David Jefferson and Sean Harrison

ICS 4UI GUI Report Journal

Thursday, March 20th:

I have completed steps one through five of my initial program specs with no issues. My current version (“Version5.java”) displays the three text fields along with their labels, and the combo box and “perform” button in the main frame. These components are currently displayed in the main frame using the “FlowLayout” layout manager, but I am considering using the “GridLayout” layout manager. Now that I see the way in which “FlowLayout” displays the components of my of the GUI, I think a grid layout would be a better way of organizing them.

Friday, March 21st:

Today I had my meeting with you, and the program specs have changed. I am now creating a calculator with the standard calculator input, rather than the text field and combo box input I was working on previously. For this, in order to format the components of the GUI in the standard layout, I will need to use the “GridLayout” layout manager as I suspected yesterday. I will shift my focus to the calculator program using the features I have included in previous steps.

Tuesday, March 25th:

I have learned about the “GridBagLayout” layout manager, and have created a GUI that resembles that of a standard calculator in my “Version6” program. I will pretty-up the button layout once I make the program functional.

Thursday, March 26th:

I have created an action listener that can be used for buttons 1 through 9 in “Version7.” “Version7” finds which button was pressed from one action listener and outputs that number to the screen.

Friday, March 28th:

I am having a hard time wrapping my head around a structure for the code needed for a calculator that performs many operations, such as 3 + 5 – 6 \* 4, so I have decided two things; Firstly, I will not create a calculator that performs BEDMAS, but will only perform calculations in the order in which they were inputted, and secondly, I will create “Version8” to only perform single operations, such as 4 + 7 or 8 \* 3. I intend to add in multi-operational functionality to the calculator in later versions, but for now I will keep it simple.

After running “Version8” and pressing a numerical button, I am faced with a number of errors in the console output. It points out one line of my code (dWorkingNum1 = Double.parseDouble(sWorkingNum1); , which is line 175) and over 30 others from Java’s pre-existing functions. The error message was: Exception in thread "AWT-EventQueue-0" java.lang.NumberFormatException: For input string: "null4". However, when I comment out the problem line, the program does everything else it is supposed to do, the only consequence being that the variables that change based on user input do not update, and therefore always result the “0.0” that they were defined with. I will fix this in my next version, “Version9,” so you can look at this error in “Version8.”

Saturday, March 29th:

I read from <http://stackoverflow.com/questions/4646577/global-variables-in-java> that, in order to use global variables, I had to call them as I would call a function. I then changed my code so that every instance of a global variable was being called from the “Version9” class that they were defined in. The good news is that I no longer get errors when I press a numeric button on my calculator. The bad news, however, is that when I press the “equals” button, zero is output regardless, telling me that the variables are not changing from what they were defined to be. This could be due to a number of problems, most likely my programming faults. I will continue to attempt to fix my program in the “Version9” version. The first method of changing from string (which is retrieved using the getActionCommand) to double was the Double.parseDouble function, but that was unsuccessful. I have also tried other ways of converting string to doubles, such as:

Version9.dWorkingNum1 = (Double.valueOf(Version9.sWorkingNum1)).doubleValue();

which is a modified line from <http://docs.oracle.com/javase/tutorial/java/data/converting.html> . However, this is yields the same result as Double.parseDouble. I am very confused at this point in time, as my (very limited) knowledge of Java leads me to believe that I have a problem with either global variables or my parsing methods, yet after some troubleshooting, I have had no success.

Sunday, March 30th:

I have decided to go take a step backward and continue from the last working version, “Version7.” The latest version, “Version10,” is based off of “Version7.” In this version, I have decided to improve the GUI of my program, and take a break from debugging the problems I had yesterday. I have just noticed that my program lacks a zero button, so I will add it. Also, I have been aware for a while that some of my buttons, such as the “+” and “-“ buttons were not aligned very well, so I will attempt to fix this problem. Also in this version, I will modify the program so that the button number will be displayed in the text field in the GUI instead of the console. I will do so just to make sure that I can display to the text field without calculating anything.

After checking my program again, I noticed that I set some of the GridBagConstraints to be Vertical, as opposed to Horizontal. I originally set it as Vertical in my experimentation with GridBagLayout, and I must have forgotten to change it back to Horizontal. This was the root of the misalignment problem, which has now been solved. I also added the “zero” button, and cleaned up the look of the GUI.

For the sake of helping myself along, I tried running “Version8” again, and to my confusion, it did not give me the 30 odd errors it did when I pressed a numeric button yesterday. Instead, it performed in the exact same way as “Version9,” it always output zero as the result. I can therefore conclude that calling the variables from the class they were defined in (e.g. Version9.dWorkingNum1) did not help me whatsoever, and is unnecessary. Overhauling the action listeners, like I have already begun in “Version10,” is looking to me like my only option at the moment.

In “Version11,” I have created a separate class for the CE button, named “Ce,” which is just so that there is one less ‘if’ in the “Calc” class (in “Version8” and “Version9,” “Calc” class includes the Clear Everything option). I justify the creation of this separate class because my code is less cluttered with it; it is easier to decipher that this is what the CE button does. I am considering creating a separate class for the “equals” button so that I do not have too much nesting, as I had in “Version8” and “Version9.”

“Version12” has had a success: in it I have got changing variables for the first time, and I have calculations work... sometimes. If I perform calculations using the same operation (such as 2 + 2 , 3 \* 4 \* 6 , or 9 – 5 – 3), The answer is what it should be. However, when I perform calculations with different operations (such as 1 + 3 \* 6), the answer is incorrect (Note: This version does not, and was not intended to, use BEDMAS). I believe I know why this is; the operation that the user specifies changes to another operation before the first one is calculated. This happened because I needed to retrieve another number from the user before I could calculate anything, but as it turns out, I changed sOperation (the variable holding “+” , “-“ , “\*” , or “/”) without realizing it. The next version will attempt to solve this problem by changing sOperation after performing the previous operation.

After doing as I planned above, I have created a program that I believe to complete (unless other people can crash it in creative and disappointing ways, which I can almost guarantee will happen). This is named “FinalVersion.”