

IBDP Computer Science

Year 12 Revision

Option D Sample

A Truck has a Driver and a weight.

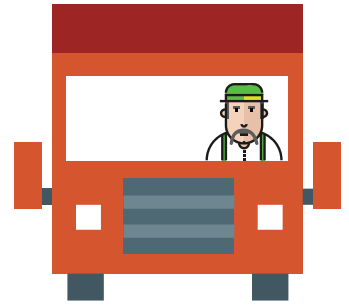
When a Truck is constructed, the total weight is set to 0.

The weight can be increased by loading items onto the Truck.

If the weight of the Truck become more than 1000kilograms, then

a horn honks and loading must stop and the driver will shout

“Let’s hit the road!”.



Here is an abstraction of a Truck in the form of a class.

```
public class Truck{

    private double weight;
    private Driver driver;

    public Truck(Driver driver){
        /*code not shown*/
    }

    public _____ getWeight(){
        /*code not shown*/
    }

    public _____ getDriver(){
        /*code not shown*/
    }

    public void loadTruck(double newItemWeight){
        this.weight = this.weight + newItemWeight;
    }

    public void soundHorn(){
        System.out.println("Honk Honk!");
    }

}
```

1

Complete the implementation of the Truck class's constructor method.

[2]

2

Complete the implementation of the Truck class's `getWeight` method.

[2]

3

Complete the implementation of the Truck class's `getDriver` method.

[2]

4

State 2 advantages of encapsulation.

[2]

5

Identify an attribute of the Truck class.

[1]

6

Identify a mutator method of the Truck class.

[1]

Here is an abstraction of a `Driver` in the form of a class.

```
public class Driver{

    private String name;

    public Driver(String name){
        /*code not shown*/
    }

    public void shout(){
        System.out.println("Let's hit the road!");
    }
}
```

7

Complete the implementation of the `Driver` class's constructor method.

[1]

8

Design an accessor method for the `name` attribute of the `Driver` class.

[2]

9

Identify the relationship between the `Truck` and `Driver` classes.

[1]

10

Draw full UML diagrams for both the Truck and Driver classes with the information currently available.

Draw a line/symbol which clearly identifies the relationship between the two classes.

[7]

11

A computer program in a separate class is being written. Here is the start of the program.

```
Truck bobsTruck = new Truck(new Driver("Bob"));  
  
//display the name of the driver (note, "Bob" is just an example)  
  
//load the Truck while the weight is less than 1000  
  
//stop loading when greater than 1000 and sound the horn  
  
//the driver says "Let's hit the road"
```

Using the comments as a guide, complete the program using Java.

Note: you can choose whatever technique you want to load the truck but you must use the `loadTruck` method of the `Truck` class.

11. contd.

[5]

12

A new `Truck` class attribute is an array of whole numbers, `weights`, which is used to store the weight of each item loaded onto a truck.

A new `Truck` class method `deaCount` will process this array and return a count of all items weighing between 50 and 75kgs, a new requirement for cargo trucks introduced by the DEA.

Implement the `deaCount` method of the `Truck` class

[3]

12. contd.

[6]

13

A collection of Trucks, TRUCKS, exists.

It has been decided that any driver in the collection of Trucks with the name "Bob" will be awarded a bonus. Write an algorithm that will display the message "Award Bonus!" for every driver in the collection with the name "Bob".

[4]

14

Note: this is a Paper 1-style question (HL)

A queue of Drivers, WAITING, exists, each one waiting to be assigned a Truck.

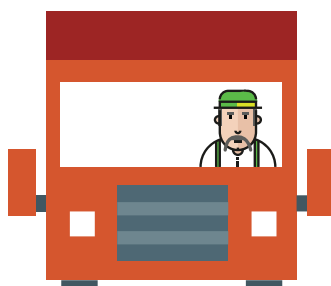
Any driver in the queue called "Bob" will be assigned to a different queue, RESERVED.

WAITING currently looks as follows:

"Chantel"	
"Bruce"	
"Bob"	
"Bob"	
"Charlene"	← front

Construct an algorithm that will empty the WAITING queue, outputting a message "Driver assigned!" each time does. If the driver is named "Bob", that driver is pushed into the RESERVED queue.

[4]



LET'S HIT THE ROAD!