PROG2007 Assignment 1

Weight: 20% of your final mark

Due: 9th of April 2021 11:00 pm

Specifications

Your task is to complete various exercises in BlueJ, using the Java language, and to submit these via the MySCU link created for this purpose.

Marking criteria includes:

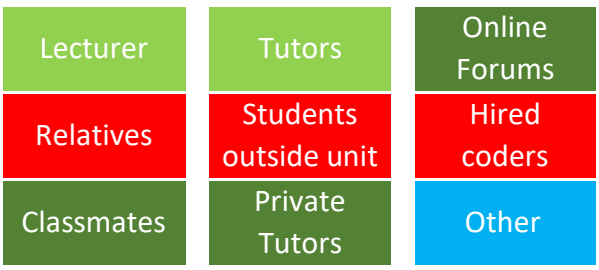
- Use of **correct coding style**, including the **use of comments**;
- **Accuracy** of coding;
- Use of **suitable coding structures**;
- **Correct submission** and **naming conventions** of assessment items as required.

Please note that all instructions in this assignment must be followed **EXACTLY**, including the names you are instructed to use. Failure to do so will **result in a loss of marks**. The reason for this is as a programmer, you will often work as part of a team and will be required to follow design documentation. If the design parameters are not followed precisely, bugs will be introduced into the software when all of the individual components of the program are assembled.

Getting Help

This assignment is to be completed individually. It is the opportunity to gain an understanding of the concepts of object-oriented programming and coding syntax. It is important that you master these concepts yourself. You are permitted to work from the examples in the study guide or textbook, but you must acknowledge assistance from other textbooks or classmates. In particular, you **must not** use online material or help from others, as this would prevent you from mastering these concepts.

Who can you get help from? Use this diagram to determine from whom you may seek help with your program.



Encouraged

Attribution Required

Ask tutor

Not acceptable

Assignment Q1

To be completed in week 1

Create a new **Microsoft Word** document and call your document as *YourFirstNameLastName-A1Q1.docx.*

Open the **house** project from chapter 1 of the reference book projects.

Open the terminal window and record method calls.

Create a picture with **at least six (6) objects** (circle, square, triangle, and person), recording all method calls.

Take a screenshot of your picture and add it to the word document you created earlier. Then **copy** the list of method calls needed to create the picture and **paste** them in the word document after the screenshot.

Add your **name and student ID in the footer** of the word document, **as well as "PROG2007 Assignment 1 Q1"**.

Assignment Q2

To be completed in week 2

Create a new BlueJ project called *YourFirstNameLastName-A1Q2*.

Create a class called **Employee**.

Make sure you write a description of your new Class in the comments, with your name as author and date as the last date you worked on this exercise.

Add definitions for the following fields:

- A **name** field of type **String**
- An **employeeId** field of type **int**
- A **wage** field of type **double**
- A **fullTime** field of type **boolean**

Write a constructor for your Employee class that takes f**our parameters** - the first of type **String** called **myName**, the second of type **int** called **myEmployeeId**, the third of type **double** called **myWage**, and the fourth of type **boolean** called **isFullTime**. Set the initial values of the corresponding fields using the constructor.

Write an **accessor** method called **getName** that returns the value of the name field.

Write a **mutator** method called **setEmployeeId** that takes a single parameter of type **int** and sets the value of the employeeId field.

Work out what other accessor and mutator methods would be useful for this Class and add them. You should be able to **get and set all fields** in the Class.

Write a method called **printDetails**, which prints out **all** the details of an Employee object. You must take into account the **fullTime** status and print a line saying either that the employee is fulltime or the employee is not fulltime.

For example, if:

- The name field holds the value "John Smith"
- The employeeId field holds the value 123456
- The wage field holds the value 25.40
- The fullTime field holds the value false

Then the **printDetails** method would print out the following:

The name of the employee is John Smith. The employee id is 123456. The wage of the employee is $25.40 per hour. The employee is fulltime.

If the fulltime field holds the value false, then the printDetails method would print out the following:

The name of the employee is John Smith. The employee id is 123456. The wage of the employee is $25.40 per hour. The employee is not fulltime.

Please Note: In the above examples, the name, employeeId, wage and whether the employee is fulltime or not (in blue) will change based on the values the fields hold. However, you must print the remainder of the statements exactly as in the above examples.

Assignment Q3

To be completed in week 3

Create a new BlueJ project called *YourFirstNameLastName-A1Q3.*

Create a class, **Assignment**, that contains the following four fields:

- A **String** called **StudentName**
- A **double** called **assignmentMark** (which will store the mark each assignment is worth e.g. for this assignment that you are doing right now the value would be 20)

- A **double** called **studentMark** (stores the mark the student gets in the assignment e.g. 15)
- A **String** called **grade**

Make sure you write a description of your new Class in the comments, with your name as author and give the version as the date you last worked on this exercise.

Define a constructor that takes and sets the **studentName**, **studentMark** and **assignmentMark**.

Also define a constructor that takes no parameters and sets the **assignmentMark** to **100**.

Create an **accessor** and **mutator** for **studentMark**. The mutator should not let the **studentMark** be set a value greater than the **assignmentMark** (as the student cannot get a mark higher than the assignment is worth) or less than 0. If the user tries to set a value that is not valid a suitable error message should be displayed.

Create a **method** that **calculates** the **grade** for the student. You will need to work out how many percent the student scored in the assignment.

If the student scored:

- Less than 50% the grade will be **fail**
- 50% - 64% the grade will be **pass**
- 65% – 74% the grade will be **credit**
- 75% – 84% the grade will be **distinction**
- Greater than 85% the grade will be **high distinction**

For example, if assignmentMark is 30 and studentMark is 15, the percentage will be 50% so the grade will be set to pass.

Define an **accessor** method to return the value of grade.

Assignment Q4

To be completed in week 4

Create a new BlueJ project called *YourFirstNameLastName-A1Q4.*

Create a class called **ListOfNames**, that has one **ArrayList** field called **names**, which holds a collection of **Strings** (each string is a male or female name in Upper case e.g. PETER).

Make sure you write a description of your new Class in the comments, with your name as author and give the version as the date you last worked on this exercise.

Define a **constructor** that initialises the ArrayList. Note that you can also add any other initialisations that you feel are relevant.

Create methods to **add** elements, **remove** elements and **get** the number of elements in the collection. Add a test for all three of these methods to check whether the operation was successful and print a message letting the user know if it was or was not.

Create a method called **printNames**. This method should loop through the collection and print out the elements (each String on a new line) as determined by the following rules:

- If the string contains any vowels (A, E, I, O and U), the method should print "The name " + the value of the String + " contains vowels, and the vowels are:" + list of the vowels in the string. The string may have more than one vowels. For example, if the name is EMMA it would print: **The name EMMA contains vowels, and the vowels are: E, A**
- If the string contains duplicate characters, the method should print "The name " + the value of the String + " has the following duplicate character(s):" + list of the duplicate characters in the string. The string may have one or more duplicate character. For example, if the name is ANNABELLA, it would print: **The name ANNABELLA has the following duplicate character(s): A, N, L**
- If the string contains any vowels (A, E, I, O and U), **and** it has duplicate characters, the method should print "The name " + the value of the String + " contains vowels and has duplicate characters". For example, if the name is LARISSA it would print: **The name LARISSA contains vowels and has duplicate characters.**
- If none of the above criteria is met, then the method should print the String element in lower case. For example, if the name is SKY, it would print: **sky**

Once you have finished your project, open the terminal window in BlueJ and turn on record method calls. Create a new **ListOfNames** object, and then add at least ten (10) Strings using the add method you wrote. You must have:

- A String that contains vowels
- A String that has duplicate characters
- A String that contains vowels and has duplicate characters

Demonstrate removing an element using the remove method you wrote, and then find the number of elements using the method you wrote that gets the number of elements. Finally, run your **printNames** method.

Copy all your calls into a **text file** called *YourFirstNameLastName-A1Q4-example.txt* and save it in your BlueJ project folder.

Assignment Q5

To be completed in week 5

Part A:

Imagine you need to write a program for a 24-hour clock with **hours**, **minutes** and **seconds**. Write a Java program in BlueJ with a method that prints all possible times the clock could display starting at 00:00:00 through to 23:59:59 when all the three numbers are the same (e.g. 01:01:01, 02:02:02, 13:13:13 and so on)

Part B:

Write a second method that takes three (3) parameters – **hours**, **minutes** and **seconds**. This method will print out all of the possible times the clock could display from one hour before the time passed to the method till one hour after the time passed to the method when all the three numbers are even (e.g. 12:20:00 or 12:20:02, NOT 12:20:01). For example:

If the method was passed the following values:

Hour = 11

Minutes = 23

Seconds = 44

The method would print all the times the clock could display from 10:23:44 until 12:23:44 when all the three numbers (hours, minutes and seconds) are even. The first printed time would be 10:24:00, and the last one would be 12:22:58)

Submission

You should now have 4 BlueJ projects, a word document, and a text file. You must zip the projects, the word document and the text file into one zip file called *YourFirstName_LastName_A1.zip*.

Submit this file via the Assignment 1 link on MySCU by the due date. Please leave enough time for it to upload, so do not submit it at the last minute!