



COMP 1030

Lab #4

Control Structures

Introduction

During this lab you will build two java classes. The first class will contain the required state and behaviour for the object **but NO main method**. The second class will contain simply the main method to give the JRE an entry point into the program, a line to instantiate a new object based upon the first class and a few lines to exercise the functionality of the first class.

We will also be working with an IDE (BlueJ) for the next few weeks. Additionally – you will also be capturing information from the user during this lab. To do so you will need to use the `java.util.Scanner` class. The Scanner object has behaviour that allows you to capture information from the keyboard such as `nextFloat()` – to capture floats, `nextInt()` – to capture ints, `nextLine()` – to capture Strings. Below is sample code for guidance.

```
import java.util.Scanner;
public class HelloWorld{

    public static void main(String []args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Deposit Amount?");
        float num1 = in.nextFloat();
        System.out.println("deposit amount was " + num1);
        in.close();
    }
}
```

When writing your code, keep these guidelines in mind:

- Start each class with the proper javadoc comment header. The first line of that comment should be the purpose of the class not just its name.
- Provide a comment for **each** line of code.
- Follow the layout for your class as illustrated below, (and in lecture two slides).

Javadoc comment header

Import statements (if required)

Class declaration

State (instance variables/data)

Constructor(s) (if required)

Behaviour(s) (method(s))

Close class declaration

- Use whitespace and indentations to make your code more readable and easy to debug.
 - Use notepad and the command prompt to do your work.
 - Be sure to clearly understand your work – do not simply copy code from someone else.
-

Challenge 1

Step 1 Download and install the BlueJ IDE from here: <https://www.bluej.org/> and launch the application.

Step 2 Click the Project tab and select new project.

Step 3 Call the project lab#4. Choose/create a new file folder under COMP1030 called lab#4

Step 4 Click the new Class button in the top left, provide an appropriate name.

Step 5 Double click the square icon to open the code editor.

Step 6 Delete the preloaded comments, then begin writing your own code following the template standards we have been using for the previous assignments. To run your code, right click on the class you wish to run and click on the second item in the list (void main ...).

Challenge 2

Step 1 Create a class called StudentRecord that is the blueprint a college student record (thus contains no main method).

- This class should have:
 - 4 appropriate fields, (state): firstName, lastName, studentID, gradeCourseOne
 - A constructor that takes no arguments.
 - A constructor that takes four arguments (firstName, lastName, studentID, gradeCourseOne)
 - A setter and getter for each field that simply sets and gets the field value – **IT DOES NOT** capture information from the keyboard, nor prints information.
 - A method which returns a letter grade for each course grade based upon the following criteria:
 - 100-90: A
 - 80-89: B
 - 70-79: C
 - 60-69: D
 - 50-59: E
 - <50: F

Challenge 3

Step 1 Create a second class called StudentRecordTestHarness which contains a main method to test the first class:

- **Note: Any interaction with the user must be accompanied with an appropriate message.**

Step 2 Instantiate a StudentRecord and pre-populate the four fields with random data using the constructor.

Step 3 Print out the data, including the letter grade, from this first record.

Step 4 Instantiate a second student record with no data.

Step 5 Ask the user for the data required to fill each field of the record.

Step 6 Print out the data, including the letter grade, from this second record.

- **Note: Any interaction with the user must be accompanied with an appropriate message.**

Challenge 4

Step 1 Add another method to the StudentRecord class which will determine the number of vowels in the student's first name based only on first names that are five letters long.

Step 2 Add a final test section to the test harness class that will ask the user for a first name that is 5 letters along with the other data and then will print out the state of the object, the appropriate letter grade along with the number of vowels in the first name.

Things to consider for success:

- Follow the layout for your class as illustrated above.
- **Write a line, compile a line**
- Write a section at a time and compile after each section so you do not have a volume of compiler errors to deal with.
- Comment as you go.
- Use indents and whitespace appropriately to make your code readable.
- Be sure that you actually understand the code you have written, simply copying someone else's code will not aid in your understanding of the java language.
- Stay focused and work diligently, collaborate with others if you are stuck.

