Lab 1 – Mobile S/W Development DT228/3 – 2015/2016

Lab Work – Java primer

The purpose of this lab is revisit key concepts in Java and refresh a bit of programming.

The java API specification is at
http://docs.oracle.com/javase/7/docs/api/
This contains all classes already written in java for you to reuse.

Eclipse

We will be using Eclipse for java development in this lab. Open up Eclipse on the machine.

You may be prompted to set the workspace (this is where code will be stored). If so, just set to **your** u drive (or wherever you want to store your code)

Set up a new java project / New package/ New class

PART 1 – Person class

Create a Person class that holds the following information about a person: name and gender (either ‘M’ or ‘F’). Include a method to print the person’s name out.

Include getter and setter methods for each attribute and make sure that they are **encapsulated**.

Create a control class (i.e. has a “main” method as below) that instantiates two people objects, and prints them out.

```
public static void main(String [] args)
{
  // your code here
}
```

Run the control class

Part 2 – Person Class with more details.

Add a constructor for the Person class that initialises the two instance variables, name and gender, to values passed as parameters in to the constructor.

Add a `toString()` method to the Person class that prints out all the person's details in a reasonably formatted way. Check out the `toString()` method at the top level Object class (Java API link is available above) that this method is overriding.

Edit your *control class* from Part 1 so that it now instantiates a Person and prints out his/her details.

Run the control class.

Part 3 - Inheritance

1. Create a Student class to become a a subclass of Person. In addition to name and gender, student should hold a student id and a courseCode.

2. Add a constructor to the Student class that initialises all four instance variables to values passed in as parameters. Use the constructor from the Person superclass.

3. Add a `toString()` method to the Student class that prints out the student details using the toString() method from the Person class to format the name & gender.

Part 4 – Interfaces

1. Write an interface "PublishDetails" that contains two methods: confirmDetails(); getCourseCode().

2. Get the student class to implement the PublishDetails interface. What happens if the interface is implemented at the class level (i.e. using "implements" but the methods confirmDetails() getCourseCode() are not included in your student class?

3. For the student class, ConfirmDetails() should print information about the student name; getCourseCode() should print out the course code.