

School of Electrical Engineering and Computer Science
The University of Newcastle
SENG1110/SENG6110 Object Oriented Programming

Lab Session 5 – Week 6

Attention: use only Bluej to edit, compile, run and **debug** your programs. In the beginning will be a little bit hard, but will be very useful in the future in SENG1110/6110 and subsequent courses. The aspect more important and more difficult is to know how to debug a program using BlueJ. You will start to see some of these aspects in this tutorial.

1. Download the codes Person.java, Couple.java and AgencyInterface.java from Blackboard. **Listen the first video**. Now follow the instructions below. Note that the instructions from a. to f. are the same from the previous tutorial. The instruction g. will show you how to debug a program.
 - a. **Execute Bluej**. Just write bluej in the command line.
 - b. **Open a new project:**
 - i. select open from project menu
 - ii. write agencyExample (it will be created a folder agencyExample and all programs related to this project will be inside the folder agencyExample)
 - c. **Add each class from a file**
 - i. Select add class from file from edit menu
 - ii. Select the file Person.java (repeat the process for Couple.java and AgencyInterface.java)
 - d. To **edit** your codes, just click twice in a class and the source code will open
 - e. To **compile**, just choose the compile menu.
 - f. To **run** the program
 - i. go to Bluej window (in AgencyInterface.java window – the class that has the method main).
 - ii. use the right button of the mouse and choose main(). Don't worry about the other options now.
 - g. To **debug** the program
 - i. Select show debug from view menu
 - ii. Go to AgencyInterface.java and click in the left of the line System.out.print("her name: ");. A symbol stop should appear. If does not appear is because you should go more to the left.
 - iii. Now run again as indicated in step f
 - iv. The program will run and stop in the line with the stop.
 - v. Now you can run each line clicking the **Step** or **Step Into** buttons and see the variables values in debug window. The difference between these two buttons is **Step Into** will enter inside a method and **Step** will not. Use **Step Into** and you will go inside each object.
 - vi. You can add more stops if necessary and you can delete a stop just clicking in the stop symbol.
 - vii. Note that during this process you can see the values inside the variables.
 - h. **(you need to complete this exercise and show, using BlueJ - debug, the result to your demonstrator)**

Add a new variable type Couple in the class AgencyInterface. Instead of one variable c of type Couple, have 2 variables Couple: c1 and c2. Modify the class AgencyInterface accordingly (add extra code to handle both Couple objects).
 - i. **Listen the second video**
 - j. Add a static variable in Person called qtdPerson and add an appropriate line of code to update this variable. For example, in the constructor.
 - k. Implement a method called equal in Person class to compare 2 Person objects. Use this method in the other classes to check if there is the same Person in different couples.
 - l. Implement an extra constructor in the classes Person and Couple
2. Candy example (**listen the third video**)
 - a. Using **BlueJ**, choose **new project**
 - i. select new Project from project menu
 - ii. write candy

- b. Add a classes Dispenser, CashRegister and CandyMachine
- c. Compile/run the program in the Bluej.
- d. Be sure that you understand the program.
- e. Declare a static variable named total in dispenser. Use it to show the total number of candies.
- f. Use this in some method in Dispenser.
- g. The machine does not give change. Implement it.

3. Consider 2 classes: **Subject** and **Student**.

- a. The **Subject** has the following attributes: **name** (String), **assign1**, **assign2** and **assign3** (representing the marks of each assignment) and **exam1** and **exam2** (representing the marks of each exam). It will have the following methods: **setAssign** (it will set the assignments marks); **setExams** (it will set the exams marks); **getAverage** (it will return the average of marks, suppose the assignments have weight 10 each one and the exams have weight 35 each one). Write in Java language the class **Subject**. The class needs to have a constructor that initializes the instance variables, the instances variables are private and the methods are public.
- b. Now write in Java language the **Student** class. The **Student** class has the following attributes: **name** (String), **subject1** (Subject object) and **subject2** (Subject object). The methods will be: **allMarks** - it will print all marks for a specific subject, the input will be the subject name (String) and it will return true or false (if this subject exist or not); **calcAverageAll** – the output will be the average mark of the student in all subjects.
- c. Now, write a java code to test both classes (use your imagination).

SENG6110:

4. Write the UML diagrams for Agency example.

Prof Regina Berretta
Apr-2017