COMP1039 – PGP - Java
Lab 5

4.1

Given the **Book.java** and **BookDemo.java**, add the package declarations to include them in the same package **bookpack**. Then compile and run the code using the command-line commands.

4.2

Modify the package declaration in **BookDemo.java**, such that it becomes a class in package **bookpack2**. Make corresponding changes to both **Book.java** and **BookDemo.java**, and then try to execute the program.

4.3

Modify the private variables in **Book.java** to make them protected. Write a class **ExtBook** in the package **bookpack2** and make it a subclass of class **Book**. Write a public method **getTitle()** which returns the value of title declared in class Book. Test whether you can call the method **getTitle** to get the value of title. Moreover, try if you can directly access the value of title in the main method.

4.4

Create a class called **Employee** whose objects are records for an employee. This class will be a derived class of the class **Person** (find it on the moodle). An employee record has an employee's **name** (inherited from the class Person), an annual **salary** represented as a single value of type double, a **year** the employee started work as a single value of type int and a **national insurance number**, which is a value of type String.

Your class should have a reasonable number of constructors and accessor methods, as well as an equals method. Write another class containing a main method to fully test your class definition.

4.5

Write an interface Moveable which contains 4 abstract methods: **MoveUp()**, **MoveDown()**, **MoveLeft()**, **MoveRight()**. Also write an implementation class called **MovePoint**. This class contains 2 variables to represent its x-axis and y-axis values. Write its constructor and implement all 4 abstract methods.

4.6

Create an abstract class **TwoDShape** and its subclasses **Circle**, **Rectangle** and **Square**. Abstract class **TwoDShape** has two variables **color** and **filled**. Both of them can be accessed by its subclasses but not the classes from other packages. The variable **color** is a string and the **filled** is a boolean value. Their default values are “white” and false respectively. Create two constructors for **TwoDShape**, one of them takes no argument and the other takes a string and a boolean value as its input. You also need to declare 4 concrete methods in class **TwoDShape**, which are **getColor()**, **setColor(String color)**, **isFilled()**

and **setFilled(boolean filled)**. These methods are used to get the current value of the TwoDShape, set it to a particular value, check if it is filled and set its filled property respectively. There are also two abstract methods: **getArea()**, and **getPerimeter()** which are used to calculate the area and perimeter for a particular TwoDshape.

We are required to write two subclasses **Circle** and **Rectangle**, both of them inherit the superclass **TwoDShape**. You are asked to write its constructor to initialize the necessary parameters such as **radius**, **width** or **height**. Moreover, you need to give detailed implementation of the abstract methods **getArea()** and **getPerimeter().**

Finally, we need to create a class **Square** which is the subclass of **Rectangle**. Similarly, we need to provide its constructor and all the methods that need to be overridden.

Math.PI can be used to represent the value of π

4.7

Given the class **Account**.java (on moodle), try to finish the following tasks.

1) Use the **Account** class as a base class, write two derived classes called **SavingAccount** and **CurrentAccount**. A **SavingAccount** object, in addition to the attributes of an Account object, should have an **interest rate** variable and a method which adds interest to the account based on the current balance. A **CurrentAccount** object, in addition to the attributes of an Account object, should have an **overdraft limit** variable. Any actions that result the current overdraft amount to exceed the overdraft limit would result in an error message. For example, suppose the overdraft limit is set to 5000 and current account balance is -4900. If we withdraw 1200 from the current account, an error message should be displayed.

2) Create a **Bank** class, an object of which contains an array of **Account** objects. **Accounts** in the array could be instances of the **Account** class, the **SavingsAccount** class, or the **CurrentAccount** class

3) Write an **update** method in the **Bank** class. It iterates through each account, updating it in the following ways: each **SavingsAccount** gets **interest** (not interest rate) added (via the method you already wrote); each **CurrentAccount** gets a letter sent if they are in overdraft.

Note that the balance of an account may only be modified through the deposit(double) and withdraw(double) methods.

The Account class should not need to be modified at all.

Be sure to test what you have done after each step.

4.8

Given an array of integer, write an insertion sort using the ArrayList.