JAVA PROGRAMMING – QUIZ 3 and EXERCISE 7

WARNING. This test is to be treated as if taken under exam conditions. Any plagiarism or collusion will result in immediate disqualification and zero marks awarded, and will be pursued as a disciplinary matter.

As explained previously, part of the coursework consists of five multiple-choice quizzes, that are designed to test your understanding of Java. They appear in weeks 5, 6, 7, 8 and 11. Each one is worth 2% of the course unit mark, must be undertaken during a lab session, that is between 2.00-4.00pm on the day you attend. Each quiz can only be taken once, and must be completed within 20 minutes of starting. You will need a password to access the quiz, which will be given during the lab.

The third of these quizzes is now available under 'Assessment' on Blackboard and should be taken NOW, before attempting the exercises below.

EXERCISE 7

In this lab, we'll be going through various tasks, which will serve to reinforce the concepts you have learnt over the past few weeks.

Instructions:

- Write, compile, run and test these programs using Notepad and command prompt, or ConTEXT.
- There is no Course Marker involved, so you can start coding right away.

Before you proceed, read the following.

Multiple Methods

```
public class Anything {

        public static void main(String [ ] args)  {
             …………………
             …………………
        }

}
```

This class has just ONE method, which is the **main** method.

You already know that this is a basic skeleton of a program.
In a program with multiple methods, the **main** method is the first to execute when a program is run :

```
public class Anything {

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

        }

        public static void calcAvg ( ) {
             ………………………
             ………………………
        }

}
```

This class contains TWO methods.
One is the **main** method (which is compulsory and must be named main), and the other one is the **calcAvg** method.

Example of using multiple methods:

```
public class Anything {

        public static void main(String [ ] args)  {
           System.out.println(“Hello”);
           System.out.println(“World”);
        }

}
```

This class has just ONE method, which is the **main** method. It runs that to print out :
Hello
World

```
public class Anything {

        public static void main(String [ ] args)  {
             sayHello ();
        }

        public static void sayHello() {
           System.out.println(“Hello”);
           System.out.println(“World”);
        }
}
```

This class has TWO methods, **main** and **sayHello**. The program will always run the **main** method first. It sees that the **main** method calls the **sayHello** method, thus executing the statements inside that method – this prints out :
Hello
World

Example of passing values between multiple classes:

```
public class Anything {
        public static void main(String [ ] args)  {
           int number1 = 10;
           double number2 = 15.5;
           sayHello(number1,number2);
        }

        public static void sayHello (int a, double b) {
           System.out.println("Number 1 is " + a);
           System.out.println("Number 2 is " + b);
        }
}
```

main has initialized an int variable called number1 and a double called number2. When it calls the **sayHello** method, it passes number1 and number2 through parameters (written in the brackets). **sayHello** accepts them as 'int a' and 'double b' and then prints them out. What happens is that number1 is copied to a and number 2 is copied to b.
Q. Would it work if a was defined as a double ? Try it!

Note: You can have multiple methods with the same name, provided they have different parameters. This is called 'method overloading'.

Example of global variables:

```
public class Anything {
  static int number1 = 10;
  static double number2 = 15.5;

  public static void main(String [ ] args)  {
        sayHello();
  }

  public static void sayHello () {
        System.out.println("Number 1 is " + number1);
        System.out.println("Number 2 is " + number2);
  }
}
```

This program works exactly like the previous example except that we don't pass values to **sayHello** using parameters. Instead, we declare the variables outside the main, as *global* variables. Global variables have a scope that extends throughout the program so can be used by all methods. Global variables will be discussed later.

Task 1:

Write a simple program called **Addition**, which prompts the user to enter two integer numbers and sums them up using an additional method called **sumUp**.

Task 2:

(I) Write a program called **Statistics**, which reads in grades for **n** students, where **n** is given by the user. The grades should have a permitted range of 0-100 and have to be stored in an integer array.

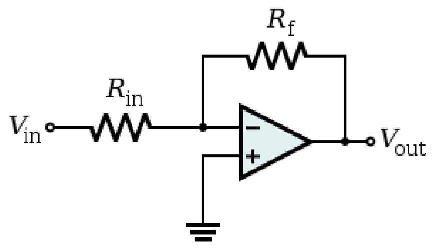
(II) Calculate the average, maximum, minimum grades using 3 different methods.

Your output shall look like:

```
Enter the number of students: 4
Enter the grade for student 1: 50
Enter the grade for student 2: 51
Enter the grade for student 3: 56
Enter the grade for student 4: 53
The average is 52.5
The minimum is 50
The maximum is 56
```

NOTE: Declare the integer array as a global variable, and pass **n** to the methods via parameters.

Task 3:



This ideal op-amp is setup in an inverting amplifier configuration.
Write a program which calculates V_{out} through an additional method called **calc**. The values of V_{in}, R_{in}, and R_f are to be passed to the method via parameters. Their values must be entered in by the user during run-time and then passed onto the **calc** method to calculate and print out V_o.

Assume: $V_{out} = -V_{in}\frac{R_f}{R_{in}}$

Task 4:

So far, we have printed out our outputs from the method itself.
It is also possible to return a value to the main method, and then print it out from there.

Example (not using global variables) :

```
public class Anything {

          public static void main(String [ ] args)  {
             int number1 = 10;
             double number2 = 15.5;
             System.out.println("The sum is: ");
             System.out.println( add (number1,number2) );
          }

          public static double add (int a, double b) {
             double sum;
             sum = a + b;
             return sum;
          }
}
```

There are some important differences here. Firstly, to return a value to **main**, the method must **not** be a 'void' type. The method **add** returns a double type here because we are returning the value of **sum**, which itself is a double. **sum** is returned to **main**, where it is printed out.

Now, write a program called **SimpleMath**, which reads in two integer numbers using a Scanner. Write two methods, called by main, where one multiplies the two numbers, and the other divides the two numbers, and each returning the result. Pass the numbers (entered by the user) to both the methods using parameters.

Task 5:
There are some additional , "advanced", java exercises, see "Week 7" under Further Exercises on Blackboard. See how many you can complete.

END