Amy Zhang

CSCI-313

HW2- Question 4: Stack Functions

For this problem, I used the array based stack class given to us on blackboard. In addition, I created a template function called printStack that will list and print all the elements in the stack. Within the printStack function, I declared an int variable called index that will store the value one less than the stackTop. This is because stackTop is pointing to an empty spot that is available for storing the next value. In our case, we want to print the first value from the stack top, so I did stackTop-1. In order to print all the values in the stack, I used a while loop and the condition is that as long as the index is greater or equal to 0 it will print the value at the specific index. Furthermore, within the while loop, it will print the value in the stack from top to bottom and every time a value is printed, we decrease the index by one in order to print the values below it.

Moving forward, I created another function called operator[] that will allow us to change values within a stack based on its position. The operator[] takes in one parameter, which is the index provided in the main function. Within the operator[] function, I included an if statement that will check if the index is out of bound. In my case, if the index is negative or greater than stackTop, then it is considered to be out of bound. If the index happens to be out of bound, a string will be thrown to notify the user. Before exiting this function, it will return the value at the input index. Since it’s returned by reference, therefore, it’s able to modify the value at the input index. For instance, in the main function where it says arrayStack[8]=17, it will use the operator[] function within the stackType class to assign the value 17 to the index 8 location.

In the main function, I created an empty array based stack called arrayStack that will be storing integer values. Next, I used a for loop to add values into the arrayStack. In this case, I added values [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]. Then, I used the object arrayStack that was created earlier to access the function printStack and print the values stored in arrayStack. In order to iterate through the stack and change one of the values based on its position, I used a for loop and operator[] function to do so. The following code performs those tasks: for(int i = 0; i < 10; i++){cout << arrayStack[i] << " ";} and arrayStack[8] = 17. Finally, after the value is changed at the desired position, the operator[] is used once again when I used a for loop to iterate through the stack and print all the values from top to bottom.