Richard Robinson

SID: 917046226

CSC 413 Assignment 1

Due: Feb 14, 2017

**Calculator Project**

**Making and Executing a jar file:**

**Summary:**

The goal of this project was to create a simple calculator with an object-oriented approach.

This calculator uses an infix approach and is capable of:

**-Multiplication**

**-Division**

**-Addition**

**-Subtraction**

**-Exponentiation**

**-Use of Parentheses**

An operand stack and an operator stack are utilized. Each operator is an extension of an abstract class “Operator”. A hash table is used to determine if the characters within a string are a supported operator, an operand, or neither and then creates a corresponding object to be pushed upon the appropriate stack. Evaluation corresponds to the given priority of operators. If a right parenthesis is found operations will be executed until the matching left parenthesis is found and then the left will be popped off the stack.

This calculator works as intended and all valid inputs work.

The calculator works with integers, so rounding may occur upon division.

My project differs from the template in a few ways:

*I have a executeOnce() method within my operator class which pops two operands and an operator and executes the evaluation of those components .*

*The GUI is much different than the template*.

\*It is assumed that the user will use it correctly and does not handle or check for correct syntax.

**Usage:**

The EvaluatorUI is the GUI component to be used.

Alternatively, arguments can be passed to the EvaluatorTester class.

**Discussion:**

There was nothing astonishingly difficult about this project, but the object-oriented approach alongside hash table and various operand classes was a novelty to me and made me think of new ways of approaching problems. I will extend this concept of doing instead of asking “if” in the future and am extremely appreciative of this project for that reason.

My code is by no means optimized and does not error check for invalid arguments or syntax. I had a very positive experience with this project and will be capable of looking at future projects in a more object oriented fashion.

**UML:**

The documentation must include the following sections:  
\* Project introduction and overview (practice concisely summarizing technical work, and  
provide information on execution and development environment). Include scope of  
work (what were you tasked with completing, what did you complete)  
\* Instructions to compile as jar and execute (you will be penalized if this is not provided)  
\* Assumptions (what assumptions did you make in order to complete the assignment)  
\* Implementation discussion (I strongly recommend the use of graphical artifacts to help  
describe your system and its implementation: class diagrams, hierarchy, etc.  
Implementation decisions, code organization)  
\* Results and conclusions (what did you learn, future work, what challenges did you  
encounter and how did you overcome them)

Ex1

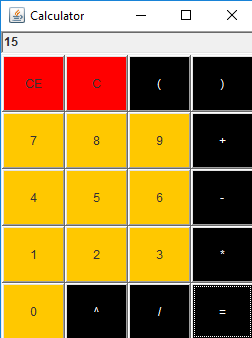
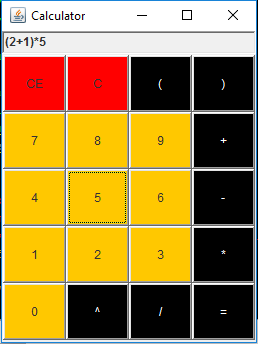


Ex2



**EX. 1**

(2+1)\*5 --------------------------- > Evaluated



**EX. 2**

(2^2)-4+4/2 ------------------------------------- > Evaluated

