

Lab Fb Reflections Document

Student: Joaquin Saldana

I created an RPN calculator class in which the constructor takes a pointer to a “Stack” class object.

Afterwards I created the respective functions for the following operators:

- Addition
- Subtraction
- Division
- Multiplication

Each of these functions performs the necessary popping of two integers from the Stack pointer passed in the constructor, performs the operand, prints the results for the user, and then stores the result in the stack.

The main function is an infinite while loop, in which the user is asked to enter an integer (positive or negative), an operator, or “q” to quit.

I used the string stream class to verify if the user entered a number or an operator, for the operator class, the RPNCalculator class has a function that check if it’s an operator.

Built in to the main function, once it’s identified it’s an operator, there are if statements that check if the user is requesting to add, subtract, multiply or divide.

I tested the code for the Stack and Queue class and found with the following test plan. In this test plan are my results for the RPN Calculator class and my main function:

| Test Case | Input Values | Driver Functions | Expected Outcome | Observed Outcomes |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Verify the Queue class and Stack class functions are working correctly Including the destructor to ensure there are no memory leaks upon destruction | Various integers | While loop pushing integers, popping integers, and peeking at integers | Verify the program is correctly linked integer nodes, removing integer nodes, and destroying the nodes in dynamic memory | No memory leaks and each function call stored the ints in stack order (first in last out) |

| | | | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|-------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| Verify the RPN calculator classes are correctly popping the numbers from the stack and performing the correct operation and result is popped back into the stack | Various integers and operators | While loop, with cout statements, and RPN Calculator class functions | For the user to be informed when an operations was performed successfully, with the numbers used for the operation and the amount/result popped into the stack | The functions worked correctly, and printed the correct numbers and results |
| Ensure the main function is detecting when the user is entering an operand, deleting the dynamically allocated stack object, and quitting the program when the user enters "q" | Various integers, operators, and "q" for quitting | Infinite while loop, with various if statement and string stream object | The program to quit when the user requests, and to perform the correct operative functions when the user enters +, -, *, or / | The calculator performed the correct functions, worked with both positive and negative numbers, and quit accordingly |