Advanced Software Engineering Lab (CS 480):

Lab 2: Design and build a software (website/standalone app/cell phone app) which can do basic arithmetic/trigonometric/logarithmic operations (“**+**”,”**-**“,”*”,”**/**”, ”**^**”, ***sin***, ***cos, tan***, ***cot***, ***ln, log***$_{10}$) with real numbers involving parenthesis (“**()**”) and curly brackets (“**{}**”). The calculator software should evaluate properly the mathematic formula based on the precedence of the operators, the parenthesis and the aforementioned mathematical functions.

Instructions

- Use the programming language of your choice to build the calculator software and the corresponding interface. The interface can be a GUI or just a text based interface. For details check the basic functionalities of a calculator. No external software library (except the math library if necessary) should be used to solve the problem! The source code should be in your code base. No need to implement a sin() function on your own, you can use one which is usually packed in the math libraries available in the different programming languages.
- The “**-**“ should be considered as unary and binary operator as well based on the syntax. Ex. “---3” (is three unary operators), “2-3” the “-“ is a binary operator.
- The mathematic expression evaluation (see PN, RPN and shunting-yard algorithms if this is the choice you select) should be implemented in the software code. Do not use ***eval()*** or similar type of internal evaluation functions! [Note: External source code can be used, but is to be explicitly mentioned in your source code!]
- All the software changes should be recorded in github or any other type of version control system.

Submission guidelines

- A zip file containing only the source files + a readme file (HowTo_Firstname_Lastname.txt) describing how to compile/execute/interpret from command line the code packed into a Source_Code_LastName_Firstname.zip file. The instructor/TA should be able to compile/run/interpret the code from command line using the instructions from the readme file. For command line see in Windows Command Prompt or Power Shell or in Linux/MacOS any regular shell of your choice. No executable file should be submitted!
- A short document describing what type of version control system has been considered, what are the advantages and disadvantages of the versioning system in use, and a proof (see a screenshot for example embedded in the document) about the commits. Name convention: Version_Control_Lastname_Firstname.pdf. This should be also included in the zip file.
- If the software is not running/compiling/executing on the school machines or similar environments (considering regular compilers/interpreters such as Java, JavaScript, Python, C/C++ (GNU), Go, Rust, Matlab, etc. running on Windows, Linux or MacOS) or the instructor/TA is explicitly requesting the software should be demoed “in-person” using the student’s equipment and software setup via some video chat solution (see Zoom, Teams, WebEx, etc.) --this is the preferable choice, or in person in one of the Samuelson building’s computer labs!

Grading criteria

1. The calculator should be able to correctly evaluate all kind of syntactically correct mathematic expressions. I.e. “-5.78+-(4-2.23)+sin(0)*cos(1)/(1+tan(2*ln(-3+2*(1.23+99.111)))=” (**5p.**)
2. The software should check for correct inputs. I.e. “**((**2+3/**)(**4**/-** = “ (**4p.**)
3. Usage of a version control system and proper description. (**1p.**)