159.272 Programming Paradigms

Tutorial 5: Java Warm-up Tutorial

No marks for this tutorial

Part 1: Command Line Java

Instructions

1. check the Java Development Kit is correctly installed on your computer
   a. open a console (mac/linux: terminal, windows: cmd)
   b. **run java -version** - must be at least version 1.7
   c. **run javac -version** - must be at least version 1.7
2. create a Java source code file from the following source code:

```
public class Calculator {
	public static void main (String[] args) {

		// parse first number from string to integer
		int number1 = Integer.parseInt(args[0]);

		// parse second number from string to integer
		int number2 = Integer.parseInt(args[2]);

		if (args[1].equals("+")) {
			System.out.println(number1 + number2);
		}
		else if (args[1].equals("-")) {
			System.out.println(number1 - number2);
		}
		else {
			System.out.println("don't know what to do");
		}
	}
}
```

remember: file names that contain the source code of a Java class must follow a certain convention, see lecture notes for details

3. compile this class with **javac**
4. execute the class with **java <classname>** , do not use any additional parameters. Note: if you get an error "Error: Could not find or load main class Calculator", try to run the command using this syntax: **java -cp . Calculator**. [1]
5. this will fail - now try to read the code, and run it with the additional parameters the program expects
6. in the program, replace line 4 with:

```
String number1 = Integer.parseInt(args[2]);
```

then try to recompile and to run the program. Read the error message, and try to explain the observed behaviour !

7. revert the last change, and replace the function name **main** in the program by **entry**. Then try to recompile and run the program. Read the error message, and try to explain the observed behaviour !
9. revert the last change, and add the following line as the first line to your program:

```
package myfirstpackage;
```

then try to recompile and run the program. Hint: have a look at https://docs.oracle.com/javase/specs/jls/se8/html/jls-7.html

10. revert the last change, and add the following line as the first line to your program:

```
package nz.ac.massey.myfirstpackage;
```

then try to recompile and run the program.
Hint: have a look at:
https://docs.oracle.com/javase/specs/jls/se8/html/jls-7.html

[1] The -cp parameter sets the classpath - the folder where Java looks for classes. In this case -cp . means - look for classes in the current folder (.).

Part 2: Create your first Eclipse project:

Instructions

Follow the instructions in the following page to create you first Eclipse Java project:
http://pages.cs.wisc.edu/~cs302/labs/EclipseTutorial/Step_02.html

1. Now create a class called **MyFirstJavaClass**. This class should have a ***main*** method.

   Note: your class should come under the source (**src**) folder
2. Declare the following variables:

   private static String name = "a name"

   private static String position = "staff"

   private static int age = 77

   All these variables should be declared as global variables (within the class but outside the main method!).
3. Print the following: your name is "a name", currently a "staff" and aged "77". Mind that the quoted words are to be feed-in from the variables declared above.
4. Run the program and see the output in the **Console** (at the bottom of Eclipse). To run, select the class that you would like to execute, **right click -> run**
5. Try to change ***Private -> Public*** and observe the effect. Also remove ***Static*** in (2) and observe the error!
6. Create a new method "**printDetails** " that takes 2 Strings and an integer and returns nothing. This should be **public.**
7. Move the print statement in (3) from the main method to **printDetails.**
8. Pass all declared variables in (2) to **printDetails** so it prints the details as given.
9. Call **printDetails** from the declared method. Remember: you need to create an object of this class in order to use this method.
10. Now create another class called **TestMyFirstClass** to observe how objects can be created from other classes. Follow the same steps in (1) to create the new class. This class should also have a main method.
11. Delete the main method in **MyFirstJavaClass.**
12. Move the variables that you have created and declared in (2) to the new class.
13. Within the main method of **TestMyFirstClass** , create an object from **MyFirstJavaClass** and then call the **printDetails** method.
14. Run the program and observe if there are any changes.
15. Try to change the modifier (declaration) of **printDetails** method from Public to Private and re-run the program. Observe the error.