COP 3330
Spring 2019
FT Assignment 2 – Final Project
Total point: 30. (20 points taken from assignments and 10 points taken from final term exam.)

Arithmetic Helper

Introduction: For this assignment, you will use Java, Eclipse, and WindowBuilder for Swing Application. In this assignment you will create a complete GUI application tool. This tool will help one to assess how fast they can complete an arithmetic operation of their choice. The top of your ALL source files must contain the following course header identifying you as an author:

```
/*  University of Central Florida
*  COP 3330 Spring 2019
*  Final project
* Author: <Your Name>
*/
```

Compliance with Rules: UCF Golden rules apply towards this assignment and submission. Programming assignment rules mentioned in syllabus, are also applied in this submission. The course instructor might randomly call students for oral exam for further assessment about the submission and authorship. You are allowed to search and learn various matters while doing this project. Copying few lines of code from internet is also fine to add any functionality. However, you should understand what have you copied. If it is found that you have copied the project from other students, you will directly get F in this course and you will be assessed for further disciplinary action from the University for your dishonesty.

Eclipse Implementation:

1. Create a Java project in Eclipse called "ArithmeticHelper".

2. You will need to use WindowBuilder for creating the JFrames for creating the GUIs. Please refer to the WindowBuilder installation manual in webcourses to set up the WindowBuilder in Eclipse.

3. Create the first JFrame for *main screen* as “ArithmeticHelperMain”. The main form will contain a number of swing components as shown in the picture. It will allow user to input various inputs.

4. For this assignment you are free to choose the number, and names of rest of the classes.

5. During this process you will need to navigate through various Jframes as shown in the images bellow. Please create necessary number of JFrame classes, add required components, and add necessary action performed methods for each components, depending on the problem requirement. Please read the problem description, and input/output format carefully.

6. Write your Java codes.
7. Save your Java codes.
8. Run your Java codes.
9. Submit the source files, and compressed JAR file in the webcourses.

The Problem:

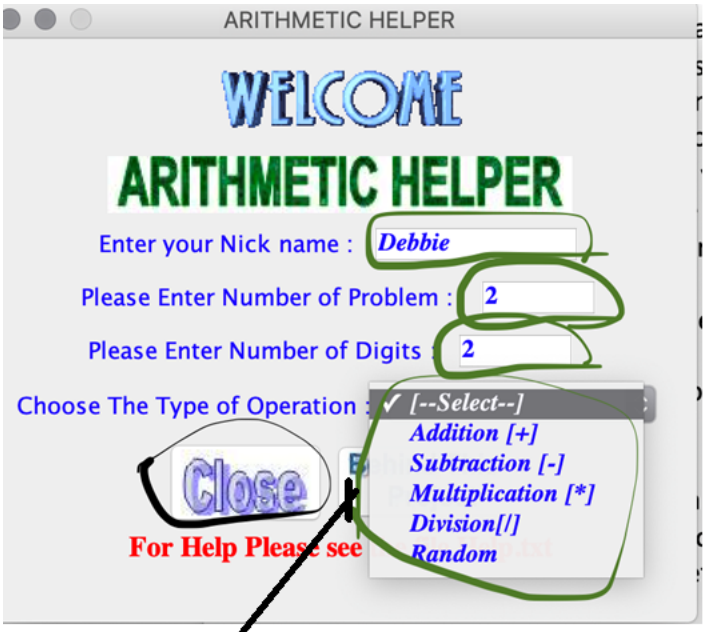
In this problem we will try to learn and implement graphical user interfaces (GUI) applications in JAVA using Swing objects. You will need to create a bunch of JFrames to provide several user interfaces for operations. At the first screen, you have to give your name, number of problems you want to solve, number of digits and operations you want to perform. The number of digits is the maximum number of digits of a number. There will be an option of random operation, this will generate any type of arithmetic operation randomly. After choosing an operation, problems will be appearing one after another in the screens (use the same JFrame for displaying all the problems and update the label texts only within that frame). In the problem screens, you have to input the answers. After completing all operations, you will able to see a new screen with how many problems you have solved correctly and time required to solve them. In this screen, you will also see your score, which is calculated based on your required time, number of digits, and percentages of corrected result. You will also be able to see the history of your problem-solving session and the rank list. From implementational point of view, the details of each screen are mentioned in the next section.

Sample Input/Output Format:

Here are the examples of some screens to guide you. You are free to choose the design and dimension of the screens. The examples are to make sure that you include the minimum required features in each screen.

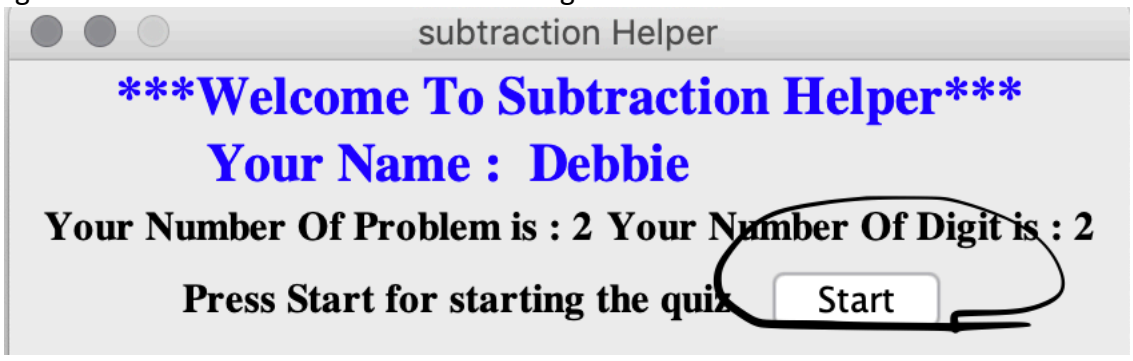
1. The first screen we call as *main screen*. In *main screen* you must provide the corresponding textboxes (marked green) to take input from user. The type of operation should include all four basic arithmetic operations (‘+’, ‘–‘, ‘*’ ,’ /’), with an extra option of random. In the screen below I choose Random. You must give a close button (marked red) to close the GUI altogether. When you click close button, it should show a popup thanking the user for using

arithmetic helper. You are free to choose any button or text field's styles. I gave an example name, the number of problems I want to solve, and the number of digits for each problem I want to solve. I gave both of them as 2. Let's see what happens in the next screen.



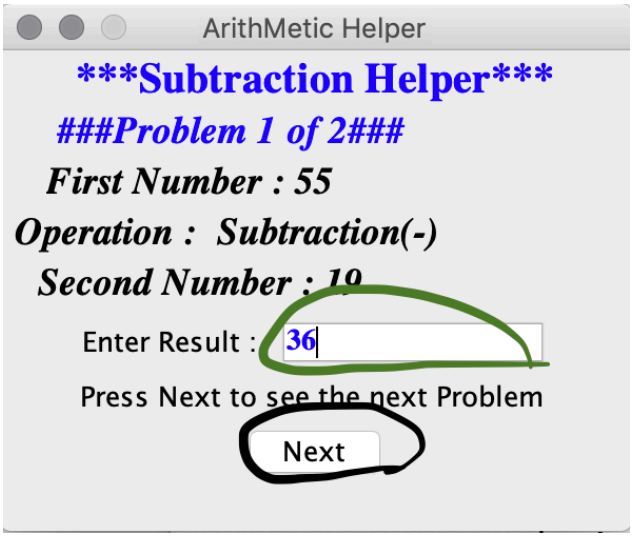
A button is not visible as I opned the drop down menu. The button is called behind this project. If you click here it should show a window that shows a brief intro about the developer

2. The second screen will show me the options and ask me if I am ready to start the quiz. Here you see, I have given Random as my operation option, it has chosen Subtraction as my operation. So, I will get all (which is two in my case) the subtractions operations with two digits numbers. I clicked the start button to go to next screen.

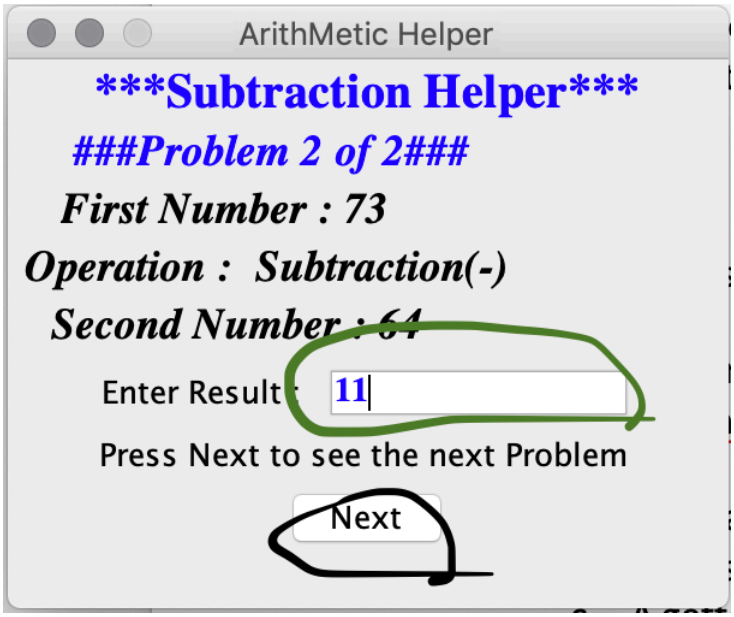


3. Now the arithmetic helper will show a bunch of screens depending upon the number of problems I entered in the main screen. I have entered 2. So, I will get two new problems in the screen one after another after clicking next button. However, for your testing please test

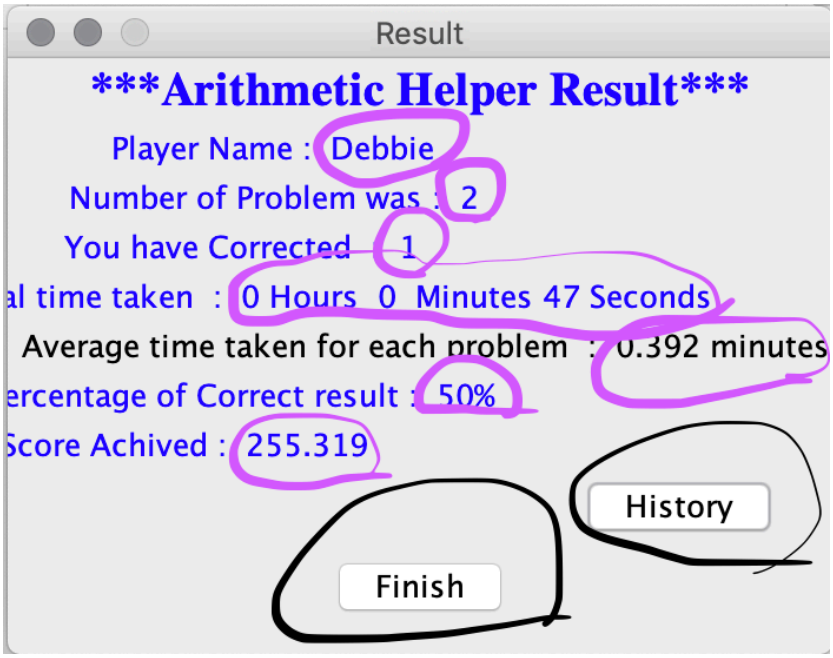
with at least 5 problems, and check if you are getting 5 problems or not. So, in this third screen, I will get my problem to solve, and give my answer in the text box, and will move forward to the next problem by clicking the "Next" button. Please note, the number will be randomly generated from your code and remember how many digits the numbers should include. For example, here the first number is 55 and second number of 19. These 55 and 19 should be generated randomly by your code within 10 – 99, as my chosen number of digits was 2. ***Also, note that you should use same JFrame for showing all your problems. You will just need to update the text in the labels for each problem.***



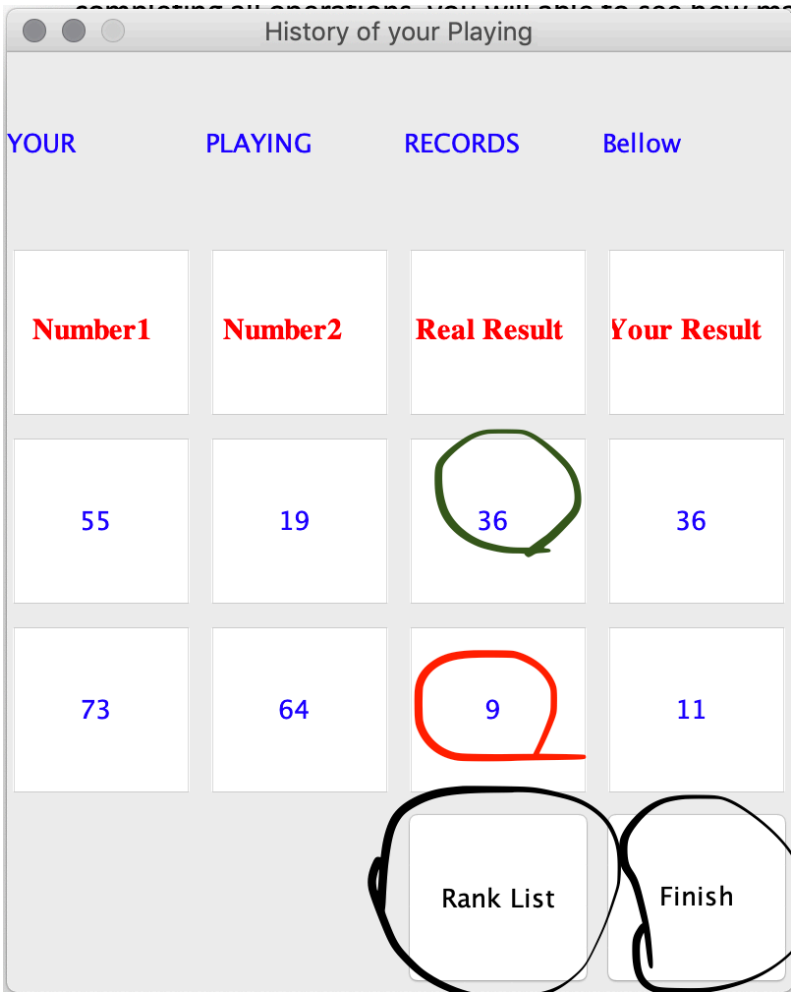
4. In the fourth screen, I will get the 2nd problem. I will solve that in similar way. However, I gave the wrong answer here. Let's see what happens next, and how do I score.



5. In the fifth screen, the arithmetic helper will give me the result of my quiz. Please follow the purple marked texts. The score should be determined based on the accuracy of my correct results, and average time taken to solve each problem. You can use your own formula, but faster completion and more correct results should always achieve better scores. The required buttons are marked in black. The finish button will finish the whole GUI application here. I clicked History to see my performance.



6. In the sixth screen, I get to see my answer record in the following form. Please remember you can choose any style and form you want to use, but you have to give at least the following components. You see, I have one correct and one wrong answer. From this screen I can either go to the next screen to look into my rank from history or finish the GUI application. I clicked "Rank List" button.



7. This is the last screen, where I will see my rank from the history of this application. For this you have to maintain a text file or binary file to record the history. In this JFrame, you have to read that text file/binary file and print the names of the players with top ten scores from the file. In this case, I scored only 255, so I could not make the top 10 rank list. Here you should give a button to go to the main screen.

8. For my case I saw total 7 screens, as I chose to solve 2 problems, if you choose to solve more problems, the number of screens appearing should be more. Please make sure that everything compiles and run.
9. Provide a "README.txt" file with instructions on how to run your designed ArithmeticHelper GUI application.
10. **Additional Requirements:** You must use a class for the 'User' in your program. Think about suitable instance and class variables and methods that you feel appropriate. The class should contain the full information, score, and time taken, and other relevant information for the user. During this process, you can consider passing objects from one form to another.
11. You can utilize currentTimeMillis()method for finding duration. Link: https://docs.oracle.com/javase/7/docs/api/java/lang/System.html#currentTimeMillis()

What to submit: Please submit the following in the webcourses.

1. A jar file of your developed tool. //Search how to create a jar file using eclipse
2. Put the full project and the jar file in compressed zip file and upload the zip file

Grading: Your submission will be graded using the grading rubric below. Please remember, late submissions will get 0 point directly.

Criteria	Points
Proper User class with good choice of attributes and methods and using it in your project	3
Correctly show the first screen, or main window with all required fields.	3
Correctly displaying behind the project window	2
Correctly showing the second screen with the selected operation, number of problems, and start button to start the quiz.	3
Showing new screens with random problems of my chosen arithmetic operation (or random operation if I choose Random as my arithmetic operation)	8
A new screen, showing the result of my taken quiz with all the corresponding information, such as score, accuracy, total time taken etc.	3
A new screen, showing the history of my taken quiz with the problem's correct result and my result.	4
A screen showing the rank list of the application.	4

Happy Coding!!!