COMP503/ENSE502/ENSE602: Week 2 – Exercises

➢ Exercise 1

a) Create the following Java class in Eclipse

```
public class Book
{
	String title;
	String author;
	int numberOfPages;
}
```

In a separate class called **BookInputTest**, write a **main** method which instantiates a new **Book** object and initialises the object's attributes using keyboard input obtained from the user. Your program should provide console output:

Example output:

```
Please enter the title of the book
The Big Sleep
Please enter the author name
Raymond Chandler
Please enter the number of pages
150
The book title is: The Big Sleep
The book author is: Raymond Chandler
The book has 150 pages
```

b) Modify the **main** method you wrote in part (a) to ensure the **Book** object stores a value greater than 0 in the **numberOfPages** attribute. The console output should look like:

Example output:

```
Please enter the title of the book
A Brief History of Time
Please enter the author name
Steven Hawking
Please enter the number of pages
0
Please enter the number of pages
-1
Please enter the number of pages
239
The book title is: A Brief History of Time
The book author is: Steven Hawking
The book has 239 pages
```

COMP503/ENSE502/ENSE602: Week 2 – Exercises

➢ **Exercise 2**

Consider the following **Student** class:

```
public class Student
{
      String firstname;
      String lastname;
      String studentID;
}
```

Create constructors that take:

- three input parameters, storing the student's first name, last name, and identifier to assign to the appropriate object attributes.
- two input parameters, storing the student's first name and last name. The remaining attributes are assigned a default value.
- zero input parameters which initialises each attribute to an appropriate default value.

Create the **StudentTest** class and write a **main** method that demonstrates the usage of each Student constructor you wrote for **Student** class. Your program should

- use **Scanner** to get the student's first name, last name and ID
- Use input data to create a **Student** object using the
    - three input parameter constructor,
    - two input parameter constructor,
    - zero-input parameter constructor
- Print out the attributes of each instantiated **Student** object.

Your program should produce output similar to the following:

Example output:

```
Please enter student's first name:
Sue
Please enter student's last name:
Perman
Please enter student's ID:
012345
Calling 3 parameter constructor:
Student's first name: Sue last name: Perman ID: 012345
Calling 2 parameter constructor:
Student's first name: Sue last name: Perman ID: ?
Calling zero parameter constructor:
Student's first name: UNKNOWN last name: UNKNOWN ID: ?
```

COMP503/ENSE502/ENSE602: Week 2 – Exercises

➢ Exercise 3

Read the following English descriptions and develop a suitable Java classes that best models the scenario. Carefully choose the name of the class and define some object attributes with appropriate names and data types.

a) The **ANX** bank stores the following customer account data: the customer's name (e.g. Robin Banks) and their account code (e.g. 0820133):
   I. Write a suitable Java class to model this scenario
   II. In a separate class, write a **main** method that instantiates at least four instances of your class, with suitable values for the object attributes.

b) To use the **NitFlux** movie streaming service, a subscriber must first open an account. The account stores an email address (e.g. myaddress@mydomain.com) and the following subscription information: how many months the subscription will last (e.g. 24 months), and the amount of money paid each month (e.g. $4.99).

   I. Write a suitable class to model **NitFlux** account data
   II. In **NitFlux** account, create **computeCustomerPay** method to compute the total dollar amount the customer pays for **NitFlux** (e.g. number of months × amount each month)
   III. Write a **NitFluxTest** class with a **main** method that creates three **NitFlux** account objects. Initialise each object with suitable values. Revoke **computeCostumerPay** for each object and print the result in console.

c) Expand the **NitFlux** class you wrote. You should add object attribute(s) to store the bank account details of each NitFlux subscriber (hint: you the object of ANX class you created in part (a))

COMP503/ENSE502/ENSE602: Week 2 – Exercises

➢ **Exercise 4**

a) Consider the following **Person** class with four instance variables:

```
public class Person {
	int age;
	double weight;
	boolean student;
	char gender;
}
```

I. Write a constructor with inputs to initialise all four instance variables
II. Write a default constructor which initialises all instance variables to default values
III. Write a **main** method that reads in appropriate values from the console and uses them to construct a **Person** object. Note that the Scanner class has the following methods available for reading a variety of data types from the keyboard**: nextInt()**, **nextDouble()**, **nextBoolean()**.
You can read a character from the console using the command: **scanner.next().charAt(0)**

Your program should output the Person object's attributes and produce console output similar to the following:

```
Please enter the person's age:
65
Please enter the person's weight:
70.95
Is the person a student (true/false):
true
Please enter the person's gender (M/F):
M
Person: age: 65 weight: 70.95 retired: true gender: M
```

b) The bus company offers a person various discounts on their bus fares depending on their status. Create a **FareDiscountApp** class with a **main** method.

I. In the **FareDiscountApp** class, create another static method with the signature of **public static int calculateDiscount(Person person)** which returns the value of discount according to the following rules. If the person age is:

- over 65 years of age then their discount is 100%
- a student aged between 10 and 20 then their discount is 50%
- a female student above the age of 40 then their discount is 75%
- age is an even number then their discount is 25%
- age is an odd number then their discount is 15%

Note1: The **Person** is given the maximum discount if more than one condition applies. If no discount rule applies, then the discount is 0.
Note2: At the beginning of the body of method, declare an integer variable **int discount = 0;** which stores a percentage value of the discount (0 - 100%) and returns calculated discount value at the end.

II. In the **main** method, create a **Person** object from console input and set the value of **discount** using **calculateDiscount** method.

Sample Output:

```
Please enter the person's age:
80
Please enter the person's weight:
100
Is the person a student? (true/false):
true
Please enter the person's gender:
M
Person: age: 80 weight: 100.0 student? true gender: M
This person's discount is: 100%
```