**Connor Welham**

**GUI Calculator**

AS91355

AS91372

AS91373

**Calculator Fundamentals**

**Introduction**

To support the teachers of junior learners, your task in this assessment activity requires you to undertake technological practice to develop a Basic four function GUI calculator.

Stakeholder/client: Daria Spence (Mom)

My stakeholder is my mom who has recently been going to St Thomas as part of her teaching course. She teaches a range of students from Year 0 – Year 6. For her primary school students, they could use a four function calculator for basic mathmatical equations.

On first consultation, she wanted the colours of the calculator to be bright. She wanted the calculator to be easy to use and the buttons to be large. This is because it is easiest for the young students.

I plan to work with the stakeholder to create a four function calculator that is visually appealing for the user. It will be fully functional and will account for any errors.

**Research**

<http://whatis.techtarget.com/definition/calculator>

A calculator is a device that performs arithmetic operations on numbers. The simplest calculators can do only addition, subtraction, multiplication, and division. More sophisticated calculators can handle exponent e.g. operations, roots, logarithm s, trigonometric functions, and hyperbolic functions. Internally, some calculators actually perform all of these functions by repeated processes of addition.

Most calculators these days require electricity to operate. Portable, battery-powered calculators are popular with engineers and engineering students. Before 1970, a more primitive form of calculator, the slide rule, was commonly used. It consisted of a slat of wood, called the slide that could be moved in and out of a reinforced pair of slats. Both the slide and the outer pair of slats had calibrated numerical scales. A movable, transparent sleeve called the cursor was used to align numerals on the scales. The slide rule did not require any source of power, but its precision was limited, and it was necessary to climb a learning curve to become proficient with it.

**AC – All Clear/Turn on**– This will wipe any previous equations on the calculator and take you back to 0. This can also be CE.

**C – Clear**– The C button erases the last number or operation entered, use it if the last number you entered was a mistake.

**DEL & INS Delete and Insert-** Delete will delete individual characters, so you can alter complex equations without clearing the whole formula. Insert allows you to re-enter numbers over a formula, like on a PC keyboard.

**+/- Plus/Negative** – This changes the number on screen to a plus or negative

**%- Percentage** of a number or percentage in an equation

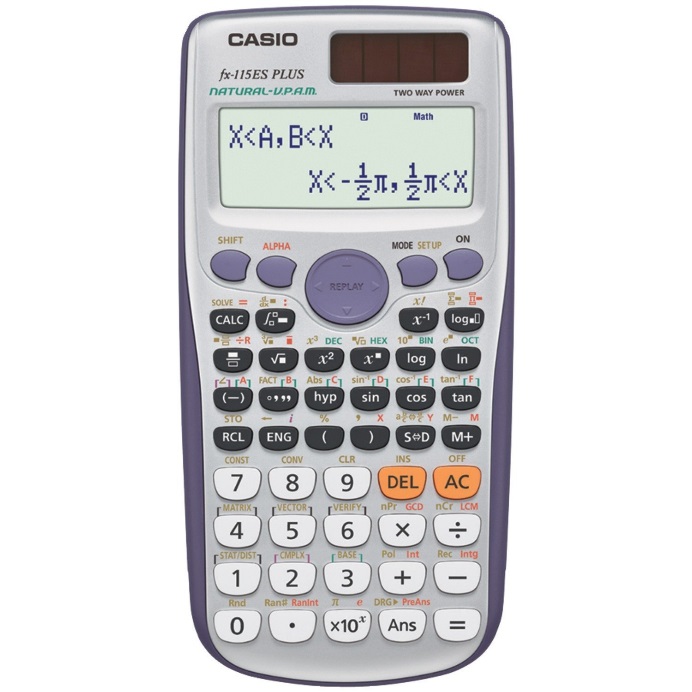
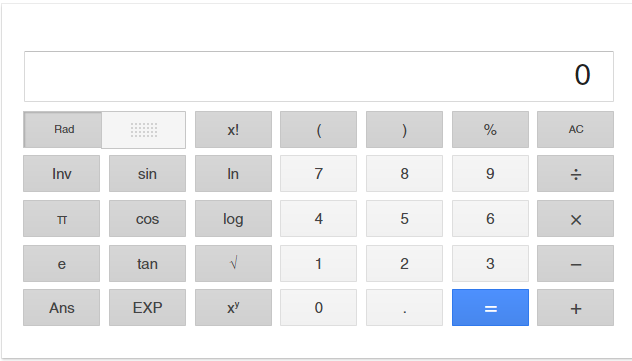
**Π or pi-** 3.14159 (continued depending how flash your calculator is)

**√- Square root.** This button will show the square route of a displayed number.

**X2** – **Square-** The x2 button computes the square of the number currently displayed. (e.g. 5×5)

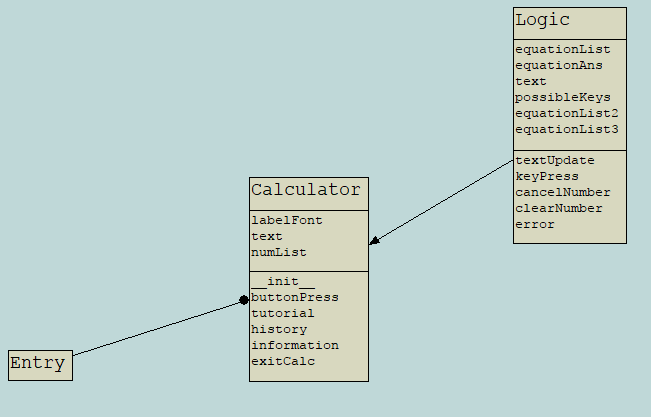
**X3-****Cube**. Computes the cube of the displayed number (e.g. 5x5x5)

**Sine**, **Cosine** and **Tangent** are all used in trigonometry. They are used to establish a number from an angle, in a right angled triangle. To work out a SIN/COS/TAN you need to know the hypotenuse, opposite and adjacent.



**Research of Other Calculators**

**PynSource: UML Diagram**



**Critical Reviews**

Critical Review 1

First critical review of the entire project. My stakeholder looked over all my work, so my research, planning, first drawings and my first ideas. She said that my research was good, but also added to look up other operations. Even though it is just a four function calculator, if I research now I could expand it in the future for older students. It could be used as a cheap alternative to a graphix calculator. In terms of my planning, again she asked to expand it in terms of oeprations. Even just to put question marks after them to show that it could be used for a future expansion of the model. Out of my two drawings, we worked together to come up with one final drawing that my calculator would look like. We had a roough idea of what colours I would use, but we were not 100% sure.

Critical Review 2

Critical review 2 took place after I had done my algorithms and coded my buttons. My client offered ideas on how to improve my algorithms and how to plan this as they looked over how I created each button individually. They suggested that there could be a better alternative. In addition, they said that text in a calculator must start from the right hand side as this is the case with all calculators. This means it will be easier to use for people who have used other calculators. My client saw my code and understood that I was just working on its functionality and not the aesthic appeal. We agreed that I would work on that when my calculator fully worked.

Critical Review 3

Daria liked that I had two buttons for deleting the text. Although it may be complex at first for the younger students to use, it is a necessary function. She asked me to change the writing for the cancel button, using a symbol instead of writing. She said that the students will recognise the symbols easier than writing. The client said that the algorithms were all good, but maybe to make them more central on the page because it looked a bit messy.

Critical Review 4

After explained the classes to my client, she udnerstood why the calculator wasn’t working currently. Instead, I showed Daria a previous version that worked without classes. She was happy that I fixed all the errors, especially the leading zero error. The key binding was also a big step because it would make the calculator much easier to use. She suggested that enter and the equals keys both allowed the calculator to calculate. Also that the delete key clears all and the backspace key just cancels one number.

Critical Review 5

Final critical review of the entire project. Client was happy with everything, just a few touch ups with the deskcheck to do to make it easier to read. Commented that the code looked nice as they tested the final version for error and the usability. Looked at different shades of a few colours, but it was better as it was.

Task 1: Analyse the problem

To support the teachers of junior learners, your task in this assessment activity requires you to undertake technological practice to develop a Basic four function GUI calculator.

Task 2: Identify the input information

*What information will the user have to enter? Copy and complete this table.*

|  |  |  |
| --- | --- | --- |
| **Variable name** | **Format** | **How entered** |
| number | String (global variable) | Button click, key press |

Task 3: Identify the output information

*What information will the program need to print out? Copy and complete this table.*

|  |  |  |
| --- | --- | --- |
| **Output** | **Format** | **When it happens** |
| equationAns | String | When equals button pressed |
| number | String | When button clicked, key pressed |

Task 4: Identify the stored information

*What information will the program need to store? Identify any constant values, if necessary.*

|  |  |  |
| --- | --- | --- |
| **Variable** | **Format** | **Equals** |
| text | Entry | Display box |
| status | Label | Status bar |
| numCol | Integer | 0 |
| numRow | Integer | 1 |
| numList | List | All buttons |
| operatorButtons | List | All operators |
| possibleKeys | List | All key binding names |

Task 5: What calculations are necessary

*Write out the calculations in terms of the variables specified above.*

Evaluate equationList variable.

When button created, add one to column.

If column equals 4, add one to row and reset column to 0.

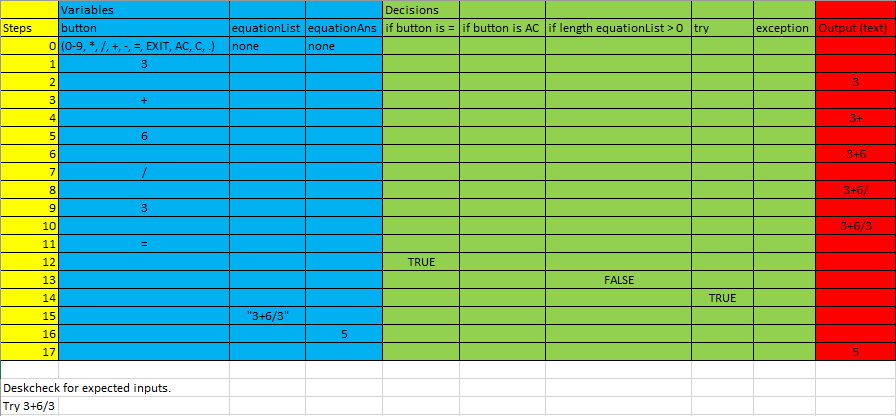
Task 6: Test Data table of expected inputs

*Create a table of expected input data using Excel or Word.*

|  |  |
| --- | --- |
| **Input** | **Output** |
| Press button (0-9, operators) | Number/operator added to entry widget |
| Press equals button | Outputs answer to equation in entry widget |
| Press delete button | Clear everything in entry widget |
| Press cancel button | Last thing added to entry widget clears |
| Press exit button | Calculator GUI closes |

Task 7: Deskcheck of steps used in processing

*Create a deskcheck showing how the output will be obtained. This could be with printed table and pen or in Word or by formula printout in Excel; and add the result to your document.*



Task 8: Create a testing procedure

*Describe the steps that the computer program will have to take to solve the problem. You may choose to do this as pseudocode or as a flowchart. You must clearly identify sequence (steps in order), selection (choice of actions) and iteration (looping or repetition).*

*You are recommended to develop several versions of your plan showing how you will continually improve and refine your program to solve the given task. The final version in your testing procedure should clearly show how the program will solve the given task on expected data inputs.*

*Make any additions to the input, output and storage tables that are necessary as you refine your plan to solve the problem.*

Algorithms (See Attached)

*All algorithms have been developed over the course of the project as I have made changes to my code. This is the final algorithm.*

ButtonPress:

Algorithm that shows what happens when you press a key or if you click a button on the calculator. This will redirect you to another definition depending on what button/key has been pressed. If the button/key is 0-9 or an operator, this algorithm shows it will be inserted into the entry widget.

TextUpdate:

Algorithm for when the equals key is pressed. Algorithm shows the leading zero error as well as any other errors that may occur in the calculator. It calculates the answer and outputs it.

DeleteNumber:

Algorithm for when the delete button/key has been pressed. This will remove the last part of the entry widget. Used for when the user has made a mistake in the code and wants to redo. Can also be accessed from the taskbar under “edit”.

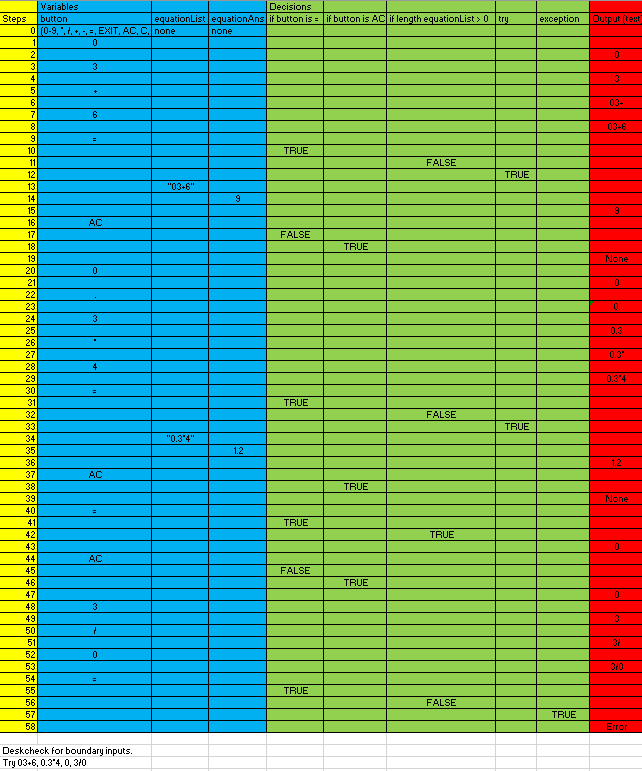
ClearNumber:

Similar to the deleteNumber, this algorithm clears the entry widget when the user wants to input a new calculation. Like the deleteNumber, it can also be accessed from the taskbar under “edit”.

Task 9: Create a testing table for boundary inputs for testing the program

*Add to your test data table to indicate what boundary inputs you would include for testing the program*

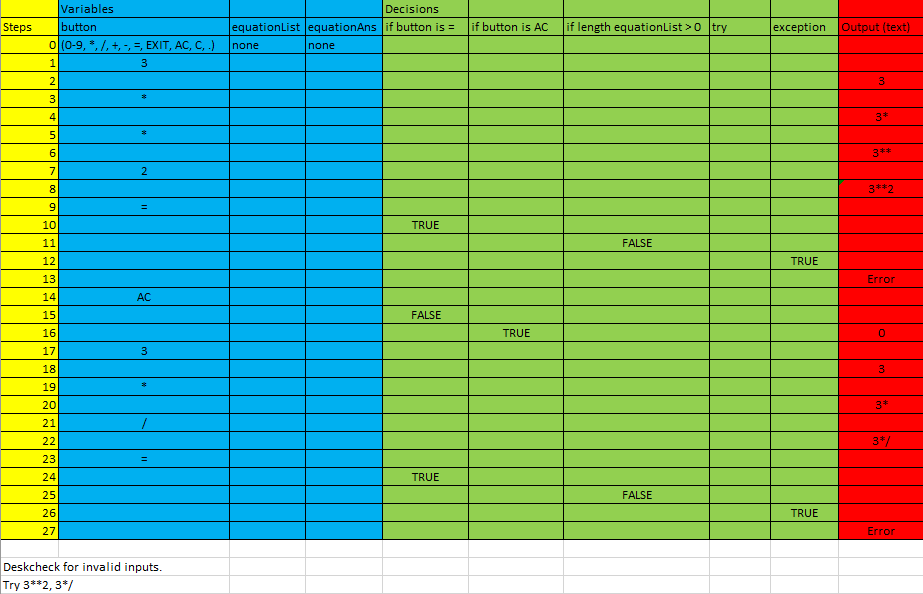
*Clearly indicate in a new version of your plan what extra steps may be needed in your program to ensure the sequence of actions correctly performs the task and has no unintended behaviour or consequences.*



Task 10: Create a testing table for invalid inputs for testing the program

*Add to your test data table to indicate what invalid inputs you would include for testing the program comprehensively.*

*Clearly indicate in a new version of your plan what extra steps may be needed in your program to make a robust and flexible program.*



**Versions**

Version One

Version one is very simple with only an entry widget and the buttons created manual. You can press the buttons, but they don’t do anything yet. The buttons are blue with the AC button red. Calculator code is very short.

*Next Step: Loop with buttons and when button is pressed, insert into entry widget.*

Version Two

Buttons are now done with a loop. Window now has a title at the top left. When you press a button, they now insert into the entry box. If you press the equals button it also produces an answer. The clear and cancel huttons do not work though. You can do basic calculations.

*Next Step: Working clear and delete buttons so that multiple calculations can be done.*

Version Three

This version uses the ttk theme that makes the calculator look nice for now, but will need to be changed eventually tos uit my stakeholder’s demands. There are status bars and the cancel/clear buttons both work. There is also a taskbar that can be used to see popups. There is also an exit button if the user wants to use that. Erros are no longer a problem with a try and except loop.

*Next Step: Leading zero error to be fixed, more robust code. Code own colours and fonts.*

Version Four

There is now key binding for the calculator and the leading zero erro has been fixed which was a big problem for me.

*Next Step: Code colours and fonts for calculator.*

Version Five

Calculator now has two classes. The calculator does not work though, but it is now in a format that will help me improve my coding for next year’s work.

*Next Step: Fix classes in code. Main problem is the inserting of text.*

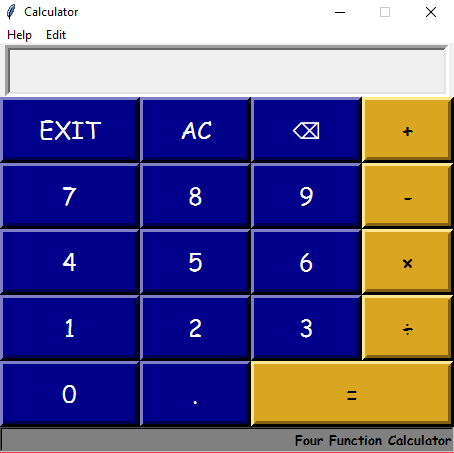
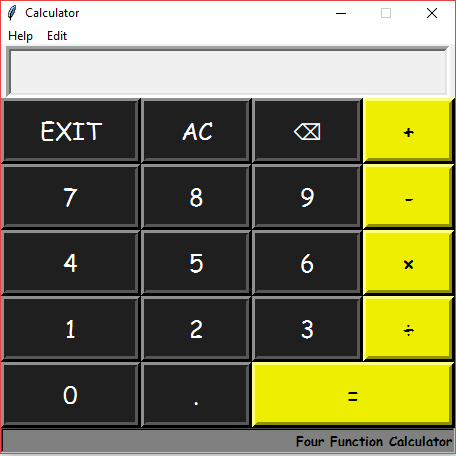
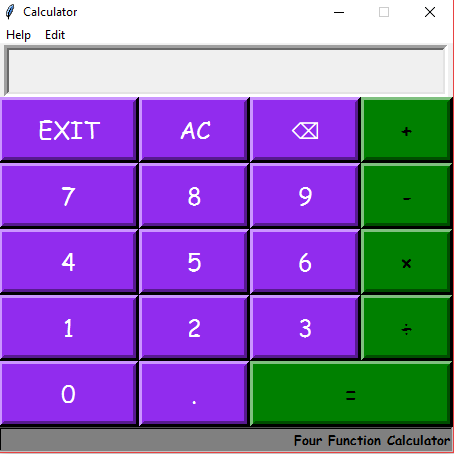
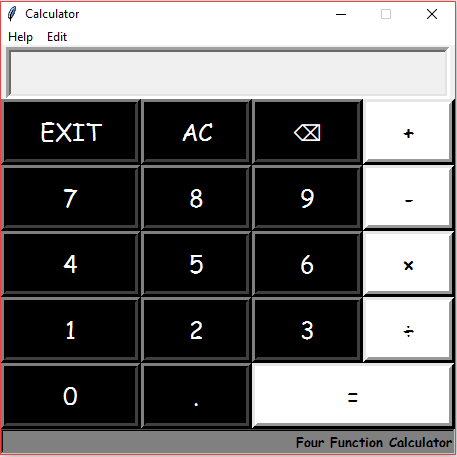
Version Six

Classes now work on calculator and it has the right colours and size of buttons. This is final version of calculator that will be presented to stakeholder.

**Colour**

The stakeholder and I decided to go with a combination of black and yellow. The black colour gives a more modern view and makes the calculator more appealing to millenials. The contrasting yellow is a very bright colour and is scientifucally proven to stand out next to dark colours. This is the same for the white text on the black buttons, which stands out more than black text on white buttons.

<https://www.smashingmagazine.com/2017/01/underestimated-power-color-mobile-app-design/>



Final design of calculator pictured left. The other possible choices for the calculator are above. I showed all four designs to the stakegholder and we agreed that this design was the best.

**Evaluation**

The calculator was a fun task to do that was very challenging. There were parts where I had to research throughouly through the manual and on line to complete it without any errors. It was a real step up from last year and looks like the program is become more and more suitable for people to use. It feels like I am able to code much more now and can actually make programs that people could use. For exampe, I could recode the rock paper scissors that we did last year with a GUI which would be much better to use.

In the future I would like to see the python program converted so people can use it without having to download python on their computer, but I hope to learn that next year. If I had to change anything about the calculator, I would like to make more functions so that older students can use it for their maths. Unfortunatley there was not enough time to do this. I would also like it so that the multiply/divide buttons can only be pressed once in a row so that there are no errors.

The stakeholder was very happy with my end result and says it would be useful for the students to use. I am pleased that I met the requirements and that the calculator looks good.