TK1114 Tutorial 9

Section A

Trace the code segments **using pen and paper** to print the output.
Note: Do not run the code segment using Eclipse or any other IDE. This task is to ensure you acquired strong understanding on the basic structure of classes.

1. Given the following programs:

```java
// Program name: Computer.java

public class Computer {
   private String brand;
   private String model;

   public Computer(String b, String m) {
      brand = b;
      model = m;
   }
   public void setBrand(String b) {
      brand = b;
   }
   public void setModel(String m) {
      model = m;
   }
   public String getBrand() {
      return brand;
   }
   public String getModel() {
         return model;
   }
   public void displayInfo() {
      System.out.println("Brand: " + brand);
      System.out.println("Model: " + model);
   }
}
```

```java
// Program name: ComputerApp.java

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

      Computer comp1 = new Computer("Apple", " iMac");
      comp1.displayInfo();
   }
}
```

a) Identify the followings:
   i. class name
   ii. attributes
   iii. instance methods
   iv. accessor methods.
   v. mutator methods.
   vi. constructor method

b) Trace the flow of the programs to display the output.

c) Define a new object named `comp2`. Create the object with the brand and model name of your own computer.

d) Modify the above programs to add an instance variable named `price` of type `double`.

2. Given the following programs:

```java
// Program name: Rectangle.java

public class Rectangle {
   private int width;
   private int height;

   public Rectangle(int w, int h) {
      ...
   }
   public int getWidth() {
      ...
   }
   public int getHeight() {
      ...
   }
   public void setWidth(int w) {
      ...
   }
   public void setHeight(int h) {
      ...
   }
   public void displayInfo() {
      ...
   }
}
```

```java
// Program name: RectangleApp.java

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

      Rectangle rect1 = new Rectangle(5, 3);
      rect1.displayInfo();
   }
}
```

a) Complete the constructor method

b) Complete the accessor and mutator methods

c) Complete other instance method.

d) Trace the programs. What would be printed?

e) Write a new method with the following declaration:

```
public void displayRectangle() {  ... }
```

that display the rectangle using '*' as discussed in tutorial 5.
For example, for width = 5, height = 3, the method should display:

```
*****
*****
*****
```

f) Write a method named `getArea()` that calculate and returns the area of the Rectangle object.

g) Write a method named `getPerimeter()` that calculate and returns the perimeter of the `Rectangle` object.

h) Modify the RectangleApp.java program to test the new methods written above.

i) Draw the UML diagram for the above classes.

Section B

1. A class called `MyTime`, which models a time instance, is designed as describe below.

   `MyTime` contains the following private instance variables:
   - `hour`: between 0 to 23.
   - `minute`: between 0 to 59.
   - `second`: between 0 to 59.

   The constructor shall invoke the `setTime()` method (to be described below) to set the instance variable.

   `MyTime` contains the following public methods:
   - `setTime(int hour, int minute, int second)`: It shall check if the given hour, minute and second are valid before setting the instance variables.
   - Setter methods:
     `setHour(int hour)`, `setMinute(int minute)`, `setSecond(int second)`:
     It shall check if the parameters are valid, similar to the above.
   - Getter methods: `getHour()`, `getMinute()`, `getSecond()`.
   - `toString()`: returns "HH:MM:SS".
   - `nextSecond()`: Update this instance to the next second and return this instance. Take note that the `nextSecond`() of 23:59:59 is 00:00:00.
   - `nextMinute()`, `nextHour()`: similar to the above
   - `previousSecond()`, `previousMinute()`, `previousHour`(): similar to the above.

   Write the code for the `MyTime` class.

   Also write a test program (called `TestMyTime`) to test all the methods defined in the `MyTime` class.

2. In this assignment, you will be practicing with the `Student` class and a `StudentTester` class. A `Student` has the following attributes and methods:

   - Four String data fields named `name`, `matric`, `college` and `phoneNum` that specify the student's information.
   - A constructor that creates a student with two parameters - `name` and `matric`.
   - A method named `setName()` that set the name of student.
   - A method named `setMatric()` that set the matric number.
   - A method named `setCollege()` that set the student's residential college.
   - A method named `setPhoneNum()` that set the telephone number.
   - A method named `getName()` that return the name of student.
   - A method named `getMatric()` that returns the matric number.
   - A method named `getCollege ()` that returns the college.

- A method named `getPhoneNum()` that returns the telephone number.
- A method named `toString()` that returns the String

Write the code for the `Student` class.

Also write a test program (called `StudentTester`) to test all the methods defined in the `Student` class.

3. In this assignment, you will be practicing with the `SmartPhone` class and a `SmartPhoneTester` class. A `SmartPhone` has the following attributes and methods:

   - Data fields named `manufacturer`, `model`, `storage` and `screenSize` that specify the smartphone's information.
   - A constructor that creates a smartphone with appropriate parameters.
   - Accessor methods
   - Mutator methods
   - A method to display the smartphone info

   Write the code for the `SmartPhone` class and `SmartPhoneTester` class.