Umar Kagzi

Question 2 Whitepaper

2. Write a prefix stack and show how it works (in main).

In Question 2 of the HW, we are asked to create a prefix stack in main and use a prefix expression to show how it works.

The first part of the program is a declaration of an array based stack, prefixStack, that we use throughout the program. We declare the basic functions such as push, pop, top, and constructors and destructors needed in the prefix program.

After declaring and initializing the class objects and their definitions, I created the *evaluate* function which would go through the prefix expression and be able to solve it properly.

First, I declared a stack and used a for loop to go through the values of the array based stack. I then checked if the topmost element is a digit or an operator using the built-in isdigit function. If it was a digit, the value would be pushed into the stack. If the value was an operator, the program would branch into a new else statement.

The purpose of the else statement is to branch out and calculate the stack digit values with the operators (+,-,*,/) correctly. I first declared two operators and popped them off the stack. The purpose of these two is to get the values of the numbers and be able to calculate them with the given operand. To find out what operand is used, I implemented a simple switch function which specifies the different cases of the operands and performs the appropriate calculation on them. For example, if a ‘+’ sign appeared, the function would add both values and push it onto the stack (and similarly for the other signs).

At the end of the program, I would simply enter in a prefix expression as a string and the function would parse and evaluate it using an array stack. The user would be able to get a correct prefix expression calculation!