Assessment 3: Python group project

Demonstration and online submission; week 23; 10% - group project

Students will be working in pairs to write code related to a Python project. The final version of their software will be demonstrated to the instructors at week 23.

Students will be tasked with creating a basic calculator in Python, with a fully functional GUI, using any library they see fit to help implement the GUI

- Tutorials explaining how to implement Tkinter
- Resources for expanding Python knowledge in regard to mathematical operations

Originality in the design and implementation of a fully functional buttons is strongly encouraged, as is the use of programming tools beyond those explained in class. Additional grades are awarded for employing more complicated mathematical expressions on your calculator

Bare minimum you should submit is a calculator that can do the following mathematically expressions –

- **Multiplication/Division x /**
- **Subtraction/Additional - +**
- **Equal/Clear = C**
- **Should operate fully with precision, there should be no rounding or missing decimal places**

In addition, Multi-threading along with other programming techniques learnt throughout your course should be applied.
Final assessment will be done in the class in week 23. You will be allocated a 10 minute slot on that day to demonstrate and talk through your code during the class. We will mark your presentation and code to award you a grade on the Middlesex 20 point scale. No presence in a given time slot will result in fail mark.

Summarising, project 3 will be assessed based on quality of final submission, and knowledge demonstrated during the live presentation, according to the rubric below.

Project 3 evaluation rubric

Item	1-4	5-8	9-12	13-16	Fail (17-20)
Code 35 %	High quality code exhibiting deep knowledge and creativity	Code shows structure, understanding of programming tools and personal initiative	Code shows some personal initiative and good understanding of basic concepts	Code works, but is mostly based on given examples with little personal contribution	Code does not work, and shows little understanding of language/algorithms
Documentation 10 %	Advanced comments and documentation	Satisfactory comments and documentation	Basic but correct documentation and comments	Poor documentation and comments	Missing or minimal comments or documentation
Mathematical Operations 25 %	Complete and fully operational scientifical calculator, with at least 4 or more extra expressions than the basic requirement	Full basic requirements met, and successfully implemented complexed mathematical expressions	Full basic requirements met, and attempt at implementing more scientific mathematical expressions	Addition/Subtraction/ Multiplication/Divisio n present. One ore more expression not functioning	Addition/Subtraction, or one or more expressions not functioning correctly
User interface 15 %	Fully functional GUI, all aspects of a calculator taken into consideration	Tidy, and clean GUI with the ability to close and or minimize the software	Basic functionality, all mathematical expressions function	Some GUI functionality/ Inefficient or messy	No interface
Presentation 15 %	Very good and organised demonstration	Careful demonstration of code and user guide	Demonstration is clear but not very organised	Basic explanation of code and animations	No clear explanation, lack of understanding

For further sources of information, please refer to the following websites:

https://likegeeks.com/python-gui-examples-tkinter-tutorial/

https://www.tutorialspoint.com/python/python_gui_programming.htm