Advanced Programming 2021 – Year 2
Labwork 7: (This lab is worth 60 points out of 500 points – there are FIVE parts to this lab)

NOTE: ALL LABS TO BE COMPLETED IN PROJECTS USING ECLIPSE OR EQUIVALENT IDE (NO MORE TEXTPAD, EVER EVER!!!)

IMPORTANT NOTES:

- **NO COPYING PERMITTED AND ZERO MARKS WILL APPLY TO COPIED WORK. FURTHER ACTION MAY BE TAKEN AGAINST STUDENTS THAT HAVE BEEN FOUND TO COPY WORK.**

- **ASSESSMENT WILL INVOLVE ONE-TO-ONE QUESTIONS ABOUT YOUR SUBMITTED WORK. USE COMMENTS IN YOUR CODE TO ENSURE YOU DON'T FORGET WHY YOU WROTE CODE YOU MAY LATER BE ASKED ABOUT.**

- **ALL WORK MUST BE SUBMITTED TO MOODLE BY DATES SPECIFIED (2 LABS SUBMISSIONS OF FIVE LABS THROUGHOUT THE SEMESTER).**

- **MANY OF THE TASKS ASSIGNED BELOW CAN BE COMPLEX AND\OR THE DESCRIPTIONS MAY REQUIRE FURTHER CLARIFICATIONS. PLEASE USE THE AVAILABLE LAB TIMES TO ASK FOR CLARIFICATIONS AND ADVICE\HINTS ON THE TASKS BELOW.**

Part 1 – Using class *Class* **(10 points)**

Create an Eclipse Project called **Lab7Part1**. Create a class called **ReflectionTest** and place it in a package called **classreflections** (Note: When you put a class in a package the full class name is the package name first then the class name, e.g., "*mypackage.MyClass*"). Tests the following **reflection** related methods using the ReflectionTest class:

- **getClass()** – use this to get the class to print its own name
- Get the name of the class and print out using the **.class.** reference
- Use the **forName()** and **getDeclaredConstructor().newInstance()** methods to create an instance of the ReflectionTestClass class and then print its name using **getName()**

Required activities and marking guideline:

- Create the test class (1 point)
- Use getClass() reflection method to output name of class (3 points)
- Use the shorthand .class reference to output the name of the class (3 points)
- Create class instance using forName/getConstructor/newInstance (3 points)

Part 2 – Inner classes **(10 points)**

Create an Eclipse Project called **Lab7Part2**. Create a class called **GUIWithInnerClassHandler** that simply contains a button and a label (inside panels to make them look neat). Add the ActionListener to the button as normal EXCEPT DO THIS USING AN INNER CLASS CALLED **ButtonHandler** (Note: THIS MUST NOT BE IMPLEMENTED AS AN ANONYMOUS INNER CLASS IT MUST BE AN INNER CLASS WITH THE CORRECT NAME - ButtonHandler). Code the actionPerformed method in the ButtonHandler so that once the button is pushed it sets the text of the outer class label to say "Inner class has set the outer label" AND add the name of the inner class to the output label using the getName() method from the class Class (examine output class name in the label!!!...it should be interesting).

Required activities and marking guideline:

- Build GUI in outer class (3 points)
- Add listener to button using inner class (ButtonHandler) (3 points)
- Implement the inner class to change label in outer class (3 points)
- Output name of the inner class in the label using getName() (1 point)

Part 3 – Anonymous inner classes (10 points)

Create an Eclipse Project called **Lab7Part3**. Create a class called **GUIWithAnonymousInnerClass** that does exactly the same as Part 2 above EXCEPT USE AN ANONYMOUS INNER CLASS AS THE ACTIONLISTENER. As before once the button is pushed set the text of the outer class label to say "Inner class has set the outer label" AND add the name of the inner class to the output label using the getName() method from the class Class (again examine output class name in the label!!!...this should be even more interesting).

Required activities and marking guideline:

- Build GUI in outer class (3 points)
- Add listener to button using anonymous inner class (3 points)
- Implement anonymous inner class to change label in outer class (3 points)
- Output name of the anonymous inner class in the label (1 point)

Part 4 – Lambda Expression Button call (10 points)

Create a class called LambdaGUI that creates a JButton to output the name of the current class when the button is pushed. YOU MUST USE THE LAMBDA APPRAOCH TO CALL THE actionPerformed method.

- Build the GUI with the JButton (2 points)
- Add the action listener to the button (2 points)
- Call the getName() to output name of the class (2 points)
- Correctly code the Lambda expression for actionPerformed (4 points)

Part 5 – Investigating class features using reflection (20 points)

Create an Eclipse Project called **Lab7Part5**. Create a class called **InvestigatorClass** that will investigate the contents of a class called TheMysteryClass. You have been supplied with a jarfile called **TheMysteryClassJar.jar** that contains the compiled code of a mystery class called TheMysteryClass (note: you will not be supplied with the source code, only the Jarfile with the compiled code!). Import this jar file into your project using the library import Project->Properties->Libraries (tab)->Add External Jar (button) – point it to your local copy of the MysteryClassJar.jar file. Create an instance of the TheMysteryClass class in the Investigator class and output the following information using appropriate class reflection methods:

- The fields in the class and their types (there are TWO fields in the class)
- The names of all of the methods in the class (there are THREE methods)
- The return type of each of the three methods
- The parameter types of each of the methods (use toString() to print type of each of the parameters returned via the getParameters() method)

Finally attempt to call the last method in the TheMysteryClass method array using the **invoke(Object, args)** method (look it up in the oracle website or Javadoc).

Required activities and marking guideline:

- Get and display all field names in Mystery class (3 points)
- Get and display all field types in Mystery class (3 points)
- Get and display all method names in Mystery class (3 points)
- Get and display all method return types in Mystery class (3 points)
- Get and display all the parameter types of each method (4 points)
- Try to call any method from Mystery class using **invoke()** (4 points)