

Object Oriented Programming

Constructor Methods



A hotel, Hotel Fandango in Malibu, USA, has decided to digitise its booking process.

One part of the system will be the Ticket that each guest receives when they check in.

The Ticket will display the following information.

- the name of the guest
- the room number

Of all the possible information relating to the booking, this is the information that has been abstracted for the given problem.

A software engineer will use **Object Oriented Programming** to solve this problem.

She will design the **Ticket** class to manage information about tickets.

After researching the problem, she has abstracted the following essential information to be stored in every Ticket object ie there will be 4 Ticket properties

Task 1

Start a new project in your IDE, *HotelMalibu*.

Make sure there are 2 class files in the project:

- Main.java
- Ticket.java

Copy and paste the Ticket class code on the unit page into your Ticket.java file.

Copy and paste the Mainclass code on the unit page into your Main.java file..

Task 2

It has been decided that another piece of information needs to be included in the Ticket class:

- the price of the room

Add this as a new class variable.

Edit the constructor method so that the price of the room is passed as a parameter and assigned to the new class variable.

Task 3

Now go to the Main.java file.

In the main() function, find **//TO DO 1** and write a line of code to construct a Ticket object using the data from the 3 variables.

Task 4

In the Main.java file, find **//TO DO 2** and write lines of code to display all of the properties of the Ticket, `newTicket`

Task 5

In the Main.java file, find **//TO DO 3** and write a line of code to print the ticket.

Task 6

Create 2 tickets.

Write code that will check if the 2 tickets have the same room number. If they do, the program should output an error message; otherwise both tickets are printed.

Task 7

This unit allows you to review Exception Handling.

This scenario involves a classic opportunity to include Exception Handling - user input. Take some time to review the exception handling code. Can you see how it works?

Exception Handling in Java involves try...catch... eg:

Java

```
try {  
  
    System.out.println("Enter a number: ");  
  
    int num = kbin.nextInt();  
  
} catch (Exception e) {  
  
    System.out.println("Something went wrong.");  
  
}
```

There is also another block that is often included in exception handling in Java:

Java

```
try {  
    System.out.println("Enter a number: ");  
    int num = kbin.nextInt();  
} catch (Exception e) {  
    System.out.println("Something went wrong.");  
}finally{  
  
}
```

Do some research. What is the purpose of the **finally** block in exception handling in Java? Write your answer here: