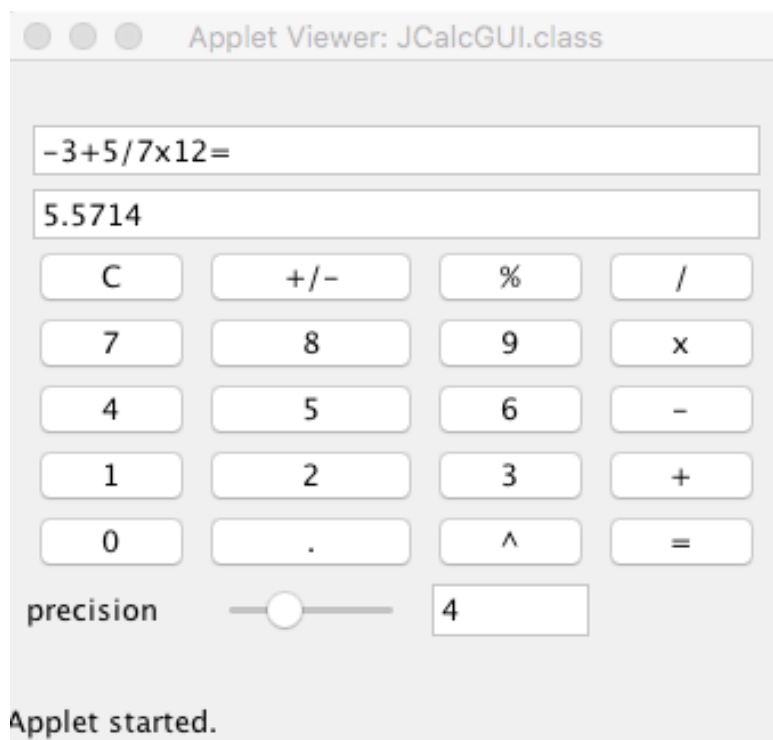


Department of Electrical and Computer Engineering
ECSE 202 – Introduction to Software Development
Assignment 6 – A Basic 4-Function Calculator

due November 28, 2017 at 5:00 pm.

Problem Description

By now you should have developed working code that converts an infix expression to postfix, evaluates the result, and returns a numeric answer. The goal of this assignment is to design a graphical user interface (GUI) that allows the user to key in an expression and have the result appear on a display, something along the lines shown below:



The design is completely up to you, but must satisfy the following minimum requirements:

1. Add, subtract, multiply and divide must be supported.
2. You need to have a CLEAR button (C in the picture). When pressed, the current expression and result are erased and a new expression may be typed in.
3. Pressing the EQUALS key causes the expression to be evaluated and the result displayed.

Feel free to add any functionality as you see fit. You can use any of the `acm` classes in your implementation. Note, you are not being asked to design a calculator in the normal sense, i.e., where the expression is evaluated one operator at a time (this would be straightforward and not require infix to postfix conversion). The goal here is to provide something a little more sophisticated, without a major redesign of your code.

Method

An easy way to tackle this assignment is to design a user interface that produces an expression that is compatible with your existing framework. Suppose a user entered the following expression using the calculator keyboard: $-33+54.9/7\times 3.4$. Parsing this expression is simplified as operands are delineated by operators: -33 by +, 54.9 by /, 7 by x, and 3.4 by the eventual press of the = key. Negative numbers are a special case; the +/- key is not treated as a delimiter but as a digit. The GUI enqueues each token as it is typed in, replacing the command line interface used in the earlier assignment. The EQUALS key is not processed as a token, but is used to initiate conversion of the expression and its evaluation with the result being displayed on the second line of the display as shown.

Notes:

1. All arguments are interpreted as doubles and the result displayed to a default precision of 6 decimal places.
2. Handling negative numbers counts for a bonus of 10 points (this was a requirement, but given the timeframe of this assignment it is now a bonus).
3. Adding a menu item that controls the number of decimal places is an additional bonus of 10 points
4. The design of the GUI is flexible, but should have the minimum functionality depicted in the figure on the previous page.

Hints:

Design of the GUI is greatly simplified by using the TableLayout manager described in the slides for Chapter 10. This allows you to easily populate a grid with JButtons and JTextFields for creating the display. You can figure out most of this by looking at the examples in the slides. Note that the setText(String text) method allows you to change the contents of any text field dynamically. This gives you most of what you need to implement the GUI.

Max grade on this assignment 120/100

To Submit:

1. Your source code files (contents of src folder)
2. A file containing a snapshot of several program runs.

As before, make sure that each class file uses the default package.

fpf/November 14, 2017