COSC 201 – Assignment #3
Fall 09

Objective: Utilize stacks to make adjustments to the calculator in the book to do something a bit more complicated.

Expectations: You will code a calculator that allows for variables. What you should eventually get is an expression evaluator that will be simple, but the first step is what we call an interpreter which evaluates code for a specific language on the fly. You will need to handle six operations, +, -, *, /, ^ and =. Note that – can have multiple functionalities depending on the situation. Your code should be clean, concise, commented and correct. You are allowed to work in groups of at most 2 people for this assignment. If you work in a group of 2, you will submit only one set of code to me, but with both names and a description of what each person contributed to the project. For both 1-person and 2-person teams, be absolutely sure that no one else looks at your code and that you do not look at anyone else's code.

Testing: I will test a variety of inputs as well as incorrect and non-sensical expressions (like +*2). Be sure to catch errors as it relates to variables not being defined. We will make the calculator simpler by only allowing one character variable names (a-z, A-Z). Multiple character errors should be caught. Note that an error due to an undefined variable should not terminate the program, rather it should ask for the number for that variable. You can assume that I will test with multiple undefined variables as well as singletons.

Sample Output:

```
Enter an expression to be evaluated (type quit to quit):  x + 5 * 4
The variable x is not defined, what is the value of x? 12
Expression evaluated to 32.

Enter an expression to be evaluated (type quit to quit): x = 20
Expression evaluated x = 20
Enter an expression to be evaluated (type quit to quit): x + 22
Expression evaluated to 42

---
```

Grading rubric will be given out at least a week ahead of the due date.

DUE: December 7th by 11:59pm in the Digital DropBox.