Student Name: \_\_Ankan Basu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class and Section CSC101 M6\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Total Points (50 points) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project: Personal Information**

CSC 201 – Computer Science

New River Community College

Problem Description:

Write a program that reads in your first name, last name, student Id, and class that you are taking this semester, and displays the information one per line on the console and also on a dialog box.

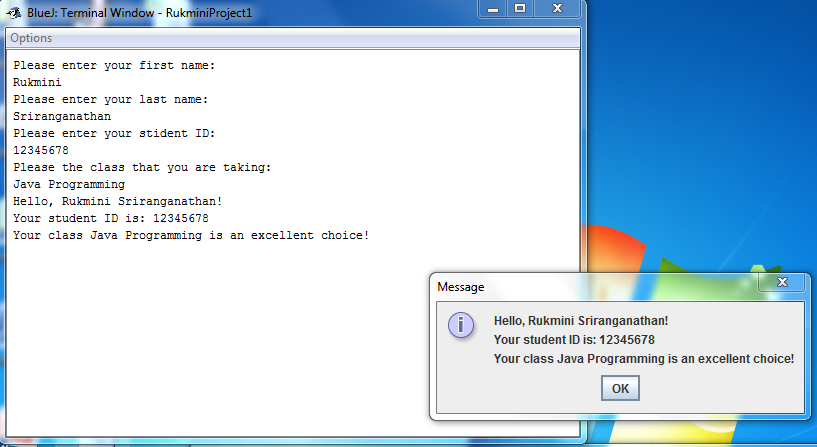
Use the following steps

1. Open BlueJ IDE (Double click on the shortcut on the desktop or go thro’ All programs)
2. Create project named JavaProject1
   1. Click on Project tab at the top and choose New Project…
   2. Choose a directory in where you want to save the project and type \JavaProject1
3. Create a class named MyInformation
   1. Click on the New Class… button on the left
   2. Type MyInformation in the text box below Class Name:
   3. Make sure the radio button to the left of class is selected. If it is not selected, please click on the radio button
   4. Click the Ok button
4. Double click on the class that you have created to open it on the editor.
5. Fill in the description, author and date in the comment box at the top.
6. Delete all the code before the Sample method inside the class as we did in class.
7. Modify the method header comment so that it would match your main method.
8. Type the header of the main method and add delete the content of the sample method between the two braces. (Refer to your book for the syntax of the main method header. Remember that java is case sensitive)
9. Inside the body of the main method, declare four variables (firstName, lastName, studentId, classTaking) of type String. Add comments for your declarations.

Example: String firstName; // variable to store the first name

1. Use System.out.ptintln statement which prompts the user to type his/her first name,
2. Use Scanner to read user input into the variable firstName. **Remember to import Scanner to your program.**
3. Repeat the process of prompting and reading for the lastName, studentId and classTaking.
4. Display the user information via the terminal window and a dialog box.

Here is the screen capture of a sample run (**Please change the inputs to reflect your own information**):



1. Make sure that your output is correct
2. Now, you need to create a jar file for submission. The following steps will help you to create a jar file:
   1. Click on the Project tab at the top and choose Create Jar File…
   2. In the Create Jar File dialog box, choose MyInformation for Main class
   3. Click on the two check boxes to include the source and BlueJ project files**. If your jar file does not contain the source code or BlueJ Project files, your assignment will not be graded.**
   4. Click on the Continue button
   5. In the File name text box, type yourNameProject1. For example, if your name is Susan Smith, you need to type SusanSmithProject1
   6. Click on the Create button to the right.

Analysis:

(Describe the purpose, processing, input and output in your own words.)

The project is designed to get inputs from the user and produce outputs in different format back to the user. By doing this project, the student is introduced to many concepts of java programming such as variable type, variable declaration, java API and utility etc.

**Design:**

**(Describe the major steps for solving the problem.)**

|  |
| --- |
| 1. Define variables and variable-types needed to store user input 2. Get user inputs using Scanner object 3. Print user input to the console window and to a dialog box |

**Testing: (Describe how you test this program)**

|  |
| --- |
| I have tested the program in multiple times by compiling at various stages of the program development. Main compilation stage was during printing to the console window. Once I was satisfied with the output, I have added addition code to print to the dialog box and make sure the fully developed was working as per directions in the project outline.  I have later found that although my version 1 of the program was running just fine, it still needed System.exit(0) command to end the JOptionPane window properly. Code modified and version 2 created with come addition cleaning up of code.  Version 3 of the program was made and in this version, comments during debugging stage were removed and indentations were placed properly. |

How to submit your assignment

* 1. Login Blackboard
  2. Click on Assignments on the left
  3. Click on Week 3 Work folder

Read the instruction there and submit the following items:

1. Your jar file with source code.
2. This document with answers for analysis, design and testing. This document is worth 10 points and the comments in your program is worth 10 points. Working code is worth 30 points.