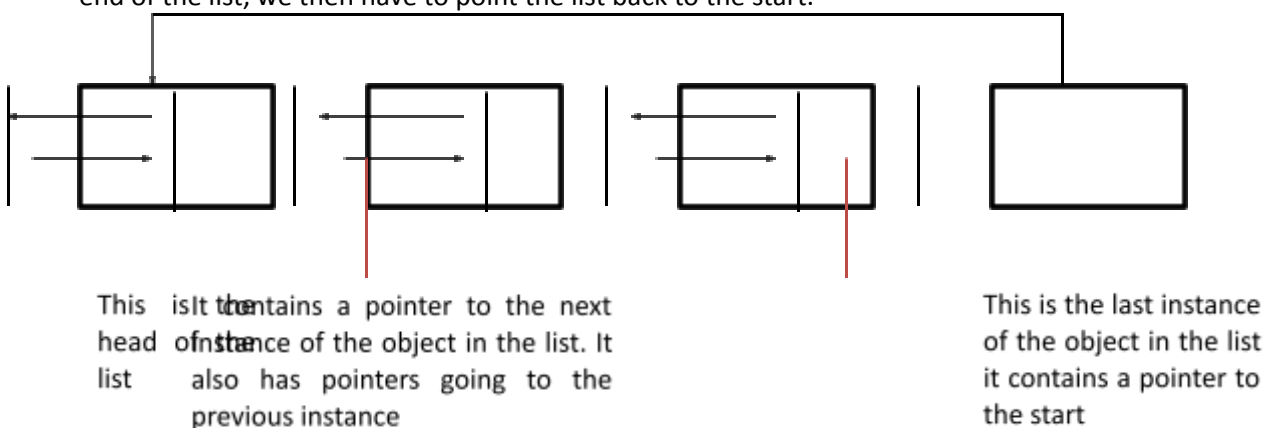
Linked lists are created using the following structure:



This is an example of a **singular** linked list, it contains a head and then one way pointers to the end of the list. The list can be traversed one way only and once it reaches the end it has to be pointed back to the start.



This is an example of a **double** linked list, it contains pointers to the next object and pointers to the previous object. This means the list can be traversed both ways. Unless we reach the end of the list, we then have to point the list back to the start.



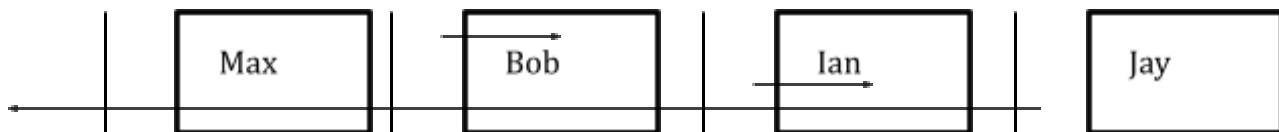
This is a **circular** linked list, it contains pointers back and forth as well as a pointer from the last instance to the first instance in the list. It is the most comprehensive of all the linked lists. However, in IB all you need to be able to code is the singular linked list.

The advantages of linked lists are as well as being able to expand and contract as required, searching and sorting does not require any physical movement of the objects just the file.

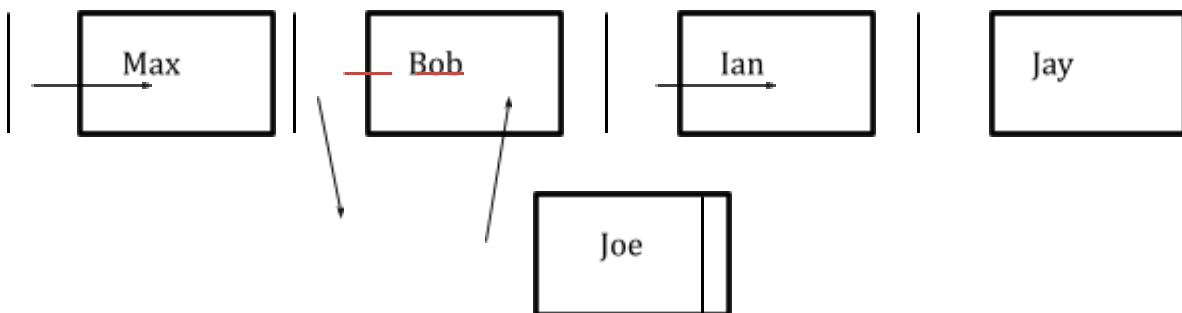
For example to sort the list



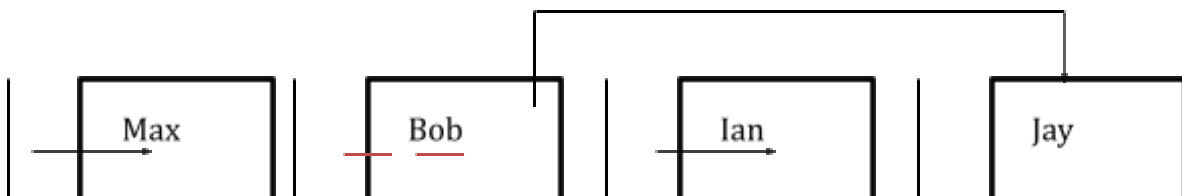
We don't move the object just the pointers. The start pointer begins at bob and then moves from there.



To add an instance break the pointer between two instances and add the object by repointing them



To remove an instance break the pointer between the two instances and repoint the pointer to the next in line



Every so often a garbage disposer (yes really) will come along and remove the objects that have been discarded.

The good news is for this course you can use the LinkedList class in java. The previous course you had to write them yourself! Methods you need to know are shown below as is an example program. One important thing you need to know about linked lists would be that they need a special helper method called an iterator to look through the list. I like to think of an iterator as a cursor blinking before the current object that allows you to attack the object but that might just be me.

Method	Action
<code>LinkedList<pupils> myClass = new LinkedList <pupils>();</code>	This code constructs an empty LinkedList that will hold instances of the pupil class.
<code>Add (E e)</code>	Add an instance of the object at the current point in the linked list
<code>Add (int index, E e)</code>	Add an instance of the object at a specified index
<code>AddFirst (E e)</code>	Will add an instance of the object to the start of the linked list
<code>AddLast (E e)</code>	Will add an instance of the object to the end of the linked list
<code>Clear()</code>	Will empty the list
<code>Get (int index)</code>	Will return the object at a specified index
<code>getFirst()</code>	will return the first object in the linked list
<code>getLast()</code>	Will return the last object in the linked list
<code>Remove()</code>	Will remove the current object in the list
<code>Remove (int index)</code>	Will remove the object at a specified index
<code>removeFirst()</code>	Will remove the first object in a list
<code>removeLast()</code>	Will remove the last object in a list
<code>Size()</code>	Will return the number of objects in the list
<code>isEmpty()</code>	Will check if the list is empty

```

import java.util.LinkedList;
import java.util.ListIterator;
public class ListTester{

    public static void main (String [] args){
        LinkedList<String> staff = new LinkedList<String>();
        Staff.addLast("Harry");
        Staff.addLast("Diana");
        Staff.addLast("Romeo");
        Staff.addLast("Tom");

        ListIterator<String> iterator = staff.listIterator();
        Iterator.next();
        Iterator.next();
        Iterator.add("Juliet");
        Iterator.add("Nina");
        Iterator.next();

        Iterator.remove();
        For (String name: staff)
        System.out.println(name + " ");
        System.out.println();

    }}

```

Check Your Understanding

- 1: Implement the program above paying particular attention to how the iterator works.
- 2: The linked List Quiz – how pirate are you?

On Facebook you constantly see updates such as “if I was a character from sesame street I would be the Count” or “the city I should definitely live in is Paris”. In this task you are going to create a program that tests how pirate someone is. In an advanced data structure you will store questions. Each will have 4 possible choices and each choice has a points value. You should keep a running total of the points and at the end depending on how many points the person has display how pirate they are. (the question are on the next page)

Class Diagram:

Question
String Question String answerA String answerB String answerC String answerD Int valueA Int valueB Int valueC Int valued
getQuestion() getAnswerA() getAnswerB() getAnswerC() getAnswerD() getValueA() getValueB() getValueC() getValueD()

Program Outline:

- 1: Create Linked List
 - 2: Populate linked list
 - 3: for each question
 - display question
 - get answer
 - get value of answer
 - add value to current total
 - 4: display pirate worthiness
 - if value is > 30
 - display very message
 - if <30 and > 25
 - display high message
 - if value <25 and > 18
 - display average message
 - if value < 18
 - display not at all message
- Questions value of answers in brackets:

1: what is your preferred pet?

Parrot (5)

Cat (4)

Dog (3)

Unicorn(1)

2: What is your favourite color?

Black (3)

Silver (4)

White (1)

Gold (5)

3: What is your preferred mode of transport?

Car (1)

Bike (3)

Horse (4)

Boat (5)

4: What is your preferred hair style?

Long (5)

Short and neat (1)

Dyed (3)

Curly (2)

5: What would be your preferred fortune?

Clothes (1)

Geeky Stuff (2)

Gold (5)

Car (3)

6: How do you say "hello"?

Ahoy (5)

Hola (3)

Good day (1)

Hi (2)

> 30

You are indeed a pirate, head out on your boat, raise your skull and crossbones and sail off into the sunset.

<30 and >25

you are well on your way to becoming a pirate, with a bit more training you could be spending your life at sea.

<25 and > 18

You are average, you dream of the sea but you are far too nice to live it. Maybe consider the wanderlust lifestyle

<18

You couldn't be less of a pirate if you tried and that is ok the world needs good people too!

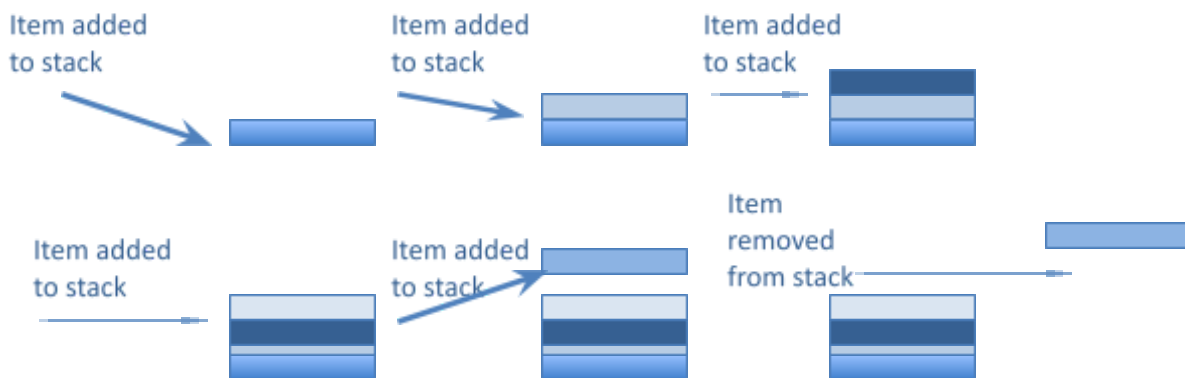
ADT – Stack

Stacks are like a plate of dishes. You can add to the top and stack up items but you can only remove from the top.

Stacks can be used to implement the following in Computer Science.

the towers of Hanoi
syntax parsing for example {}
performing calculations
in quick sorts to store temporary partitions

How a stack works:



Stacks are an example of a first in last out (filo) system. The methods of the stack that you need to know are:

push (add something to the top of the stack)
pop (remove the top of the stack)
isEmpty (checks for an empty stack)

The following is a program that makes use of a very simple string stack:

```
public static void main(String[] args) {

    Stack <String> myStack = new Stack();

    myStack.push("hello");
    myStack.push("my name is: ");
    myStack.push("Linz");

    System.out.println("peeking: " + myStack.peek());

    while(!myStack.isEmpty()){
        String temp = myStack.pop();
        System.out.println("the pushed item was: " + temp);
    }
}
```

Check Your Understanding

1: implement the program above and make sure you have a good understanding of what the pop, push and peek functions do.

2: A past paper question in the past has used the following example. “Students on a field trip are collecting data for an experiment” Explain why a stack is a good data structure for this task. Assuming they are on a Biology trip and counting the number of plants that appear in the field.

I did this at school using a frame with string squares and counted the number of daisies in each square but this might have aged me!



Program the stack to show why this is a good way to count daisies. E.g. it should allow the users to enter as many daisies as they like then calculate the total.

ADT – Queue

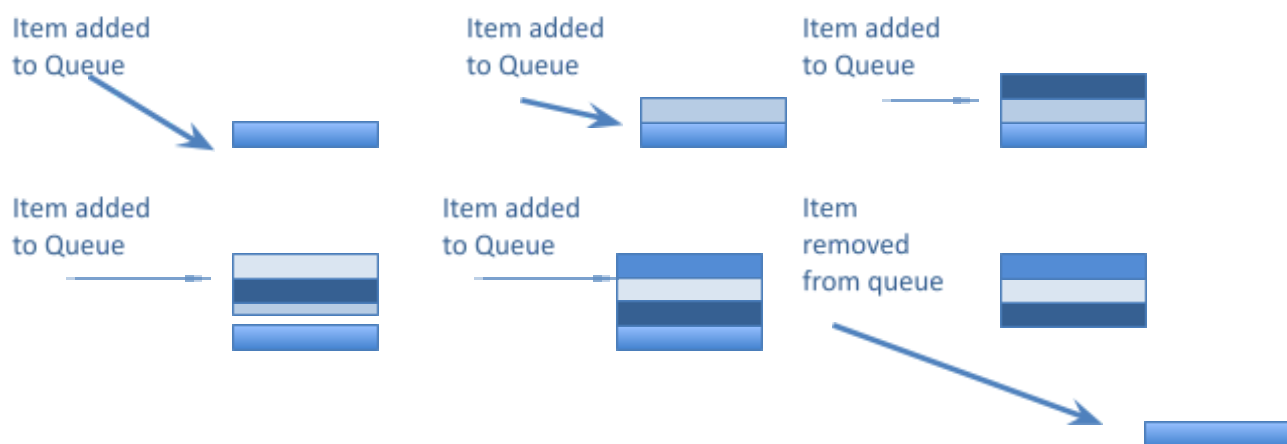
Queues are, as the name suggests, like a queue of items (escalator, check out queue, coin dispense). You add to the queue at the bottom and remove from the top (metaphorically).

Queues can be used to implement the following in Computer Science.

- high priority queuing is used to manage bandwidth. For example making sure VOIP gets priority over non urgent requests. This prevents bandwidth becoming over full.
- It can be used in the programming of simulations
- It can be used in scheduling



How a queue works:



Queues are an example of a first in first out (fifo) system

```
public class ExampleQueuev2 {

    LinkedList <String> myQueue;
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        new ExampleQueuev2();
    }
    public ExampleQueuev2() {

        myQueue = new LinkedList <String>();

        myQueue.addLast("how do like");
        myQueue.addLast("them apples");
        myQueue.addLast("bad movie");
        myQueue.addLast("reference");
    }
}
```



```

        System.out.println(myQueue.removeFirst());

        myQueue.removeFirst();
        myQueue.removeFirst();

        System.out.println(myQueue.getFirst());
    }

}

```

As you can see there is no queue class, therefore we have to use a linked list to create a queue. The example above is the format the IB use for linked lists.

Check Your Understanding

1: Implement the above linked list and check that you understand how it works. Be aware that the main method only calls the constructor of the “ExampleQueuev2” method. This is a little different to how we have done things before.

2: A past paper question for this unit is shown below. Implement the queue suggested to prove that the queue is the best data structure for this. You will need a bus class which stores the driver, the route and the id number of the bus.

“A bus company decides to run a simulation over a particular route to see what happens when several buses are started on the route a set time apart. A queue will be used to hold the individual bus instances. Identify three features of a queue that make it suitable for this purpose”



Northeastbuses.co.uk

The bus information is as follows

Bus: 1011, Driver: C Humbley Route: Lanchester

Bus: 2012, Driver: M Hillier, Route: Lanchester

Bus: 2911, Drive: J Inglis, Route: Chester – le – Street

ADT – Binary Tree

A tree is a dynamic data structure which uses pointers to show where to get the next data from. As trees are sorted, if a tree is balanced, it can be more efficient to search for data in the tree.

Trees can be used for storing unique keys (references to files) so they can be retrieved quickly. They can be used for searches when the search data is changing regularly.

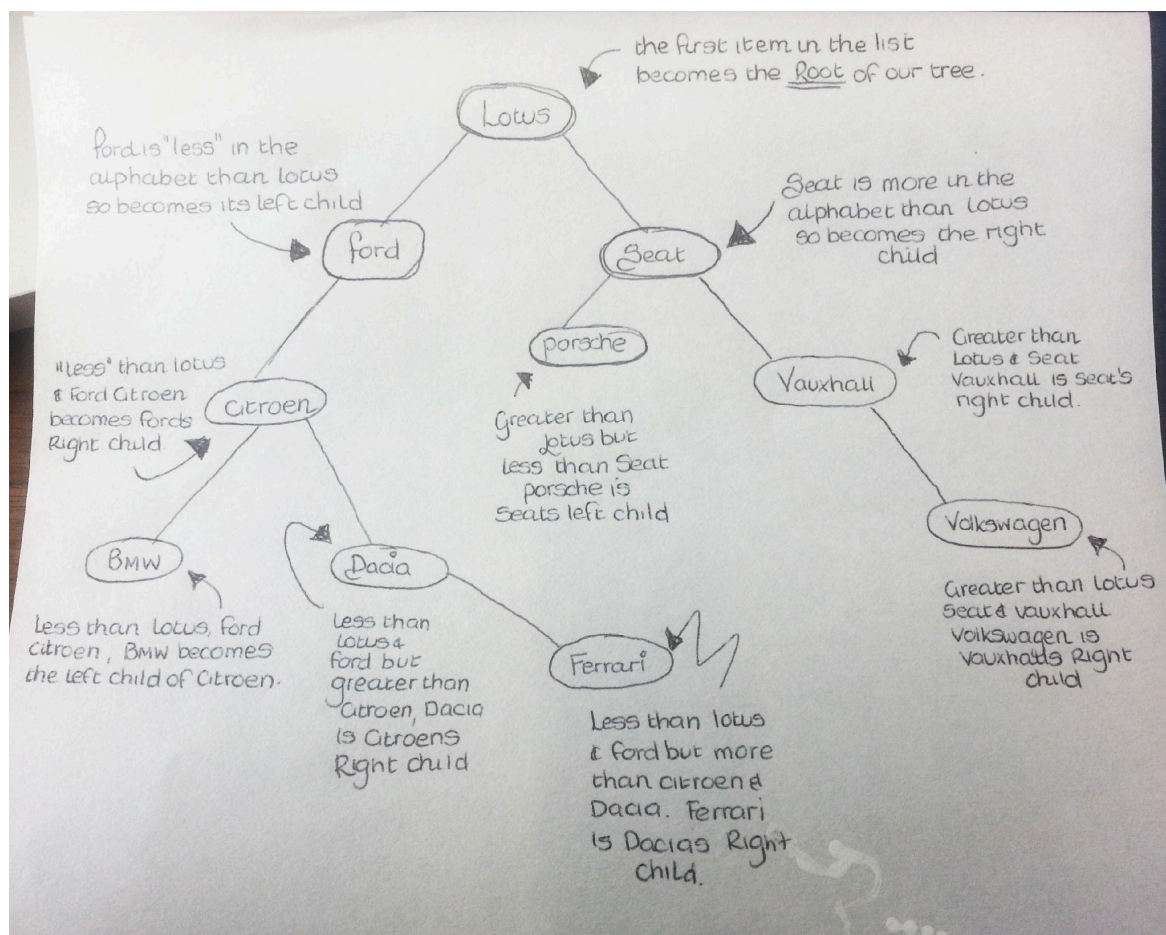
Non binary trees – in a non binary the roots can have more than two children

Binary trees – in a binary tree the roots can have a maximum of two children

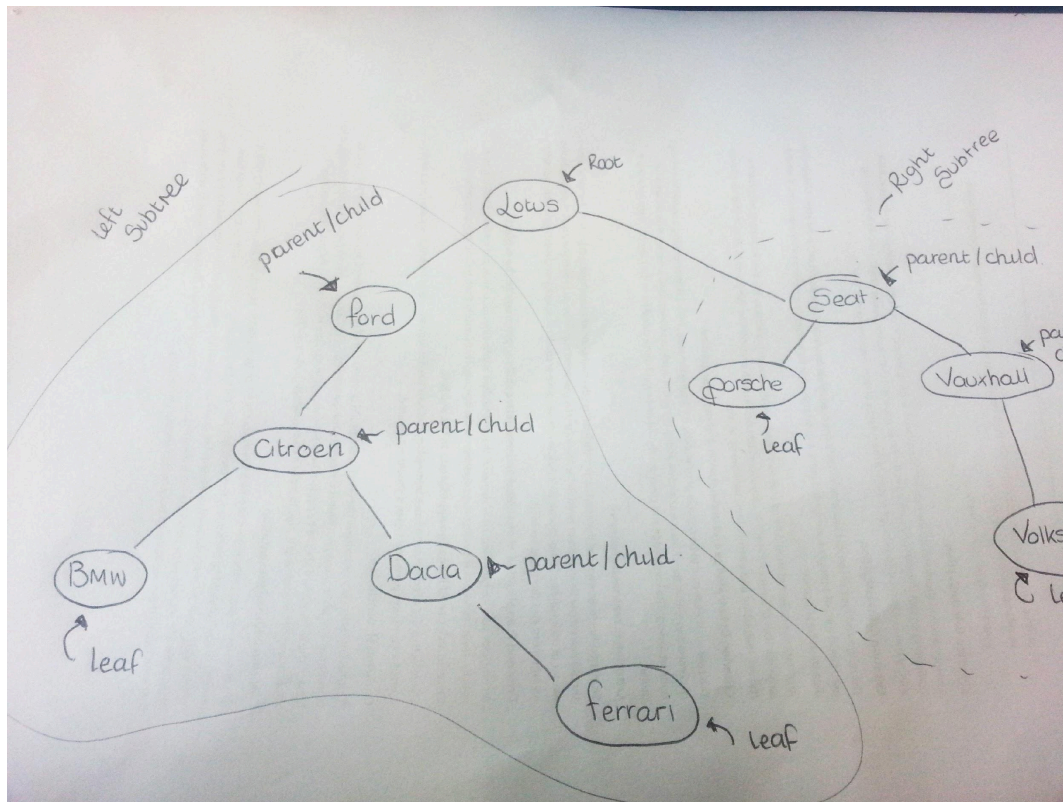
Representing Binary Trees

Binary trees work in the following way:

Lotus, Ford, Seat, Citroen, Vauxhall, Volkswagen, Porsche, BMW, Dacia, Ferrari



Each node can have two children (one left and one right), each child has one parent. The nodes at the edge of the tree are called leaves. The starting node is the root and each whole branch from a node is called a subtree.



In order traversal we read the tree in the following way:

Traverse the left sub tree
 Visit the root
 Traverse the right sub tree

BMW, Citroen, dacia, Ferrari, ford, lotus, Porsche, seat, Vauxhall, volkwagen

Post order traversal we read the tree in the following way:

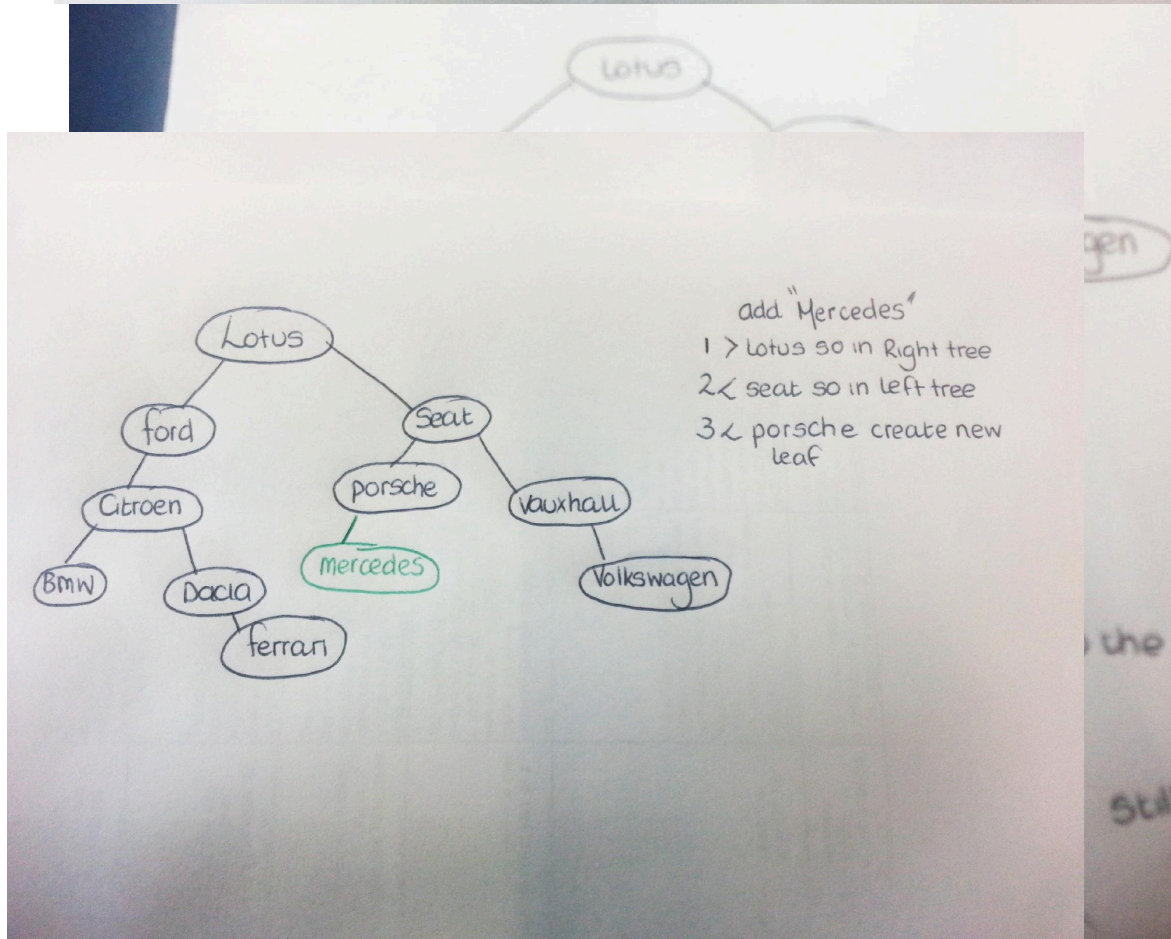
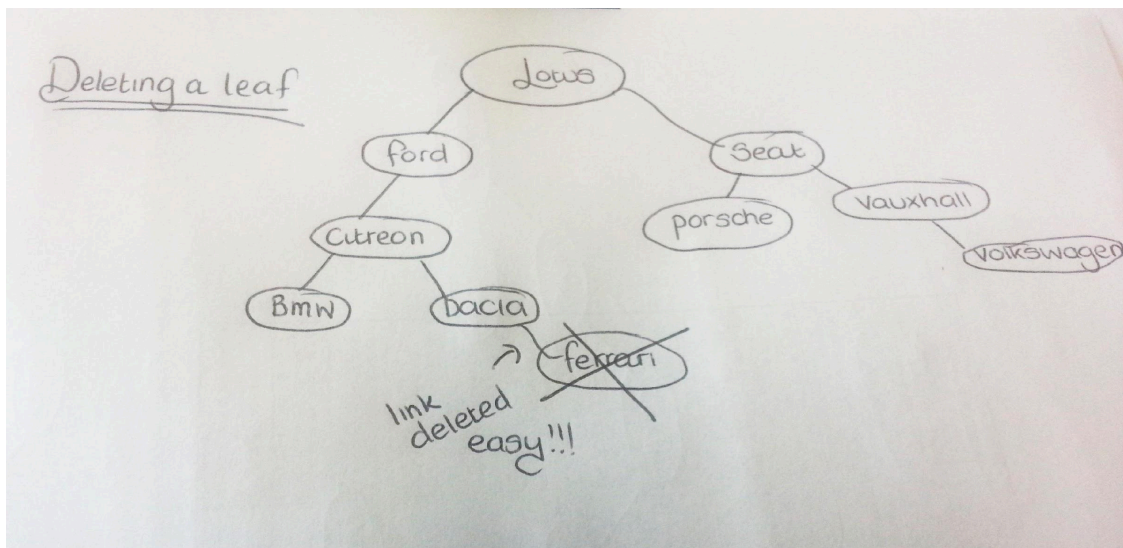
Visit all left most nodes
 come up the right from the bottom
 do the same in right sub tree
 visit root

bmw, Ferrari, dacia, citreon, ford, Porsche, volkwagen, Vauxhall, seat, lotus

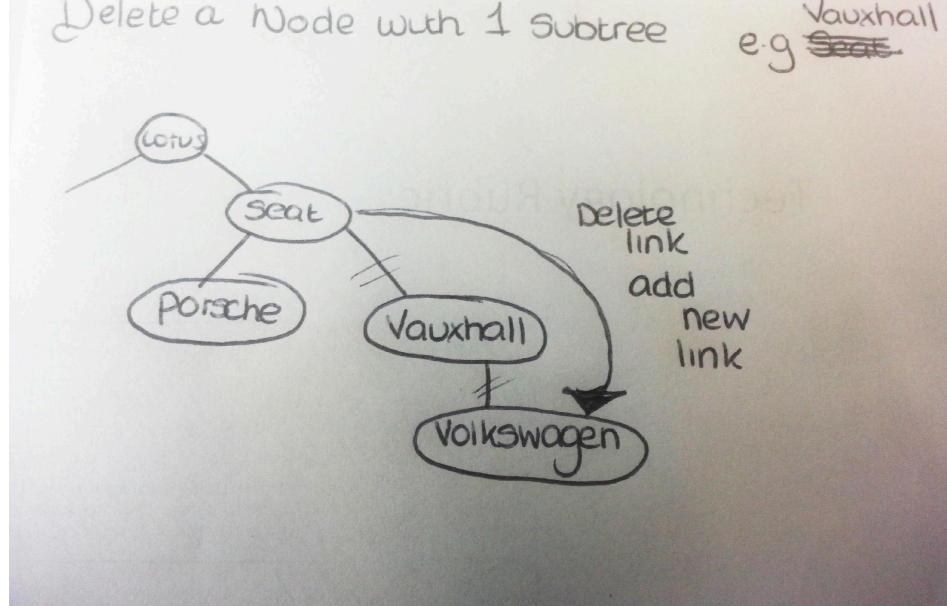
Pre order traversal we read the tree in the following way:

visit the root
 traverse the left sub tree
 traverse the right sub tree

Lotus, Ford, Citroen, BMW, Dacia, Ferrari, Seat, Porsche, Vauxhall, Volkswagen



Removing a node and the following happens:



Adding a node and the following happens:

To be truly effective Binary Trees need to be balanced.

Tree Implementation

Tree implementation is quite complex however here is code that will allow someone to add, remove and search a tree.

```

    *
    */
    public class Node {

        public Comparable data;
        public Node left;
        public Node right;

        public void addNode(Node newNode){
            int comp = newNode.data.compareTo(data);
            if (comp < 0){
                if (left == null) left = newNode;
                else left.addNode(newNode);
            }
            if (comp > 0){
                if (right == null) right = newNode;
                else right.addNode(newNode);
            }
        }

        public void printNodes(){
            if(left != null)
                left.printNodes();
            System.out.print(data + " ");
            if (right != null)
                right.printNodes();
        }
    }
}

```

```

public class BinarySearchTree {

    private Node root;

    public BinarySearchTree()
    {
        root = null;
    }

    public void add (Comparable obj){
        Node newNode = new Node();
        newNode.data = obj;
        newNode.left = null;
        newNode.right = null;
        if(root == null) root = newNode;
        else root.addNode(newNode);
    }

    public boolean find (Comparable obj){
        Node current = root;
        while (current != null){
            int d = current.data.compareTo(obj);
            if (d == 0 ) return true;
            else if (d > 0) current = current.left;
            else current = current.right;
        }
        return false;
    }

    public void remove(Comparable obj){
        Node toBeRemoved = root;
        Node parent = null;
        boolean found = false;
        while (!found && toBeRemoved != null){
            int d = toBeRemoved.data.compareTo(obj);
            if (d == 0) found = true;
            else{
                parent = toBeRemoved;
                if(d > 0) toBeRemoved = toBeRemoved.left;
                else toBeRemoved = toBeRemoved.right;
            }
        }
        if (!found) return;

        if (toBeRemoved.left == null || toBeRemoved.right == null){
            Node newChild;
            if (toBeRemoved.left == null)
                newChild = toBeRemoved.right;
            else
                newChild = toBeRemoved.left;

            if (parent == null)
                root = newChild;
            else if (parent.left == toBeRemoved)
                parent.left = newChild;
            else
                parent.right = newChild;
            return;
        }
    }
}

```

```

    }
    Node smallestParent = toBeRemoved;
    Node smallest = toBeRemoved.right;
    while (smallest.left != null){
        smallestParent = smallest;
        smallest = smallest.left;
    }
    toBeRemoved.data = smallest.data;
    if (smallestParent == toBeRemoved)
        smallestParent.right = smallest.right;
    else
        smallestParent.left = smallest.right;
}
public void print(){
    if(root!=null)
        root.printNodes();
    System.out.println();
}
}
}

```

```

    */
    public static void main(String[] args) {

        BinarySearchTree mytree = new BinarySearchTree();
        Scanner myScanner = new Scanner(System.in);

        // adds some things to the tree and then prints out the tree
        mytree.add("linzie");
        mytree.add("hates");
        mytree.add("pineapples");

        // do a comparison to see if something exists in the tree and print out an appropriate message
        System.out.println("what would you like to exist?");
        String userInput = myScanner.next();

        if (mytree.find(userInput)){
            System.out.println("it does indeed exist the world is a better place");
        }
        else {
            System.out.println("no i am sorry but here is the list");
            mytree.print();
        }

        mytree.remove("hates");
        mytree.print();
    }
}

```


Check Your Understanding

- 1: Implement the above tree and make sure you understand how it works.
- 2: If you are feeling brave, because the tree is abstract it will hold any object in the nodes. Use the above code to store objects in a binary tree.

Conventions

When developing code it is very rare you develop on your own. Therefore it is important you follow certain conventions when coding. This will help to keep your code:

- readable so others can help / further develop it
- save money as people understand it quicker

So how do we use the conventions:

1: First of all at the start of your program have a long comment explaining the name of the class, the author, the version, the date created and a description of what the class does

```
/** Class: The Mystery Class
    Author: L Stephenson
    Date: 16th September 2015
    Version: 2.1
    Description: this class covers the implement-
    Ation of the mystery class, incl, the sorting */
```

2: All Variables should have meaningful names i.e. the variable name should correspond to what it stores

```
int total_value;
String first_Name;
Boolean game_End;
```

3: All methods should have meaningful names i.e. from the name of the methods you should be able to understand what the method does

```
public int getTotalValue()
public void calculateMaxValue ()
public void resetCurrentValue (int newVal)
```

4: Constant variables (variables whose values do not change) should be all capitalized

```
int VAT_RATE = 0.20;
```

5: All pieces of code should be commented, this will allow people to fully understand what is happening in the program.

```
// this method will total the values in the
// array and then divide by the length of the
// array to find the average and return as an int
```

6: Spacing should be used to indicate which lines of code are between which brackets.

Variables

Start of code

Loop

If statement

Code to execute if true

End if

End loop

End method

Practical Only – File input and Output

Simple File I/O

So far we have only written programs that use active memory. Once the program is close any data entered into the program is lost. It is therefore useful to be able to read and write to files.

The following program is a simple program that allows us to read from one file and add it and some other data to a new file. You can find a video of it working on schoology.

```
// import items required for input / output
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Scanner;
import javax.swing.JFileChooser;
/**
 *
 *
 */
public class LineNumberer {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws
    FileNotFoundException {

        // create scanner item and ask for output file name
        Scanner console = new Scanner(System.in);
        System.out.print("Output file: ");
        String outputFileName = console.next();

        // use an array to get the data from the user
        String [] mytester = new String [5];
        for (int y = 0; y < mytester.length-1; y++){
            System.out.print("enter your data: ");
            String temp = console.next();
            mytester[y] = temp;
        }

        // create a print writer object to create the file
        PrintWriter out = new PrintWriter(outputFileName);

        // choose the other file to get data from
        JFileChooser chooser = new JFileChooser();
        Scanner in = null;
        if(chooser.showOpenDialog(null) ==
        JFileChooser.APPROVE_OPTION)
        {
            File inputFile = chooser.getSelectedFile();
```

```

        in = new Scanner(inputFile);
    }

    // write the first file to the new file
    while (in.hasNextLine()){
        {
            String line = in.nextLine();
            out.println(line);
        }
    }

    // write the array to the new file
    for (int i = 0; i < mytester.length-1; i++){
        out.println(mytester[i]);
    }
    // close files very important
    in.close();
    out.close();
}
}
}

```

Check Your Understanding

Try and create the program in your own folder. Be careful the files must be in the same folder as your Java folder.

File I/O using Bytes

So far we have looked at input and output based on the `FileInputStream` and `PrintStreamWriter`, these are fine when you would like to deal with text input and output however it can also be useful to use the `inputstream` and `outputstream` methods. These write the files using Bytes, a more complex way of storing the data.

Using input and output stream – Caesars Cipher

Encryption has moved on in the past 2000 years however the first recorded use of encryption was with Julius Caesar in the Roman times. He wrote letters and then moved the letters forward by a set number in the alphabet. These were then decrypted using the number by the recipient.

```

public CaesarCipherer (int aKey){
    key = aKey;
}

public void encryptStream (InputStream in, OutputStream out) throws IOException{
    boolean done = false;
    while (!done){
        int next = in.read();
        if (next == -1) done = true;
        else
        {
            byte b = (byte) next;
            byte c = encrypt(b);
            out.write(c);
        }
    }
}

public byte encrypt(byte b){
    return (byte) (b + key);
}
}

```

This class ciphers the data from the in file that is passed in and writes it to the out file that is passed in. there is two methods encryptStream which reads in the data byte by byte. The encrypt method takes the byte and adds the selected amount to each byte

Main Method:

```

//
public static void main(String[] args) {

    Scanner in = new Scanner(System.in);
    try{
        System.out.print("input file: ");
        String inFile = in.next();
        System.out.print("Output file: ");
        String outFile = in.next();
        System.out.print("Encryption key: ");
        int key = in.nextInt();

        InputStream inStream = new FileInputStream(inFile);
        OutputStream outputStream = new FileOutputStream(outFile);

        CaesarCipherer cipher = new CaesarCipherer(key);
        cipher.encryptStream(inStream, outputStream);

        inStream.close();
        outputStream.close();
    }
    catch (IOException exception)
    {
        System.out.println("Error processing file: " + exception);
    }
}
}

```

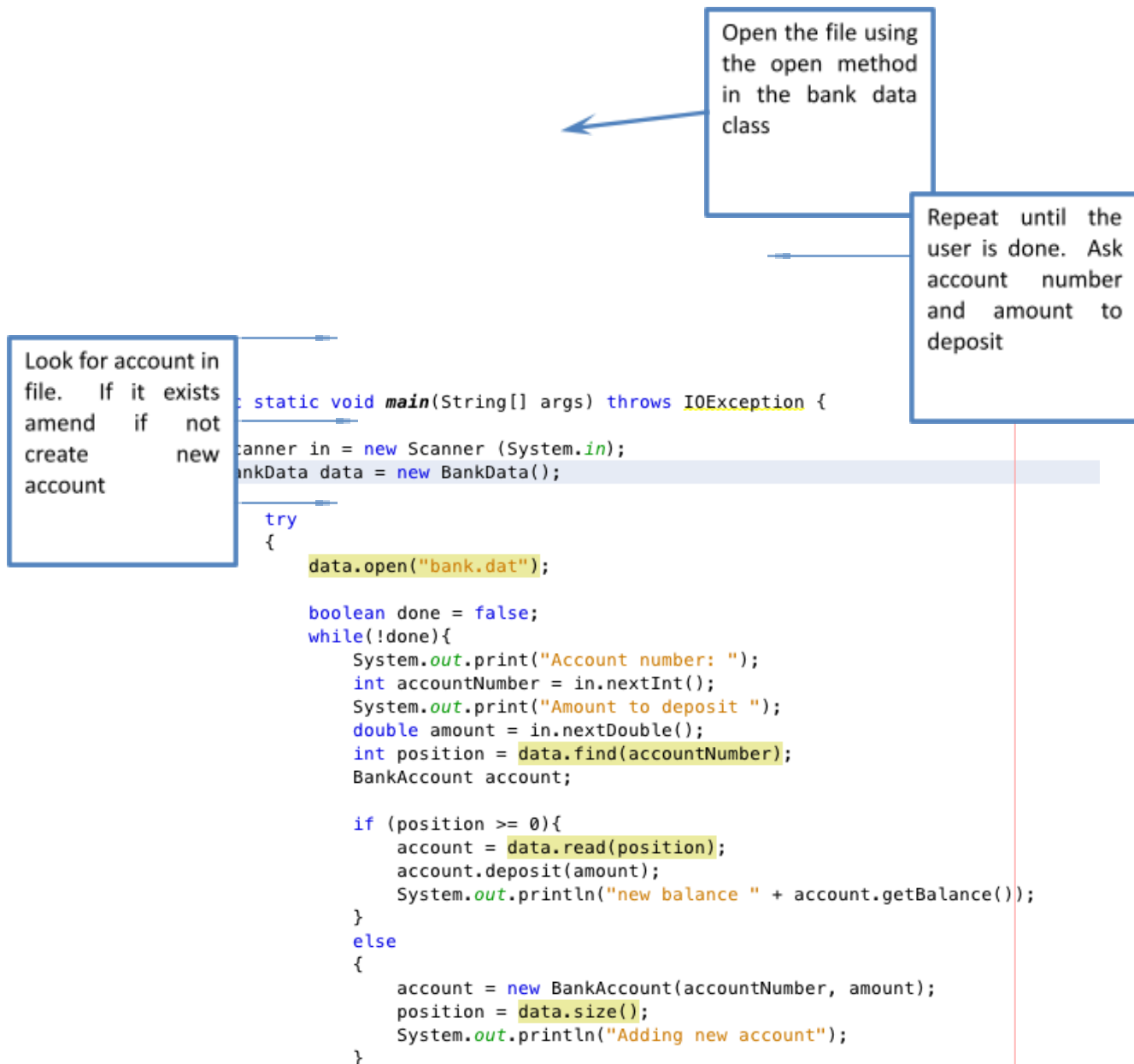
Check Your Understanding

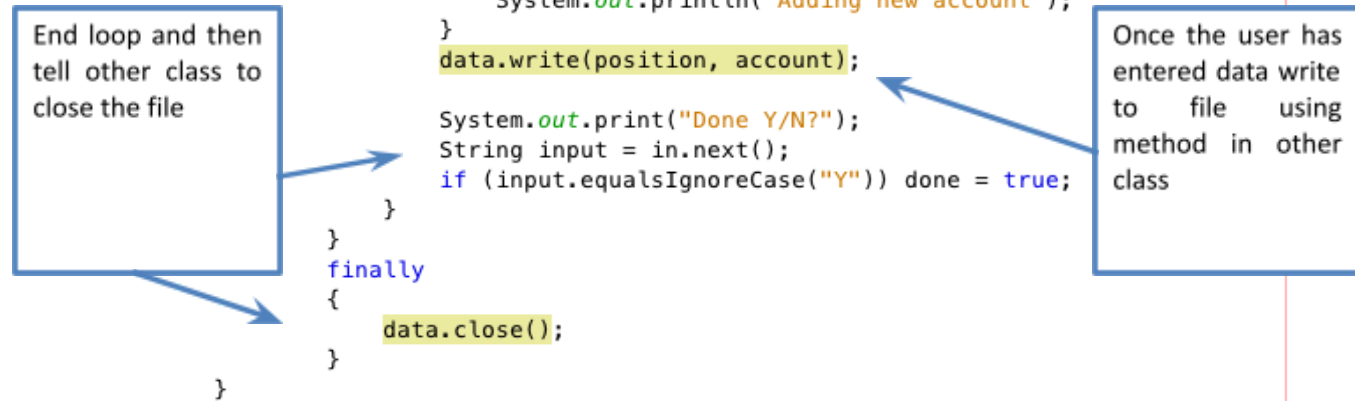
Make the Caesars Cipher program

File I/O - Random Access Files

In Caesars Cipher we looked at using the data stream tool to encrypt files (badly) we will now take this the step further and look at how we can use this to store records in a file. We can then search for these files and updates the.

Main Method:





Bank Data Method: (next page)

```

public class BankData {

    private RandomAccessFile file;
    public static final int INT_SIZE = 4;
    public static final int DOUBLE_SIZE = 8;
    public static final int RECORD_SIZE = INT_SIZE + DOUBLE_SIZE;

    public BankData(){
        file = null;
    }

    public void open(String filename) throws IOException{
        if (file != null) file.close();
        file = new RandomAccessFile (filename, "rw");
    }

    public void close() throws IOException{
        if (file != null) file.close();
        file = null;
    }

    public int size() throws IOException{
        return (int) (file.length()/RECORD_SIZE);
    }

    public BankAccount read (int n) throws IOException {
        file.seek(n * RECORD_SIZE);
        int accountNumber = file.readInt();
        double balance = file.readDouble();
        return new BankAccount(accountNumber, balance);
    }

    public int find (int accountNumber) throws IOException{
        for (int i = 0; i < size(); i++){
            file.seek(i * RECORD_SIZE);
            int a = file.readInt();
            if(a == accountNumber)
                return i;
        }
        return -1;
    }

    public void write(int n, BankAccount account) throws IOException{
        file.seek(n * RECORD_SIZE);
        file.writeInt(account.getAccountNumber());
        file.writeDouble(account.getBalance());
    }

}

```

Once it has been established that the record exists the method gets the data from it

This data attempts to find the account in the bank file

This method writes the record to the end of the file.

Check Your Understanding

Implement the above Random Access File program in Java

File I/O – Using Files with ADT's

In the Bank Account simulator we used the random access files to store and read from the files however if we had many files this would slow down the process. It is better for us to use an Array List (or similar) to store the changes temporary and then write them to file when we finish with the program.

The code I have created shows you how to make use of this feature in a very simple program. In order to be able to use this code in your internal assessments you must be able to understand it.

You could once you modify the code read the file into a linked list or a binary tree.

The Pupil Class – very simple class which allows us to create and access pupil data

```

1 import java.io.Serializable;
2
3 /**
4  *
5  */
6
7 public class pupil implements Serializable {
8
9     private int pupilID;
10    private String pupilName;
11    private String pupilClass;
12
13    public pupil (int myID, String myName, String myClass) {
14
15        pupilID = myID;
16        pupilName = myName;
17        pupilClass = myClass;
18    }
19
20    public int getID(){
21        return pupilID;
22    }
23    public String getName(){
24        return pupilName;
25    }
26
27    public String gettheClass(){
28        return pupilClass;
29    }
30 }

```

The only interesting thing is the addition of serializable

The Main Method (some of this could be moved to another class similar to bank data)

```

ArrayList<pupil> myPupils = new ArrayList<pupil>();
Scanner console = new Scanner(System.in);

File f = new File("pupillist.dat");
if (f.exists()){
    ObjectInputStream in = new ObjectInputStream (new FileInputStream(f));
    myPupils = (ArrayList<pupil>) in.readObject();
}
else{
    // adds some pupils to a file if none exist so far
    myPupils.add(new pupil(001, "Paul", "IB"));
    myPupils.add(new pupil(002, "Linz", "IB"));
}

int input = 0;

do {
    System.out.println("what would you like to do?");
    System.out.println("1 - add a pupil");
    System.out.println("2 - view a pupil");
    System.out.println("3 - see all pupils");
    System.out.println("4 = remove a student");

    if (choice == 1){
        int id;
        System.out.println("enter their ID");
        id = console.nextInt();
        String nameOf;
        System.out.println("enter their name");
        nameOf = console.next();
        String classOf;
        System.out.println("enter their class");
        classOf = console.next();
        myPupils.add(new pupil(id, nameOf, classOf));
    }

    else if (choice == 2){

        String search;
        Iterator<pupil> a = myPupils.iterator();

        System.out.println("who are you looking for?");
        search = console.next();
        boolean found = false;
        while(a.hasNext() && found == false){
            pupil temp;
            temp = a.next();
            if (temp.getName().equals(search)){
                System.out.println("Pupil ID: " + temp.getID());
                System.out.println("Pupil Name: " + temp.getName());
                System.out.println("pupil Class: " + temp.gettheClass());
                found = true;
            }
        }
    }
}

```

Create an array list and read from file if it does not exist make it

Offer options

Add a pupil to the ARRAY LIST code

Search and display pupil from ARRAYLISTw

```

    }
    else if (choice == 4){
        Iterator<pupil> a = myPupils.iterator();
        System.out.println("Who should be removed");
        String search = console.next();

        while(a.hasNext()){
            pupil temp;
            temp = a.next();
            if (temp.getName().equals(search)){
                a.remove();
            }
            System.out.println(search + " has been removed");
        }
    }
    System.out.println("to exit press 1 otherwise press 0");
    input = console.nextInt();
} while(input != 1);

```

Search and
remove person
from ARRAY LIST

Write the ARRAY
LIST to the file

```

ObjectOutputStream out = new ObjectOutputStream (new FileOutputStream(f));
out.writeObject(myPupils);
out.close();

```

Check Your Understanding

Create the program above, this should now be used as a template for all future programs.

Extension – Graphical User Interfaces

Text based programs are very rarely acceptable anymore due to the nature of people using computer systems. As computer systems are accessed by most people graphical user interfaces have become indispensable. In this section you will use an online resource to learn how to make a graphical user interface and then create your own quiz game.

This netbeans tutorial is very good at showing how to add functionality to a graphical user interface. You should follow the tutorial and create your own “calculator”.

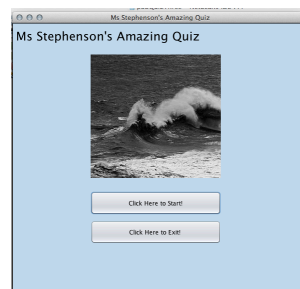
<https://netbeans.org/kb/docs/java/gui-functionality.html>

What essentially is happening is you have built an interface with a number of objects attached to it. Each object is listened too by the object listener. Once the listener detects your object has been clicked it carries out the event attached to the object.

The next stage is to take the information you now know and apply it to the quiz game. The quiz game can be split into three stages:

First Stage:

- 1: Create the start screen
- 2: add buttons and picture

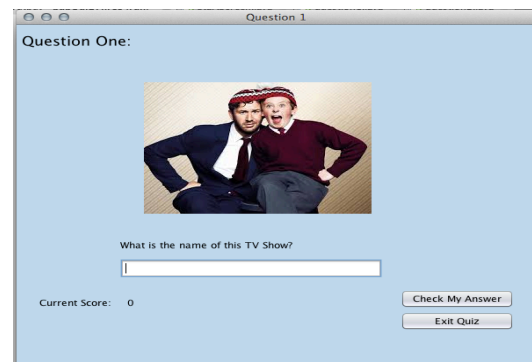


Second Stage:

- 1: Add code to the start screen
- 2: exit – system.exit
- 3: initialize score variable to 0
- 4: start button – creates and instance of q1 and opens it
- 5: sends the score to q1

Third Stage:

- 1: Create question 1 screen
- 2: Add the picture and labels
- 3: Add a text field for the answer
- 4: add a label for the current score



Fourth Stage:

- 1: Add the code to the buttons
- 2: create the iniScore method which gets the score from the other screen

On the following page is some of the code you might find useful.

On the page for question one you should initialize a variable for the score. There should then be a Score initializing method which takes in the value of score from the previous question and displays it on the screen. The code for this is shown in Figure 1 and 2 below.

```
//
public class question1 extends javax.swing.JFrame {

    int score;

    /**
     * Creates new form question1
     */
}
```

Figure One

```
public void iniScore(int x){
    score = x;
    q1Score.setText(String.valueOf(score));
}

}
```

Figure Two

The following code shows the IF statement used to check if the answer is correct or not. When the user presses the button on the question 1 screen the following IF statement is carried out. The text from the box is read into a variable called Answer. If the answer matches the correct answer a message is displayed, 1 is added to the current score and the next question is shown. Otherwise, the correct answer is displayed and the next screen is shown.

```
private void q1AnswerButtonMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    String Answer = q1Answer.getText();
    if (Answer.equals("moone boy")){
        Component frame = null;
        JOptionPane.showMessageDialog(frame,"Correct!!! ", "Well Done!", JOptionPane.WARNING_MESSAGE);
        score = score + 1;
        question2 q2 = new question2();
        q2.setVisible(true);
        this.setVisible(false);
        q2.iniScore(score);
    }
    else {
        Component frame = null;
        JOptionPane.showMessageDialog(frame, "Incorrect - it was moone boy!", "whoops", JOptionPane.WARN
        question2 q2 = new question2();
        q2.setVisible(true);
        this.setVisible(false);
        q2.iniScore(score);
    }
}
```

Finally on the start screen we set the initial value for the score variable that will be passed from frame to frame. Also when the button is clicked the quiz is started. When the exit button is pressed the game ends.

```
public class StartScreen extends javax.swing.JFrame {  
  
    int score = 0;  
  
    private void startButtonMouseClicked(java.awt.event.MouseEvent evt) {  
        // TODO add your handling code here:  
        question1 q1 = new question1();  
        q1.setVisible(true);  
        this.setVisible(false);  
        q1.initScore(score);  
    }  
  
    private void exitButtonMouseClicked(java.awt.event.MouseEvent evt) {  
        // TODO add your handling code here:  
        System.exit(0);  
    }  
}
```

Additional Projects – HL

The following section contains ideas for additional projects. Some can only be implemented if you have reached higher level but there are plenty for everyone to attempt. You should attempt as many as you can in order to get your programming skills up to scratch. A recommendation would be that you make a plan before you start to code.

Project One – Penguins with problems (additional file in appendix one)

This program will make use of files and Linked Lists to create a program which gives presents the user with a selection of logic problems which they must answer before moving on. The program will also store a list of users and their scores. It will save the users progress and will display top scores when asked.

Here is how I envisage the pseudocode for the program. It is up to you how you implement the code. If you managed to create this it is around the level expected for your internal assessment.



```

1: Display Menu
    if user selects option 1
        enter game
        load game file
        load player file
    else if user selects option 2
        display top scores and progress
        load player file

    else
        exit
  
```

Game:

```

1: Ask user name
2: add player to array list
3: set temp score variable to 0
4: Do
4.1   display task
4.2   get answer
4.3   check answer
4.4       if answer is correct
4.5           mark as complete
4.6           display next question
4.7           add one to levels variable
4.8       else
4.9           redisplay task
4.10  until user presses X
  
```

Scorer

```

1: Load File
2: For each item in array
3: Display name and scorer
  
```

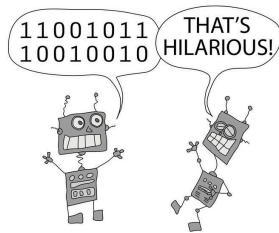
Class Diagrams:

Task
Int lvlNumber String intro String question String options String correctAns
getNumber getIntro getOption getCorrectAns

Player
String firstName String surname Int score Int levels Double percComplete
getName() getFName() getScore() setScore() setLevel() setPercComp()

List of tasks are in appendix one.

Project Two – Binary Converter



Binary to Decimal and Back Converter – Create a convertor that changes binary numbers into decimal numbers. To make the program even more useful the program can change decimal numbers into binary. If you are unsure how these numbers are connected – ask!

Tools recommended: An algorithm and lots of IF statements.

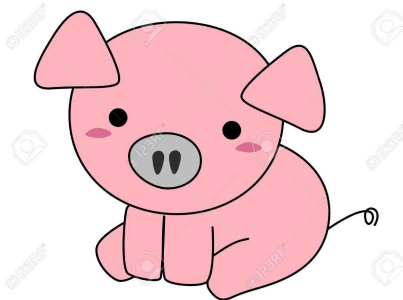
Project Three – Distance Calculator

Distance Between Two Cities – Calculates the distance between two cities and allows the user to specify a unit of distance. I would limit this to major cities in one country to start.

Tools recommended: An 2D array would probably be a good idea.



Project Four – Pig Latin



Pig Latin – Pig Latin is a game of alterations played on the English language game. To form the Pig Latin form of an English word the initial consonant sound is transposed to the end of the word and an ay is affixed (Ex.: "banana" would yield anana-bay). Read Wikipedia for more information on rules.

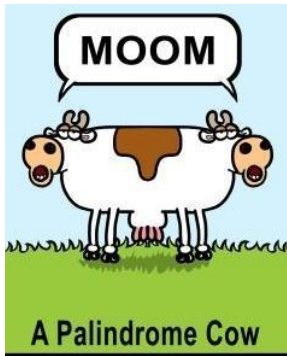
Tools recommended: the string class

Project Five – Vowel Counter

Count Vowels – Enter a string and the program counts the number of vowels in the text. For added complexity have it report a sum of each vowel found.

Tools recommended: the string class.





Project Six – Palindrome Checker

Check if Palindrome – Checks if the string entered by the user is a palindrome. That is that it reads the same forwards as backwards like “racecar”

Tools recommended: the string class

Project Seven – Fortune Teller

Fortune Teller (Horoscope) – A objects to randomly generate a Remember that horoscopes and generic so everyone thinks it Tools recommended: a linked list



program that uses horoscope. should be nice relates to them. or similar.



Project Eight – Random Gift Suggestor

Random Gift Suggestions – Enter various gifts for certain people when you think of them and when its time to give them a gift (xmas, birthday, anniversary) it will randomly pick one and perhaps places you can get it.

Tools recommended: A linked list or similar.

Project Nine – Recipe Manager

Recipe Creator and Manager – Create a recipe class with ingredients and a put them in a recipe manager program that organizes them into categories like deserts, main courses or by ingredients like chicken, beef, soups, pies etc.

Tools recommended: A linked list or similar.



Project Ten – Vending Machine



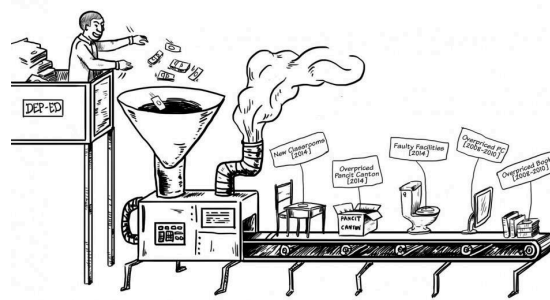
Vending Machine – Create an application which takes money and dispenses various types of candy or other item. The user enters a number and letter sequence, like D9, and have it return an instance of “Item” which of the proper type. Example when they press D9 it will return a type of candy bar which is an instance of dairy milk.

Tools recommended: A linked list or similar and 2D arrays.

Project Eleven – Josephus Problem

Josephus Problem – Create a program which links together various node objects and then every Nth object is removed until you have one object left. This last object is the sole survivor. Look it up on Google under “Josephus Algorithm”

Tools recommended: A linked list or similar.



Project Twelve – Family Tree Creator



Family Tree Creator – Create a class called “Person” which will have a name, when they were born and when (and if) they died. Allow the user to create these Person classes and put them into a family tree structure. Print out the tree to the screen.

Tools recommended: A linked list or similar and files.

Project Thirteen – Doctor

Patient / Doctor Scheduler – a doctor class. Have a doctor patients and setup a scheduling can only handle 16 patients
Tools recommended: A linked



Scheduler

Create a patient class and that can handle multiple program where a doctor during an 8 hr work day.
list or similar and files.

Project Fourteen – The Password Safe



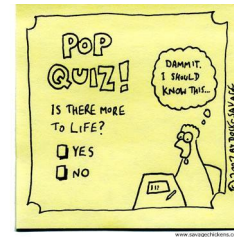
Password Safe – A program which keeps track of passwords for sites or applications and encrypts them with a key so that no one can read them.

Tools recommended: A linked list or similar.

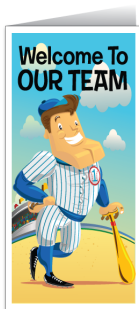
Project Fifteen – Quiz Maker

Quiz Maker – Make an application which takes various questions from a file, picked randomly, and puts together a quiz for students. Each quiz can be different and then reads a key to grade the quizzes.

Tools recommended: A linked list or similar and files.



Project Sixteen – The Collector



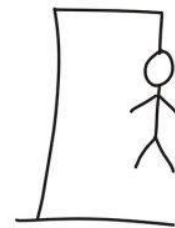
Baseball / Other Card Collector – Create an online application for keeping track of a collection of cards. Let the user enter all cards in a set, check off which ones they have, which ones they need and generate lists of cards they are looking for. For extra complexity, have it sum up sets and generate reports on how close they are of completing sets or the current value of a set.

Tools recommended: A linked list or similar and files.

Project Seventeen – Hangman

Hangman – Randomly select a word from a file, have the user guess characters in the word. For each character they guess that is not in the word, have it draw another part of a man hanging in a noose. If the picture is completed before they guess all the characters, they lose.

Tools recommended: A linked list or similar and files.



Disclaimer:
No stick men were harmed
in the making of this game.

Project Eighteen – Play Your Cards Right



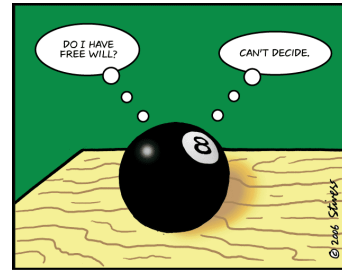
High / Low Number Guessing – Randomly choose a number in a range of numbers and have the user guess. If their guess is too low, tell them “too low” and if their guess is too high tell them “too high”. Let them continue to guess until they get the right answer. Keep track of the number of guesses they made before getting it right.

Tools recommended: lots of IF statements.

Project Nineteen – Magic Eight Ball

Magic Eight Ball – Create a file full of random sayings and answers. Then have the magic eight ball pick one at random in response to a user's question. Allow the user to enter the question and then show them the randomly picked answer.

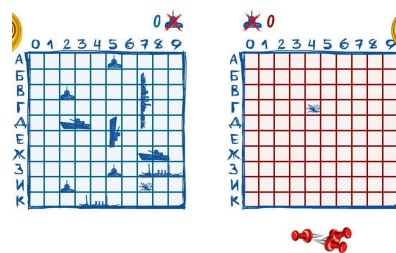
Tools recommended: A linked list or similar and files.



Project Twenty – Battle ships

Battleship – Create two game boards and let each player place a number of war ships. Each player can't see the other person's board. They then take turns firing at one another by guessing one of the board squares. If the square they guess contains part of a ship, it is a hit. Otherwise it is a miss. They sink a ship when all squares containing that particular ship have been uncovered. The player wins when all their opponents' ships have been sunk.

Tools recommended: A 2D array.



Project Twenty One – Grand National

The grand national is a famous horse race in the UK that happens every April. Your task for today is to create a computerized version that makes use of libraries, arrays of classes and decisions to create a game that people can play.



There are three levels to the game, as always start simple and get the rest working afterwards:

Simple (everyone should achieve)

- your game will allow you to enter the details of five horses
- your game will allow five players to choose the first, second and third place horse
- your game will allow the user to display a list of the horse names and odds
- your game will randomly pick a first, second and third place horse
- your game will print out what the five players have chosen and you manually check the horses

Enhanced (some of you might need more help to get here)

- your game will include all functions of the simple game
- your game will automatically check to see if any of the players checked correctly
- your game will print out a list of players and correct matches

Advanced (more challenging)

your game will include all functions of the simple game and the enhanced game
 your game will calculate the winnings based on the bet the user placed and the odds the horse had
 your game will print out the winnings for each player

Information stored about the horse:

- 1: the number
- 2: the name
- 3: the odds for first place
- 4: the odds for second place
- 5: the odds for third place

information stored about the player

- 1: the players name
- 2: their guess for first
- 3: their guess for second
- 4: their guess for third
- 5: the value of their bet

Project Twenty Two – Poker

In this project, you will implement a simulation of a popular casino game usually called video poker. The card deck is shuffled. You need to devise a fair method for shuffling. (it does not have to be efficient). Then the top five cards of the deck are presented to the player. The player can reject none, some, or all of the cards. The rejected cards are replaced from the top of the deck. Now the hand is scored. Your program should pronounce it one of the following:



Commons.wikimedia.org

No Pair – the lowest hand, containing five separate cards that do not match up to create any of the hands below.

One pair – two cards of the same value, for example two queens

Two pairs – two pairs e.g. two queens and two fives

Three of a kind – Three cards with the same value, for example three queens

Straight – five cards with consecutive values

Flush – five cards of the same suit

Full house – three of a kind and a pair

Four of a kind – four cards of the same value

Straight flush – A straight and a flush: Five cards with consecutive values of the same suit

Royal flush – the best possible hand in poker, A 10, jack, queen, king and ace all of the same suit.

Chapter Seven / Eight - Review Questions

An online music streaming service stores details of artists in a system which can be used to display information on the screen.

The current system uses an online database which the company finds clumsy and unnecessarily complex. As part of the programming team which is developing the new system you have decided to use an Object – Orientated programming (OOP) approach. The team will be divided into sub groups.

The program development team are aware of the following requirements:

Certain songs are classified as **starSongs** these songs make 25cents per stream and can be set to have heavy rotation list (this means they get played more often in a playlist set to random). Other songs are classified as **albumSongs** these songs can not be set to heavy rotation and only make 15 cents per stream. Album songs can be set to skipped, so they do not feature in any playlist.

No album can contain no more than 5 starSongs and no more than 15 albumSongs.

```
public class Artist {

    private int artistID;
    private String artistName;
    private String artistOrigin;

    public artist (int id, String theName, String origin){
        this.artistID = id;
        this.artistName = theName;
        this.artistOrigin = origin;
    }
    public int getArtistID(){ return artistID; }
    public String getArtistName(){ return artistName;}
    public String getArtistOrigin() return artistOrigin;}
}

public class Song {

    private String songTitle;
    private double songPayment;
    private int songDuration;

    public song (String title, double pay, int duration){

        this.songTitle = title;
        this.songPayment = pay;
        this.songDuration = duration;
    }

    public String getTitle(){ return songTitle; }
    public double getPayment(){ return songPayment;}
    public int getDuration(){ return songDuration;}
}
```

Question 6 Continued:

```

public class Album {

    private String title;
    private artist theArtist;
    song [] theStars = new song [5];
    song [] theAlbumTracks = new song [15];
    private int stars = 0;
    private int albums = 0;

    public album (String theTitle, artist Artist){

        title = theTitle;
        this.addArtist(Artist);
    }

    public void addStarTrack(song x){

        theStars[stars] = x;
        stars ++;
    }
    public void addAlbumTrack(song x){

        theAlbumTracks[albums] = x;
        albums ++;
    }
    public artist getArtist(){
        return theArtist;
    }
    public void addArtist(artist x){

        theArtist = x;
    }
    public void printAllAlbum(){

        // code to be completed
    }
}

```

- a) Describe, with the aid of a UML diagram how the song class could make use of inheritance. [4]
- b) Describe how the use of encapsulation is of benefit to the development team. [2]
- c) In the main class an array catalogue is used to store the albums available in the music system. There is a counter variable that records the number of album currently in the catalogue.

A new album is to be added to catalogue. The name of the album is "unicorn" by the Potato Army (ID006) who are from Switzerland.

[4]

State the code required to add this band into Catalogue.

- d) Describe the purpose of a constructor method, using the code to help you. [2]

- e) It has been decided that the `public void printAllAlbum()` method in the `Album` class should print the name of the Album, the name of the Artist, and all tracks from the Album. Write the code for this method. [6]
- f) All of the albums are stored in a `Catalogue` array. Construct a method `showArtists()` that could be used in the `main` class which accepts `artistID` as a parameter, then uses this to return albums by the artist. [6]

[End of Standard Level Paper]

- 7: Currently the software uses a linked list to store the list of songs the user places into a playlist.
- a) The playlist should be able to provide a shuffle function. Explain why using an array would be beneficial when providing a shuffle feature. (You may use a diagram to illustrate) [2]
- b) Construct the method `addToPlaylist()` which will remove the songs from the linked list `currentSongs` and place them in an array called `Playlist`. [4]

Chapter Nine – Internal Assessment

Essential Understanding: “Computational thinking and the analysis skills learned in Systems Fundamentals unit have real world applications.”

Key Questions:

How can I use object orientated programming to solve a specific problem?

How can I use the skills I have learned about gathering information in a practical way?

Is there ever a perfect system?

What are the limitations of the skills we have learned?

ATL Skills:

Social: Take responsibility for ones own action plan.

Self Management: Plan long term and short term assessments in order to meet deadlines. Set goals that are challenging and realistic.

Thinking: Apply skills and knowledge in unfamiliar situations.

Why is it important?

The internal assessment is an individual project that is undertaken to demonstrate your knowledge of computational thinking and system fundamentals. It is designed so you can showcase the skills you have learned in these units.

At standard level the Internal assessment is worth 30% of your overall grade. At higher level it is worth 20%. Both courses are expected to spend around 40 hours on it. You can use more time but this will be outside of class and you should take care to ensure you do not infringe any academic honesty agreements.

This document is designed to help you through the process. There are additional help forms in the appendices should you need them.

What is it?

The Internal assessment consists of a **Product** and **development documentation**. It is assessed using the following criteria:

- Planning
- Solution Overview
- Development
- Functionality and extensibility of product
- Evaluation

Choosing a topic

The project that you choose for your internal assessment **MUST** have an identified client.
Examples of this from previous years include:

The history department (middle school) – creating an app quiz to help students quickly test their knowledge

IB economics students – creating a program to help them draw graphs quickly and easily

Assistant principal of the school – a program to keep a record of academic periods missed by students due to absence

School secretary – create a program to identify free teachers in any given block and to assign them cover lessons

It might be useful at this point to complete the “choosing a topic sheet”. This will help you to identify your strengths as a programmer and identify a client.

Once you have identified possible projects complete the proposal sheet and hand it to your supervisor. **ONLY** when your supervisor has agreed your topic and your identified client has agreed to by your client can you proceed with your internal assessment.

TOK Links

Reasoning – you need to use reason as a way of knowing in order to reach a workable solution for your client.

Hopefully your final solution can provide new knowledge to the classes shared knowledge.

Academic Honesty

It is essential that all work completed in this internal assessment is your own. You can reference other materials (all programmers do) and use your course notes to help you but this must be clear in the text and in your bibliography. Students have been found cheating in the past and dealt with the consequences.

Below is a list of behaviours which will be deemed as academic dishonesty with regards to this piece of work. The list is not exhaustive, if in doubt ask!

- 1: Copying code from the Internet – you can be influenced by the code of others however it must be made clear using comments in your code and appear in your bibliography.
- 2: Working with others – I know the standard of programming each of you have, it is fairly clear when you are producing work that is not your own
- 3: Copying the exemplar – everyone has their own writing style mine has been provided as an *example* use your own style to describe what you have done.
- 4: Getting help from others – it is an assessment others in the class can not help you, ask me and I will help.

Planning Section - Marks

In this section you will:

- fully describe the problem
- identify your supervisor/client
- Discuss why the planned solution is suitable
- Identify success criteria
- Identify the stage you need to complete in order to be successful

You will find the planning documents in the appendices useful for this section.

Marks	Description	What this actually means
0	The response does not reach a standard described by the descriptors below.	You have done nothing
1 -2	An appropriate scenario for investigation for an identified client is stated. The rationale for choosing the proposed product is identified. The criteria for evaluating the success of the product are generally inappropriate.	You have identified a problem although it is not clear. You have a client but haven't said why they are suitable. You have said why the product you have chosen to make is good in a simple way. The criteria for evaluation are all subjective and not measurable.
3 – 4	An appropriate scenario for investigation for an identified client, providing evidence of consultation, is stated. The rationale for choosing the proposed product is partially explained and includes some appropriate criteria for evaluating the success of the product.	You have a challenging problem and have a client. You have shown that you have discussed the problem with client. You have explained why your planned product is a suitable product for the problem. Most of your criteria for success is measurable and not subjective.
5 -6	An appropriate scenario for investigation for an identified client, providing evidence of consultation, is stated. The rationale for choosing the proposed product is justified and includes a range of appropriate criteria for evaluating the success of the product.	You have identified a challenging problem that would genuinely be useful for the identified client. You have discussed the problem with the client and it is clear from your consultation you have used what the client said to help you with analysis of the problem. The reason why you have chosen the solution to the product is clear and you have identified reasons for using a computerized solution and justified your choice of software. Your success criteria are measurable and you have a clear idea about how you can test the objectives.

Solution Overview - Marks

In this section you will:

- provide a record of your tasks
- provide a system flow or higher level pseudocode
- show class diagrams
- show UML diagrams
- Have pseudocode for complex algorithms
- Have drawings of the user interface
- Fully plan testing

You will find the “overview” documents in the appendix useful

Marks	Description	What this actually means
0	The response does not reach a standard described by the descriptors	You have done nothing
1 – 2	The record of tasks and the design overview, including an outline test plan, are limited. From this information it is difficult to see how the product was developed.	You have provided a record of tasks, however they do not fully show how the product has been developed. The test plan does not show how the product will be tested effectively.
3 – 4	The record of tasks and design overview, including an outline test plan, are partially completed. They provide a basic understanding of how the product was developed	The record of tasks shows clear thinking and a clear development of the product. There is some evidence of clear planning with class diagrams etc. however it is not clear how the product works together. The testing plan shows clear tests and from these tests the product could be tested effectively.
5 - 6	The record of tasks and the design overview, including an outline test plan, are detailed and completed. From the information it is clear how the product was developed.	The record of tasks is comprehensive. It shows evidence of consultation with the client throughout. It shows how the product was developed and discusses any unexpected problems with the development. All documents associated with designs are clear and how the program functions is evident (this includes class diagrams, uml diagrams, pseudocode and a system flow diagram). The user interface is clear. The testing plan shows clearly how the product will be tested, this includes end user feedback and 1 or 2 tests per objective.

Development - Marks

In this section you will:

- Identify techniques used to develop the product (algorithms, data structures etc.)
- Explain the techniques used

You will find the “development” documents in the appendices useful.

Marks	Description	What this actually means
0	The response does not reach a standard described by the descriptors below.	You have not done anything.
1 – 4	The use of techniques demonstrates a low level of complexity and ingenuity or does not address the scenario identified in the criterion A. It is characterized by limited use of existing tools. There is no explanation of why the techniques are used or how they are adequate for the task. Sources are used but not identified.	The program is very simple and you have not used many of the techniques we have learned in class. it also does not do the job you said it would do. You have not said why you have used the techniques you have.
5 – 8	The use of techniques demonstrates a moderate level of complexity and ingenuity in addressing the scenario identified in criterion A. It is characterized by some appropriate use of existing tools. There is some attempt to explain the techniques used and why they are adequate for the task. All sources are identified.	The program uses some of the techniques we have used in class but mainly the simple ones. The program does do what you said it would do. You have discussed which tools you have used and said why you used them.
9 – 12	The use of techniques demonstrates a high level of complexity and ingenuity in addressing the scenario identified in criterion A. it is characterized by the appropriate use of existing tools. The techniques are adequate for the task and their use is explained. All sources are identified.	The program demonstrates many of the techniques learned in class using more complex ones. The program does what you said it would do and it is obvious that you have discussed this with the client. You have justified why you have used the tools, discussing any weaknesses when present.

Functionality and Extensibility of Product - Marks

This section shows:

- how well the program functions
- how the product can be modified
- how the product uses complex techniques

you will find the “making the video” document at the back of this guide very helpful in this section.

Marks	Description	What this actually means
0	The response does not reach a standard described by the descriptors below	You have done nothing
1 – 2	The video shows that the product functions partially. Some expansion and modification of the product is possible but difficult.	You have covered how the program works but you only show it running a couple of times and not handling exceptional data. It can be seen how the product can be expanded but it would be difficult / it is not explained.
3 – 4	The video shows that the product functions well. Some expansion and modifications of the product are straightforward.	You have shown how the program functions with regular data and exceptional data. You have shown how the product can be expanded and explained it clearly.

Evaluation - Marks

In this section you need to have:

- your completed testing plan
- a discussion about how well your program functions
- analysed feedback from your client
- recommendations for future improvements

you will find useful documents in the “evaluation” appendix.

Marks	Description	What it actually means
0	The response does not reach a standard described by the descriptors below.	You did not do anything
1 – 2	There is a limited attempt to evaluate the product against the success criteria identified in Criterion A. There is limited evidence of feedback from the client/adviser and any recommendations for future improvement are trivial/unrealistic.	You have discussed the how well the product meets the objectives. You don't discuss what your client thought and your recommendations would not add any functionality to the product.
3 – 4	The product is partially evaluated against the success criteria identified in criterion A including feedback from the client / adviser. Recommendations for further improvement of the product are largely realistic.	You have discussed how the product meets the objects. You have discussed a little what the client thought but it is evident that you did not ask useful questions. You have discussed how the product could be improved.
5 – 6	The product is fully evaluated against the success criteria identified in criterion A including feedback from the client / adviser. Recommendations for further improvement of the product are realistic.	You have discussed fully how the product met the success criteria discussing both successes and areas for improvement. You have asked the client detailed questions and have identified what they thought was good and bad. You have shown you understand the limitations of your program and you have identified areas of improvement that are realistic and would add functionality.

Choosing a Topic - Ideas Sheet

List of Strengths I have as a programmer:

Weaknesses I have as a programmer:

Proposal Form

Option 1

Client:
Problem:
Possible Solution:
Possible Techniques:
Any limitations you would need to overcome:

Option 2

Client:
Problem:
Possible Solution:
Possible Techniques:
Any limitations you would need to overcome:

Client:
Problem:
Possible Solution:
Possible Techniques:
Any limitations you would need to overcome:

You must discuss your options with your supervisor and they will guide you how to make a decision before moving on.

Planning Section Help Documents

Problem:
In this box write out the problem that you and your client have identified.
Useful words to include: Discuss, investigate, found, discovered, issue,
Client / Advisor:
In this box write out who your client and advisor will be
Rationale:
In this box write out the reason why you have chosen to create a solution, how will it help your client, why are you using java?
Useful words: consult, discuss, clear, interview, client, java, advantages, data, cross platform
Word Count: (enter here)

Solution Overview Help Documents

Test Table		
Description of the test here	The reason why you are carrying out the test here	The expected outcome here

Steps to help produce the list of tasks:

Think carefully:

- What do I need to do?
- Who do I need to speak to?
- What order should I do these in?

This is a big part of computational thinking which the assessment is assessing. Show that you have a clear idea of the *logical steps* you need to follow in order to complete the task.

Gantt Chart

Copy the stages you have in your list of tasks and paste them into the column.

You have 30 periods in class to do the IA, you may use additional periods but they must be in one of the tech rooms supervised by a Technology Teacher. Who will then sign your sheet.

These should be accounted for on your Gantt Chart.

Things in the gantt chart can and will change this should be reflected in your chart.

System Diagram

You must provide a diagram that shows how the classes interact with each other.

Use  for classes and  for files

UML Diagrams

It is *essential* you use class diagrams to show the instance variables and methods in your class. Ideally your methods should show the return types and the parameters expected.

Class Name
Instance variables (include types)
Methods (include return types and parameters)

Class Name
Instance variables (include types)
Methods (include return types and parameters)

Explanation

In this section discuss why you have structured the program in this way and why you have chosen these classes.

Useful Words: Classes, methods, interface, logic, steps

Record of Tasks

This is a requirement of the IB you must maintain a record of tasks to show what you were doing and when. It should show logical progression in your programming, show consultation with your client and show that you adapted to problems:

Task Number	Planned Action	Planned Outcome	Time Estimate	Target Completion Date	Criterion
1					
2					
3					
etc					

Development – Help Sheet

In this section you have to evidence all the skills you have shown throughout the project. In my project I had file I/O, Advanced data structures, loops, a search algorithm and IF statements. I had used sensible variable names.

This is your chance to **SHOW** the examiner what you have learned and how you have used it.

If I were you I would

- look through the resources from class, what have you made use of?
- Highlight the areas of the program where you have used computational thinking,
- (where have you used logical thinking? Procedural thinking?)

Use the word limit to explain the complexity in your program. Include explained extracts of your code.

Video Help Sheet

Your video needs to contain the following:

1 – your working program

- several runs of your program
- **once with normal test data**
- **others with exceptional test data**
- it does not matter if it shows flaws as long as you reflect on them

1 – the functionality of the program

- explain the code of the complex methods
- show files if you have them
- discuss the user interface working

3 – how does the program help people

- explain the problem
- explain how your program solves the problem

- there are examples of last years videos on schoology. I suggest you view them to get an idea of what is required.
- Plan your script carefully and watch the video back **can the examiner fully understand and appreciate your program based on what you have. If not go back and ADD more information.**

Evaluation Help Sheet

Before you start – it is essential you carry out interviews, surveys, observations etc. with your target audience. This will allow you to receive the feedback required in order to answer all the questions.

Meeting the success criteria

Copy and paste your criteria from the start

- for each discuss how you met or did not meet the criteria

Test Table

Copy and paste your test table from section 2

- add a new column “actual results”
- in this column write down what actually happened in the test, use this to inform your feedback

End user feedback

You must have conducted interviews etc. place these in the appendix

- discuss what your user thought was good
- discuss improvements your user thought were necessary
- discuss any problems your user may have had
- make reference to the relevant pages / areas in the appendices

Improvements

What do you think you could do to improve your program, if you had more time?

What do you think others with more programming skills could do to improve your program?

How would these improvements make the program better for the end user?

Recommendations for future

If your program was adopted by the end user then what could be added to make it more useful, *you do not need to be able to do these improvements*

Example Planning Section

(no more than 500 words should be used in this section)

Problem:

The languages department would like an interactive game that tested student vocabulary. They would like to be able to add vocabulary to the program as the students learn it. This would make the program more useful. At present there is not a product on the market that supports this.

My client for this project will be Mr Smith the German Teacher in my school. My advisor will be Ms Stephenson my Computer Science teacher.

Rationale:

After discussing the problem with Mr Smith it is clear that students need to be continuously reminded of the vocabulary they have learned throughout the year. However exposing students to too much vocabulary too quickly can put them off learning the language. He would like a program that presents students with vocabulary tests based on what they should know. Ideally if a student is happy with a word they should be able to mark it as something they are happy with, if a student is unhappy it should be left unmarked. (see appendix I for full interview transcript)

By putting the vocabulary in a computer program it will be easier for the students. The vocabulary can not get lost. If the vocabulary is stored on the computer it is easy for the teachers to add to the data. Due to the data being stored on the computer many people can have access to it but the teacher only needs to change it once. It can also be customized for different users.

I am using Java as it is cross platform therefore students with Macs can use the product as can students with PC's.

Success Criteria

- the program can run on both Mac computers and PC's
- the program stores all the vocabulary required for a course
- the teacher can add vocabulary
- the teacher can delete vocabulary
- students can mark whether the word is something they do not understand
- the program provides two way translation and a description

(word count 328)

Example Solution Overview Section

Testing Plan		
Test	Reasoning	Expected Outcome
Test one: Run the program on both an apple computer and a PC, ask the users to note any problems in a feedback form.	This will test that the program runs on both PC computers and mac computers. Therefore any student can run the application on their computer.	The program will run on both machines without any significant errors.
Test Two: After discussing with Mr Smith it is thought that the ability to store 500 entries would be sufficient.	The program stores all the required vocabulary for a course.	The file can store 500 entries which can be read into the Array List without an issue.
Test three: I will ask Mr Smith and his colleagues in the language B department to test the program. I will observe them adding vocabulary.	The teachers should be able to add vocabulary to the program.	The teachers can add the vocabulary to the program. Ideally they will be able to add it without too much additional help.
Test Four: I will ask Mr Smith and his colleagues in the language B department to test the program. I will observe them deleting vocabulary.	The teachers should be able to delete vocabulary from the program.	The teachers can delete vocabulary from the program. Ideally they will be able to do this without too much additional help.
Test Five: I will gather a group of test users. If a user understands the vocabulary then they can add a star. When the program is run there should be more unstarred words appearing than starred words.	I will get the students to complete a questionnaire regarding usability. Also they will have to count how many times a word reappears and how many times a starred word appears. They can also use the program easily.	The students can mark whether they understand a word or not. They can also use the program easily
Test Six: I will ask Mr Smith and his colleagues to check the translations as they appear in the file.	The program will allow two way translation	The vocabulary is accurate
Test Seven: I will run the program this will show the translation between two languages.	The program will allow two way translation.	There is two languages on the screen but the words mean the same.

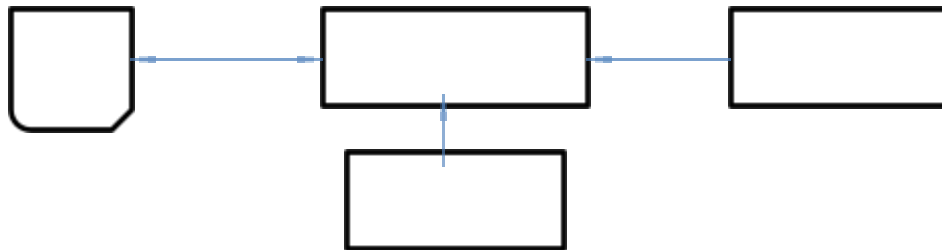
In order to complete the project I have identified the following tasks that required completing:

- 1: Collect Sample Data from the Language Department
 - 1.1 ask Mr Smith for examples of the vocabulary being stored
 - 1.2 Collect the documents
 - 1.3 Analyse the vocabulary to be stored and requirements
- 2: Identify classes required
 - 2.1 identify objects
 - 2.2 identify how objects will work together
- 3: identify methods required
 - 3.1 use requirements documentation and conversations with mr smith to help identify the methods needed
- 4: Discuss with client how the program will work
 - 4.1 apply feedback from client to design
- 5: code
- 6: test the program myself with sample data
 - 6.1 use IDE to debug program
 - 6.2 use exceptional and normal data to test program
- 7: get the code tested by target audience
 - 7.1 issue program to target audience
- 8: Conduct feedback interviews
 - 8.1 ask test users to complete survey
 - 8.2 conduct feedback interview with Mr Smith

1: Collect Sample Data from the Language Department		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1.1 ask Mr Smith for examples of the vocabulary being stored																															
1.2 Collect the documents																															
1.3 Analyse the vocabulary to be stored and requirements																															
2: Identify classes required																															
2.1 identify objects																															
2.2 identify how objects will work together																															
3: identify methods required																															
3.1 use requirements documentation and conversations with mr smith to help identify the methods needed																															
4: Discuss with client how the program will work																															
4.1 apply feedback from client to design																															
5: code																															
6: test the program myself with sample data																															
6.1 use IDE to debug program																															
6.2 use exceptional and normal data to test program																															
7: get the code tested by target audience																															
7.1 issue program to target audience																															
8: Conduct feedback interviews																															
8.1 ask test users to complete survey																															
8.2 conduct feedback interview with Mr Smith																															

Record of Tasks Form (example)

Task Number	Planned Action	Planned Outcome	Time Estimate	Target Completion Date	Criterion
1	Collect sample data	Have the sample data	1 hour	20 th August	A
2	Analyse documents	Have classes and methods	2 hours	23 rd August	B



Class Diagram – how the classes interact with each other.

UML Diagrams

MainMethod
Question <> set File inputFile
DisplayMenu() sortAlpha() displayDifficult() displayHappy() addWord() removeWord() runTest()

Question
String AWord String Eword String Defiinition String diff
setAWord(String x) getAWord() setEWord(String x) getEWord() SetDef(String x) GetDef() SetDiff() getDiff()

PseudoCode

For the more complex methods, I have wrote out the pseudocode, this will allow me to plan my methods carefully and will make coding easier.

Test Class

- 1: Ask user how many questions they would like
 - 1.1 noofQ = user input
 - 1.2 quesset = user input
- 2: Display the question
 - 2.1 IF quesSet = 1
 - 2.1 FOR 0 to noofQ
 - 2.2 get random number (o to length of array list)
 - 2.3 display qset(random number).getAdef()
 - 2.2 IF quesSet = 2
 - 2.2.1 FOR 0 to noofQ
 - 2.2.2 get random number (0 to length of array list)
 - 2.2.3 display qset(random number).getEdef()
- 3: Check answer
 - 3.1 answer = user input
 - 3.2.1 IF quesset =1
 - 3.2.1.1 if user input = gset(random number).get Edef()
 - 3.2.1.2 score = score + 1
 - 3.2.1.3 display correct message
 - 3.2.1.4 else display incorrect message
 - 3.2.2 IF quesset =2
 - 3.2.1.1. if user input = qset(random number).getAdef
 - 3.2.1.2 score = score +1
 - 3.2.1.3 display correct message
 - 3.2.1.4 else display incorrect message
- 4: Display Score
 - System print your score is = score / no of Q

Main Class

Display menu

DO

- PRINT options
- cOption = user input
 - if option 1 - call sort method
 - if option 2 - call display all method
 - if option 3 - call display difficult method
 - if option 4 - call display add word

```

        if option 5 - call display remove word
        if option 6 – call run test method
    UNTIL cOption = 0

```

Display Difficult words

```

FOR all elements in the array list
    If getDiff() at current element = *
        PRINT current element.getAWord
        PRINT current element.getEWord
    END IF
END LOOP

```

Display Easy Words

```

FOR all elements in the array list
    IF getDiff() at current element = " "
        PRINT current element. getAword()
        PRINT current element.getEword()
    END IF
END LOOP

```

Add word

```

Aword = USER INPUT
E word = USER INPUT
Def = USER INPUT
Set<> add new QUESTION(aword, eword, def)

```

Remove word

USER INPUT = removalWord

```

    FOR each item in array list
        If currentelement.getaword or currentelement.getEword = removal word
            Set <> remove current element
        END IF
    END LOOP

```

Sort Data

Sort algorithm pseudocode here

Development Example

1: Techniques Used

- a) User Interface
- b) Variables
- c) Loops
- d) If Statements
- e) Search Algorithm
- f) Object Reader / Writer

A – User Interface

The user interface I have used is a text based user interface. The main reason for this is because my client did not specify that it had to be a touch based system for a tablet. Also by having a text based system the emphasis is on the vocabulary and not the graphics.

```

choose language 1 - german, 2 - spanish, 3 - french
1
Please choose one of the following options
Option 1 - sort list alphabetically
Option 2 - show all words
Option 3 - display all my difficult words
Option 4 - Add word to list
Option 5 - Remove word from list
Option 6 - test me!!!
Option 7 - add difficulty
Option 8 - end

```

Figure 1

```

Correct!
roepwang
that answer is incorrect!
Your final score was: 1 out of 2

```

Figure 2

```

Please enter the number of questions required
2
enter 1 for aquistion to english enter 2 for english to aquistion
1
Translate this word into English: habe
enter your translation:
have|

```

Figure 3

```

Aquistion Word: habe ,English word: have ,Definition to be in possession of something ,Difficulty *
Aquistion Word: schiff ,English word: ship ,Definition ,Difficulty null
Aquistion Word: Hutte ,English word: Hat ,Definition ,Difficulty null
Aquistion Word: papier ,English word: paper ,Definition this stuff that you write on ,Difficulty *
Please choose one of the following options
Option 1 - sort list alphabetically
Option 2 - show all words
Option 3 - display all my difficult words
Option 4 - Add word to list
Option 5 - Remove word from list
Option 6 - test me!!!
Option 7 - add difficulty
Option 8 - end

```

Figure 4

```

8
Please choose one of the following options
Option 1 - sort list alphabetically
Option 2 - show all words
Option 3 - display all my difficult words
Option 4 - Add word to list
Option 5 - Remove word from list
Option 6 - test me!!!
Option 7 - add difficulty
Option 8 - end
2
Aquistion Word: oui ,English word: yes ,Definition term of agreement i have something etc ,Difficulty null
Please choose one of the following options
Option 1 - sort list alphabetically
Option 2 - show all words
Option 3 - display all my difficult words
Option 4 - Add word to list
Option 5 - Remove word from list
Option 6 - test me!!!
Option 7 - add difficulty
Option 8 - end

```

Figure 5

Figure one shows the menu that is displayed to the user when they first run the program. Figure two shows the test in action and how it deals with regular and non regular input (it accepts the input but marks it wrong). Figure 3 shows the user input when setting up the test, at the moment there is no way to stop users choosing more questions than there are words. Figure 4 shows the listing of the dictionary and figure 5 shows how irregular input it dealt with when loading a dictionary. At present the user can choose 1 to 3 however anything over 3 will load the French dictionary. Unfortunately if the user chooses a letter or symbol the program breaks.

– Variables

The variables I have used in my program vary between variables that would be easy to maintain and variables that make sense only to me. In order to make this program more maintainable I would probably change the variable names. I have when appropriate used a mix of local and global variables. All the file names are appropriate.

– Loops

loops have been used throughout the program. The main loop is to continue the program until the user presses exit.

Another while loop has been used as a search throughout the array list. For loops have been used to loop repeat the question code for the number of questions the user has asked for.

```
do {
    } while(end == false);
// repeat the following code for the number of questions
for (int y = 0; y < noOfQ; y++){

    while(a.hasNext()){
        Question temp;
        temp = a.next();
        // add difficult rating to work
        if (temp.getAword().equals(search)||temp.getEword().equals(search)){
            temp.setDiff();
        }
        else{
            i++;
        }
    }
}
```

– If Statements

IF statements have been used throughout to show determine user choices. This is in the start sections, and the menu section.

```
// option 1 sort the array list in alphabetical order by english word
if (uinput == 1){
    System.out.println("option1");
}

// option 2 display all of the words in the question array list
else if(uinput == 2){
    for (int i = 0; i < myQ.size(); i++){
        Question q = myQ.get(i);
        System.out.println (q.toString());
    }
}
// option 3 display all the words the user has marked as difficult
else if(uinput == 3){
```

– Search Algorithm

Several search algorithms have been used throughout the program. For example in option 3 to search for the difficulty words (figure 6), to search for the word to be removed (figure 7). To search for the word to add difficulty to (figure 8).

Figure 6

Figure 7

```

while(a.hasNext()){
    Question temp;
    temp = a.next();
    // add difficult rating to work
    if (temp.getAword().equals(search)||temp.getEword().equals(search)){
        temp.setDiff();
    }
    else{
        i++;
    }
}
}

```

Figure 8

2 – file input and output

I have used the object input stream and object output stream to read information into an arraylist for manipulation and then it is saved back into the file when the program is closed.

```

File f = new File(filename);
if (f.exists()){
    ObjectInputStream in = new ObjectInputStream(new FileInputStream(f));
    myQ = (ArrayList<Question>)in.readObject();
}
else{
    // test array to check functionality of program
    myQ.add(backup);
}

// this code writes the array list to the correct file so it is saved for next use.
ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(f));
out.writeObject(myQ);
out.close();

```

(word count 387)

Example Evaluation Section – Criterion E

1: Meeting the success criteria

A) the program can run on both Mac computers and PC's

As the program is java based it runs on both Macs and PC's. This is important as there is a mix of both in our school and for the program to be truly useful for the language department everyone needs to be able to use it.

B) the program stores all the vocabulary required for a course

As the program can accept input from the user, the dictionary can be expanded to include more vocabulary as required by the user or by the teacher.

C) the teacher can add vocabulary

The program has an add word function which allows the user to add a word to the dictionary as required.

D) the teacher can delete vocabulary

the program has a search and delete function which allows the user to search for a word in either English or the acquisition language and delete it as required.

E) students can mark whether the word is something they do not understand

The program has a search function which allows the user to search in English or the acquisition language and add an * to the word if they find it difficult.

F) the program provides two way translation and a description

the program stores both English and the acquisition language, the user can be tested either by seeing the English word or by their acquisition language.

2: Testing Table

Testing Plan			
Test	Reasoning	Expected Outcome	Actual Outcome
Test one: Run the program on both an apple computer and a PC, ask the users to note any problems in a feedback form.	This will test that the program runs on both PC computers and mac computers. Therefore any student can run the application on their computer.	The program will run on both machines without any significant errors.	The program is stable on both platforms. However, there are problems with the showing difficult words method, it generates an error. This is something I would need to fix.
Test Two: After discussing with Mr Smith it is thought that the ability to store 500 entries would be sufficient.	The program stores all the required vocabulary for a course.	The file can store 500 entries which can be read into the Array List without an issue.	The program does store 500 words.
Test three: I will ask Mr Smith and his colleagues in the language B department to test the program. I will	The teachers should be able to add vocabulary to the program.	The teachers can add the vocabulary to the program. Ideally they will be able to add it without too much additional help.	The teachers can add vocabulary easily after being shown once how to add a word. They found the user interface intuitive.

observe them adding vocabulary.			
Test Four: I will ask Mr Smith and his colleagues in the language B department to test the program. I will observe them deleting vocabulary.	The teachers should be able to delete vocabulary from the program.	The teachers can delete vocabulary from the program. Ideally they will be able to do this without too much additional help.	The teachers were able to delete the words easily after being shown the process once.
Test Five: I will gather a group of test users. If a user understands the vocabulary then they can add a star. When the program is run there should be more unstarred words appearing than starred words.	I will get the students to complete a questionnaire regarding usability. Also they will have to count how many times a word reappears and how many times a starred word appears. They can also use the program easily.	The students can mark whether they understand a word or not. They can also use the program easily	The students were able to use the program easily (please see questionnaire in appendix) however, there were some small problems with the display difficulty method.
Test Six: I will ask Mr Smith and his colleagues to check the translations as they appear in the file.	The program will allow two way translation	The vocabulary is accurate	The vocabulary I have entered is accurate however, there is no way to change a word once it has been added.
Test Seven: I will run the program this will show the translation between two languages.	The program will allow two way translation.	There are two languages on the screen but the words mean the same.	The translation is for the correct word, where there is no direct translation between languages another word has been substituted and noted in the description.

3: End user feedback

To obtain my end user feedback I conducted an interview with Mr Smith (see appendix), I conducted observations (see notes in appendix X), I also conducted questionnaires with students (raw feedback can be found in appendix X). For the end user feedback the overall response to the program was positive. Teachers would be able to use this product as an additional tool to help students learn vocabulary, students found it useful and liked the competitive element of the quiz. The teachers were not bothered about the text interface as

it allowed students to focus on the vocabulary however, the students would have preferred a graphical user interface. Both suggested that before it was used by all members of the school it might be useful to solve the difficulty problem or remove that function as it does not add much value to the program.

4: Improvements

The improvements I would make to this program based on my own thoughts and the end user feedback would be:

- 1 - add a search function that would display a particular entry in the dictionary
- 2 – complete the sort function so students can see the data in alphabetical order either by English word or acquisition word
- 3 it might be useful if the data was formatted better so it was easier to read
- 4 it might be good if there were a students version and a teacher version so students could not add and remove words

5: Recommendations for further development

I think it is a good start to this program, however, the following improvements could be made. I would add a function to show specific entries. I would have a start menu that asked if you were a student or a teacher and password protect the teacher area. These would be easy additions to make to the program. As previously discussed the variables, method and filenames are all logical and the code is well commented so this should be an easy job for anyone to do.

To expand the program would be simple. Firstly more languages could be added for the mother tongue students. There could be further games available for use. Verb tables could also be added to help students with the correct verb endings. Finally there is no reason why this program could stop at languages, science and maths could make use of the program by having a scientific terms dictionary.

(word count 1175)

Appendix One – Penguins File

Intro

Welcome to penguins with problems. Your task is to help Alfie the penguin get home. To begin press 1, to see top scores press 2, to exit the game press 9 (also throughout)

Question 1

Alfie says “ Help me I am trapped in a dark and cold place and can’t seem to find a way out.” To help alfie out answer the following:

Alfred and Roland have been hired to sow flower seeds. They split the field in half so they can work independently. Alfred plows the field twice as fast as Roland. However Roland sows the seeds three times faster. If sowing the plot earns 100 CHF how much should go to Roland?

- 1 – 33
- 2 – 66
- 3 – 50 *
- 4 – 25

Correct! They sowed half the field each there for half the money each. Alfie says “great! I can see a door.

Question 2

Alfie Says” oh no the door is locked”

Help Alfie through the door by answering the following question.

“The first letter of alphabet is A and the letter B comes after the letter A. However the letter you need to be worrying about is the last letter. What is the last letter of alphabet?

- 1 – z
- 2 – a
- 3 – t *
- 4 – e

Correct! The last letter of alphabet is T. Alfie has escaped the door but is confronted with a slippery tunnel.

Question 3

Alfie says “I Can see a wooden ramp, maybe I can reach it”

Help Alfie reach the ramp by answering”

SOS! 15 people are trapped aboard a ship that’s going to sink in exactly 20 minutes. Their only chance of survival is the five person life raft stowed on their vessel. To make matters worse, the waters around the ship are teeming with man eating sharks. So swimming to safety is out of the questions.

The nearest Island and back to the boat takes nine minutes of the raft. How many people will see dry land.

- 1 – 5
- 2 – 13 *
- 3 – 10
- 4 – 15

Correct! 13 people will make it. Alfie has reached the wooden walkway and is heading towards the light.

Question 4

Alfie has made it outside but there is an arctic storm. Without his puddle of penguins Alfie has no chance of survival. Help Alfie out by solving the following question:

Ten candles stand burning in a dining room. A strong breeze blows in through an open window extinguishes two of them. Checking back on the candles later, you see that one more candle has gone out. To make sure no more flames go out, you shut the window. Assuming the wind doesn't extinguish anymore candles, how many candles do you have left in the end.

- 1 – 4
- 2 – 10
- 3 – 7
- 4 – 3 *

Correct! Alfie has found a shed to shelter from the storm.

Question 5

Alfie is so cold his flippers no longer work so he needs to find some skis. If you answer the following question he should be able to find some.

A jokerless deck of 52 cards sits on the table. The cards are shuffled thoroughly and divided into two stacks of 26 cards labeled A and B. If you divided the cards as described above and check the contents of each piles 1,000 times, how many times could you expect the number of red cards in one pile match the number of black cards in the other.

- 1 – 1000 *
- 2 – 500
- 3 – 250
- 4 – 10

Correct! Alfie found some skis!

Question 6

Alfie says "oh no I can't ski and I am stuck in a snow drift"
Answer the question to help Alfie escape.

A glass jar hold a single germ. After one minute, the germ into two germs. One minute after that, the two germs each split again, forming a total of four germs. Continuing at this rate, a

single germ can multiply to fill the whole jar in exactly one hour. Knowing this, how long in minutes would it take to fill the jar if you have started with two germs?

- 1 – 59 *
- 2 – 60
- 3 – 61
- 4 – 62

Alfie has managed to get the right way up in the snow drift.

Question 7

Alfie is confused and disorientated the snow drift has not been good to him. He thinks he can get out but he needs your help.

“You only have one match left. You want to light the room with an oil lamp, start a fire to warm the room and heat your bath water. In order to complete all of the above actions, which of these should you light first?

- 1 – the fire
- 2 – oil lamp
- 3 – the bath heater
- 4 – the match *

Correct! Alfie has made it out of the snow drift.

Question 8

Alfie is walking along the road and comes to a junction. He can't decide between going north, west or east. Help Alfie make the correct decision by answering the following question.

“A father and son are chatting when the son poses the question “Dad I’m 22 now but just how old are you? The father replies you wanna know how old your old man is, eh? Hmm, well, I tell you what. I’m as old as your age, plus half my age. How old is the father.

- 1 – 44 *
- 2 – 55
- 3 – 66
- 4 – 36

Correct! Alfie turned west and managed to find the correct route.

question 9

It has been a while since Alfie ate, he reaches the sea and is wondering whether to take his chances and find some fish. Give Alfie a helping hand through answering this question.

“Three identical – looking umbrellas are sitting upright in a stand. Assuming the owners don’t check their umbrellas labels, what percentage chance is there that only two people will walk off with their own umbrella?”

- 1 – 0% *
- 2 – 33
- 3 – 66
- 4 – 100

Correct! Alfie decides to jump into the ocean to look for some tea.

Question 10

Whilst swimming in the ocean looking for tea Alfie spies a great white shark. Help Alfie reach safety by solving the following equation.

- $8 - 6 = 2$
- $8 + 6 = 2$
- $7 + 6 = ?$

- 1 – 5
- 2 – 11
- 3 – 1 *
- 4 - 13

Correct! Alfie is now hiding under a ice ledge.

Additional Story Ideas:

Alfie is back off track he has decided to try and following the sun to the west. Help Alfie find the right track by answering the following question.

12

Alfie has reached a cliff, there is no way across help alfie come up with an inventive idea by answering this question.

13

After making it down the cilff alfie is in a valley he thinks he sees someone in the distance. Help him reach them with the following

14

Alfie calls his friend but they cant quite hear him, amplify alfies voice:

15:

Unfortunately Alfies shouts have attracted a seal. Help him escape by quickly answering the following.

16

Alfie meets his friend he needs food he is exhausted.

