Sukkur IBA University
Merit - Quality - Excellence

CSC-150 – Object Oriented Programming
Semester II (Spring 2020)
Course Instructor(s): Dr. Sher Muhammad, Muhammad Atif, Khalid Hussain

Lab 12: Graphics With Swing

1. Design and code a Swing GUI to translate text that is input in English into Pig Latin. You can assume that the sentence contains no punctuation. The rules for Pig Latin are as follows:

    a. For words that begin with consonants, move the leading consonant to the end of the word and add “ay.” Thus, “ball” becomes “allbay”; “button” becomes “uttonbay”; and so forth.

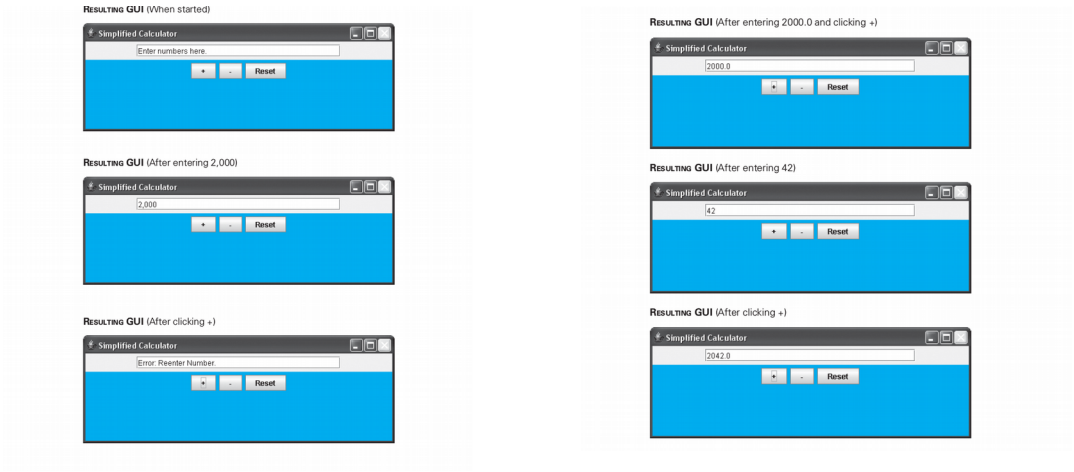
    b. For words that begin with vowels, add “way” to the end of the word. Thus, “all” becomes “allway”; “one” becomes “oneway”; and so forth.

2. Design and code a Swing GUI for a two-player tic-tac-toe (noughts and crosses) game on a 3 * 3 game board. The JFrame should use a BorderLayout with a JLabel in the NORTH region to display messages (e.g., who won the game), and a JPanel in the CENTER region to display the game board. For the game board in the JPanel , use a GridLayout manager with a 3 * 3 layout of JButton ’s in each cell to display the game board. The button labels should initially be blank. When a player clicks on an empty button, an appropriate “X” or “O” should be placed in the label field of the button. If there is a winner (three in a row), then the program should display the winner in the JLabel located at the top of the window. If all nine cells have been filled without a winner, the program should indicate that there is a tie.

3. Design and code a Swing GUI calculator. You can use Picture below as a starting point, but your calculator will be more sophisticated. Your calculator will have two text fields that the user cannot change: One labeled "Result" will contain the result of performing the operation, and the other labeled "Operand" will be for the user to enter a number to be added, subtracted, and so forth from the result. The user enters the number for the "Operand" text field by clicking buttons labeled with the digits 0 through 9 and a decimal point, just as in a real calculator. Allow the operations of addition, subtraction, multiplication, and division. Use a GridLayout manager to produce a button pad that looks similar to the keyboard on a real calculator.

When the user clicks a button for an operation, the following occurs: the operation is performed, the "Result" text field is updated, and the "Operand" text field is cleared. Include a button labeled "Reset" that resets the "Result" to 0.0 . Also include a button labeled "Clear" that resets the "Operand" text field so it is blank.

Hint: Define an exception class named DivisonByZeroException . Have your code throw and catch a DivisonByZeroException if the user attempts to “divide by zero.” Your code will catch the DivisonByZeroException and output a suitable message to the "Operand" text field. The user may then enter a new substitute number in the "Operand" text field. Because values of type double are, in effect, approximate values, it makes no sense to test for equality with 0.0 . Consider an operand to be “equal to zero” if it is in the range -1.0e-10 to +1.0e-10 .



4. (The Swing part of this project is pretty easy, but to do this programming project you need to know how to convert numbers from one base to another.) Write a program that converts integers from base ten (ordinary decimal) notation to base two notation. Use Swing to perform input and output via a window interface. The user enters a base ten numeral in one text field and clicks a button with "Convert" written on it; the equivalent base two numeral then appears in another text field.
Be sure to label the two text fields. Include a "Clear" button that clears both text fields when clicked. (Hint: Include a private method that converts the string for a base ten numeral to the string for the equivalent base two numeral.)

5. Here we will a game known as Trivia (Kon banayga Crorepati) use a layout of your choice with the appropriate text fields, labels, and buttons to implement your design. The game should ask only one question at a time with the time limit of 60 seconds to answer and output the correct answer if the player answers a question incorrectly or time limit exceeds (if time limit ends answer will be considered wrong). When all questions have been answered, show the final score and exit the program.