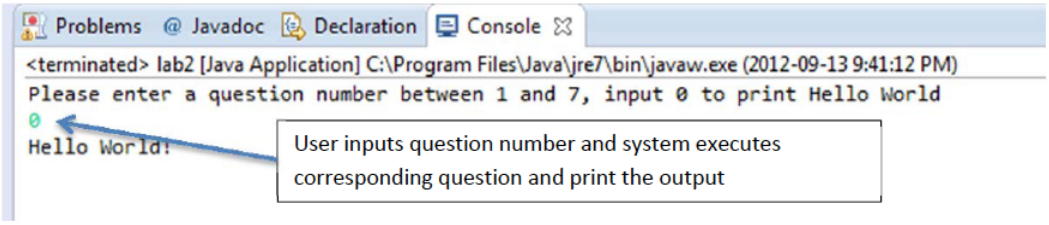
CSCI2010U – Laboratory #2
Java Programming

Introduction

This lab will assess your understanding of the Java programming concepts presented during lectures. It is assumed that you already have Eclipse installed on your laptop. To start you will need to create a project named “lab2”. Next, you need to create 2 Java classes named “lab2” and “questions” respectively and put the “main” function into “lab2” class. After the java files have been created you can copy the source code in “questions.java” on the course website into your “questions.java” file. There are 5 predefined functions in the “questions.java” file, each function corresponds to one lab activity.

Activity 0: Design a simple command line user interface (1 mark)

In this activity, you need to design a command line UI. User input a number between 1 and 5 inclusively; the system can execute the corresponding question. The following image shows an example of such UI.



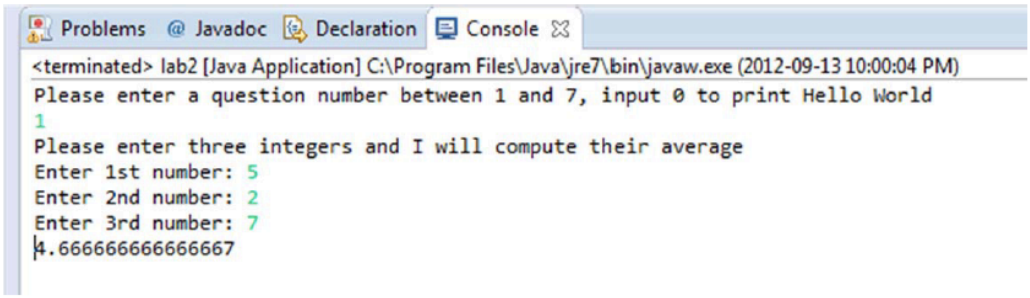
Activity 1: Computing Average (1 mark)

In this activity, user input 3 integers, and system prints the average. Part of the source codes is shown in the following image, Fill in the empty part without changing existing part. Put your code into “q1” function.

```
12  void q1()
13  {
14      int val1, val2, val3;
15      double average;
16      Scanner scan = new Scanner(System.in) ;
17
18      // get three values from user
19      System.out.println("Please enter three integers and " +
20                         "I will compute their average");
21
22      //Calculate average here
23
24      //Print it out here
25
26  }
27
```

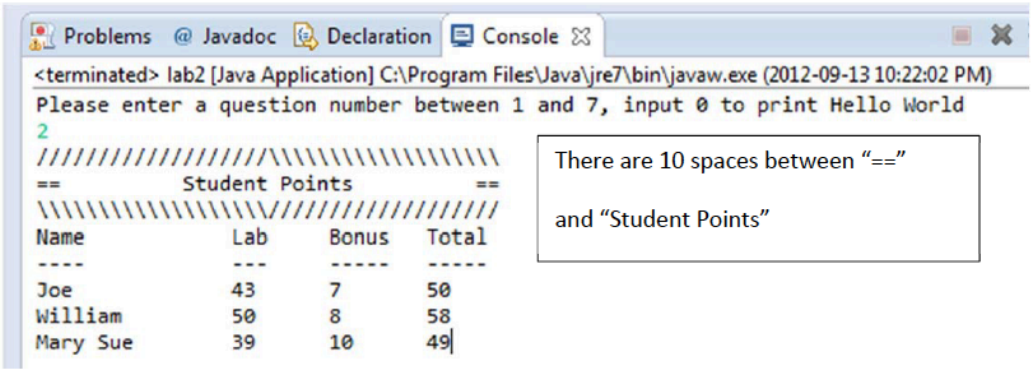
Finish the function here

Your system output should like the following image:



Activity 2: A Table of Student Grades (1 mark)

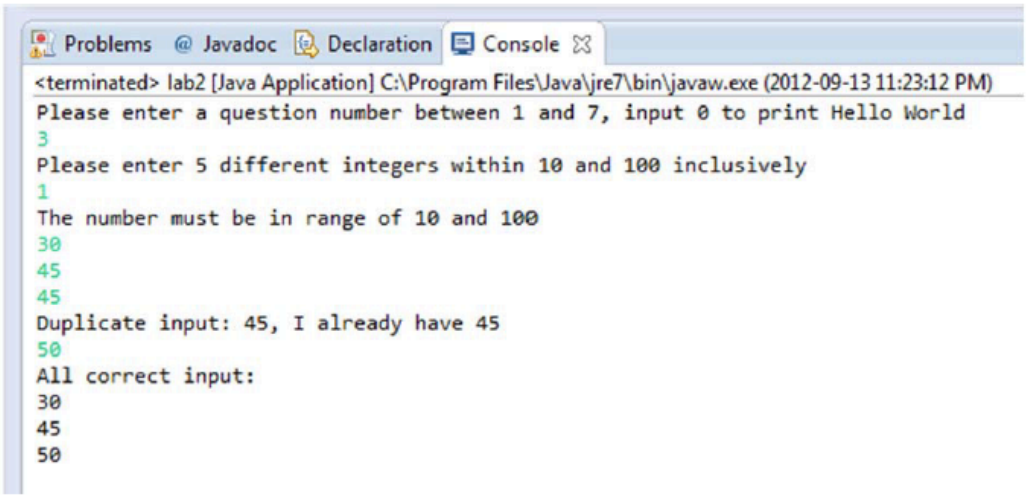
Fill in the code in “q2” and print out the following strings.



Activity 3: Duplicate Elimination (3 mark)

Write an application that accepts five numbers, each between 10 and 100, inclusive. As each number is read, if the number is a duplicate, print out a message about the duplication. Use the smallest possible array to solve this problem. Display the complete set of unique values input after user enters all 5 integers.

An example output is shown below.



Activity 4: Using String Objects (2 marks)

Fill in the blanks in the program below as follows: (Section 3.2, especially the example in Listing 3.1, should be helpful):

(a) declare the variable town as a reference to a String object and initialize it to "Anytown, USA".
(b) write an assignment statement that invokes the length method of the string class to find the length of the college String object and assigns the result to the stringLength variable
(c) complete the assignment statement so that change1 contains the same characters as college but all in upper case
(d) complete the assignment statement so that change2 is the same as change1 except all capital O's are replaced with the asterisk (*) character.
(e) complete the assignment statement so that change3 is the concatenation of college and town (use the concat method of the String class rather than the + operator)

You can find the incomplete code in “q4” function, fill in this part and finish the function.

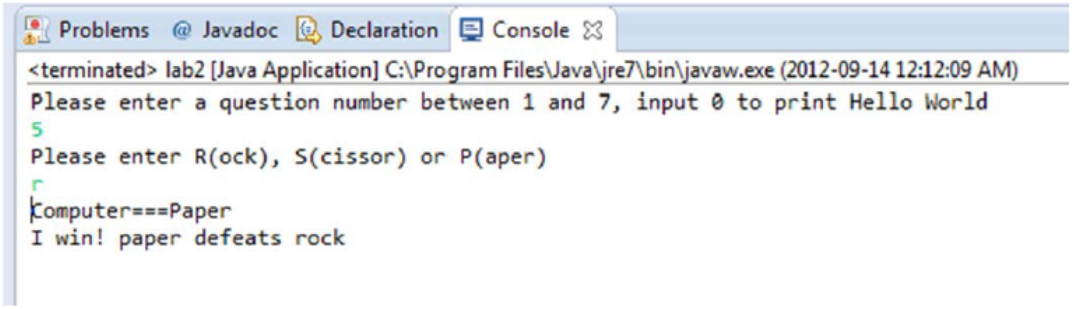
lab2.java | questions.java

```
06    void q4()
07    {
08        String college = new String ("PoDunk College");
09        // part (a)
10
11        int stringLength;
12        String change1, change2, change3;
13        // part (b)
14
15        System.out.println (college + " contains " + stringLength + " characters.");
16        change1 =       //part (c)
17        change2 =       // part (d)
18        change3 =       // part (e)
19        System.out.println ("The final string is " + change3);
20    }
21
```

Activity 5: Rock, Paper, Scissors (2 marks)

Function q5 contains a skeleton for the game Rock, Paper, Scissors. Add statements to the program as indicated by the comments so that the program asks the user to enter a play, generates a random play for the computer, compares them and announces the winner (and why).

Note that the user should be able to enter upper or lower case r, p, and s. The user's play is stored as a string to make it easy to convert whatever is entered to upper case. Use a switch statement to convert the randomly generated integer for the computer's play to a string. A possible output is displayed below.



What needs to be submitted?

Please submit your “lab2.java” and “questions.java” files on Blackboard. You do not need to submit your Eclipse project file.