Advanced Programming 2017 – Year 2
Labwork 1: (5% - or 50 points out of 500 points for labwork this semester)

NOTE: ALL LABS TO BE COMPLETED IN PROJECTS USING ECLIPSE

Part 1 – Getting started with Eclipse projects (10 points)

Create an Eclipse Project called **Lab1Part1**. Create an 'EclipseFirstExample' java class within the Lab1Part1 project that executes and prints any string by calling a method called *printFavouriteSubject(String subject)* – where *subject* is the String parameter to print to the default output device (System.out). Fully Javadoc the program including *@param* tag for the *printFavouriteSubject(String subject)* method. Place the entire program into a JarFile called *Lab1Part1.jar*.

Required activities and marking guideline:

• Create the Eclipse project Lab1Part1	(2 points)
• Write *printFavouriteSubject(String subject)* method	(3 points)
• Add and generate the Javadoc	(2 points)
• Jar and execute the program from the jar	(3 points)

Part 2 – Multiple classes communicating in one project (10 points)

Create an Eclipse Project called **Lab1Part2**. Create three classes within this project that will communicate with each other through method calls. Put ALL three classes into a package called *numbersAndStrings*. Call the first class **NumberToStringConverter.java** and call the second class **StringToNumberConverter.java** and the third class should be called **StringConversionTester.java**. Add a method to the NumberToStringConverter.java class which will receive an integer and return (non void method) a String version of that number using a method called convertToString(int num) (Note: 1 becomes "One", only do three numbers, e.g., 1, 2 and 3). Add a method to the StringToNumberConverter.java class called convertToNumber(String str) which will receive a String (as a number, e.g., "One") and return the integer version of that number. Finally test the string-to-number and number-to-string conversions using the StringConversionTester class which should at a minimum test one integer to it's sting form and one string form of a number to it's integer form (e.g. 1 to "One" and "One" to 1). Fully Javadoc each of the classes and methods in the classes.

Required activities and marking guideline:

• Implement the three classes in Project	(3 points)
• Write the methods required	(3 points)
• Add and generate Javadoc	(2 points)
• Run the test class StringConversionTester	(2 points)

Part 3 Multiple packages communicating in one project (10 points)

Create an Eclipse Project called **Lab1Part3**. Create two classes, within two separate packages within the project. Call the first class **MathHelper.java** and place it in a package called *mathematics*. Call the second class **Application.java** and place it in a package called *application*. Add a static method to the MathHelper.java class called toThePowerOf(int x, int y) that receives two integers and returns (non void method) the result of the first number to the power of the second number. Add a main method to the Application.java class and call the MathHelper toThePowerOf method with two test values. Finally, Jar this project and run it outside Eclipse using the jar file. Javadoc all classes and methods.

Required activities and marking guideline:

• Implement the two classes	(2 points)
• Place the classes in their packages	(2 points)
• Write the multiply method in MathHelper.java	(2 points)
• Test MathHelper toThePowerOf method in Application.java	(2 points)
• Jar and Javadoc (run the Jar file)	(2 points)

Part 4 Putting it all together week 1 (Jars, Packages, IDE) (20 points)

Create an Eclipse Project called **Lab1Part4**. Create a simple JFrame GUI with a two JLabel's and two JTextField's and one JButton at the bottom. Call the frame **StringNumberConversionFrame.java**. Set the text of the first JLabel to "Enter an Integer to convert to String: ". Set the text of the second JLabel to "The text version of the number entered is: ". Add listeners and handlers so that when the user inputs the integer to convert in the first JTextField and the JButton (labeled "Convert") is pushed the String version of the integer must be returned and displayed in the second JTextField (number 1, 2 and 3 is enough!). Place the GUI JFrame class in a package called *gui* and place the conversion class in a package called *numbersAndStrings*. Finally Javadoc all classes and methods used, Jar the project and run it from the Jar file (Note: you may re-use some of the code written in Part 2 above if you wish: Note: in fact if you Jar Part 2 of this lab you can import the jar into this project including using Project->Properties->Libraries->Add External Jar and call the methods written above!).

Required activities and marking guideline:

• Create GUI with 2 labels and 2 fields and button (with layout)	(5 points)
• Add listeners and handlers	(4 points)
• Place classes in packages	(4 points)
• Javadoc all of the project	(3 points)
• Jar the project (include Javadoc in the jar)	(2 points)
• Run the project from the Jarfile	(2 points)