The University of Newcastle
SENG6110 Object Oriented Programming

Computer Lab – Week 5

This week we will start to use a new environment to edit, compile, run and debug our Java codes. It is called BlueJ. Today, you will learn how to edit, compile and run using BlueJ.

It will be very useful to use an IDE (such as BlueJ) for SENG6110. In the subsequent courses you should use a more sophisticated IDE (more information will be available in Canvas at the end of our course).

1. Watch the Lab-week5a-Using_BlueJ_with_Classes video available in the 'computer lab 5' in Canvas for the first introduction of BlueJ using an example similar to the one that you will be working on today.

2. Download the program Student.java and TestStudent.java from Canvas. Now follow the instructions below. a.

   Execute Bluej. Just write Bluej in the command line.

   b. Open a new project:

      i. Select New Project from project menu

      ii. Write StudentProject (this will create a folder called StudentProject and all Java files related to this project will be inside this folder)

   c. Add a class from a file

      i. Select 'Add class from file' from 'Edit' menu ii. Select the file Student.java iii. Add TestStudent.java

      as well

   d. To edit your code, just click twice and the code will open

   e. To compile, just press the compile button.

   f. To run the code

      i. Go to the Bluej window (go to TestStudent.java icon).

      ii. Use the right button of the mouse and choose main(). Don't worry about the other options now.

   g. Check the Student.java code and implement the method getHighScore (at the end of the file Student.java you will find more details). At this point, if you have difficulty in understanding how the classes/objects work, watch the Lab-week5b-Creating_and_Using_Objects_and_Debugging_Objects_with_BlueJ video before completing this exercise. This video will also show how to debug a code using BlueJ.

   h. Write a line of code that use the method getHighScore in TestStudent.java (use your imagination).

3. Bank example

   a. Using BlueJ, create a new project

      i. Select New Project from project menu ii. Write BankProject

   c. Add a class from a file

      i. Select 'Add class from file' from edit

      menu ii. Select the file BankAccount.java

      iii. Again, select add class from file from edit menu iv. Select the file BankAccountManager.java d.

   Compile/run the program in Bluej.

e. Be sure that you understand the code.

f. Notice that BankAccount.java has the instance variables name, idNumber and balance. The BankManager.java just manages the balance. Modify the program in such a way that it can work with the name and the id of the client. For example, before the menu, the program could ask for the name and the id of the client (it will be necessary to complete task f).

g. Write the methods getName and setName in BankAccount.java. Use them in BankAccountManager.java. g.

Document the program well

h. Watch Lab-week5c-More_about_Classes_and_Code_Documentation_using_BlueJ video to understand better the concept of encapsulation and how to generate documentation.

4. Write the UML class diagrams for the Bank and Student example from previous exercises.

Prof Regina Berretta -