# CS 3358 Section 252 – Assignment 1

Due Date: February 19, 2023

This assignment has two parts pertaining to the stack implementation. The parts of code are given in the `.cpp` and `.h` files. The places you need to fill out in the code are marked by `// TODO`.

The stack implementation is supposed to be **array-based** in this assignment.

- (60 points) In `mystack.cpp`, implement the member functions of the class `MyStack`, which is a class for integer stacks.

- (40 points) In `stacktest.cpp`, complete the implementation of the function `PostfixTest()`, which uses an integer stack to evaluate postfix expressions. For simplicity, you can assume the postfix expression is input character by character (not an entire string), and each operand is a non-negative, single-digit integer, i.e., 0, 1, ..., 9. Nonetheless, you are still supposed to detect an invalid or illegal postfix expression input, e.g., “4 5 + -”.

Compilation:
You will need to compile `mystack.cpp` and `stacktest.cpp` separately to two `.o` files and then link them together to the executable. You may want to check out two PDF files under Module ‘References’ on Canvas regarding C++ programming in the Linux environment.

Submission:
You should submit your work via Canvas. You should pack `mystack.cpp`, `mystack.h`, and `stacktest.cpp` into a single `.zip` file to upload to Canvas. The `.zip` file should be named as `a1_yourNetID.zip`, such as `a1_gwc38.zip`.

Sample tests:
Note that success in getting the following test results does not guarantee the correctness of your work and thus does not guarantee a satisfactory grade, while failure in getting the following test results probably does indicate flaws in your work and you may lose points.

Test case 1 for stacktest:

```
Testing the basic functions of your stack...
Please enter the max size of your stack: 3
Please enter 'p' for push, 'o' for pop, 'e' for exit: p
Please enter an integer value you would like to push: 1
Please enter 'p' for push, 'o' for pop, 'e' for exit: p
Please enter an integer value you would like to push: 5
Please enter 'p' for push, 'o' for pop, 'e' for exit: p
Please enter an integer value you would like to push: 7
Please enter 'p' for push, 'o' for pop, 'e' for exit: p
Nothing can be pushed in since the stack is full!
Please enter 'p' for push, 'o' for pop, 'e' for exit: o
7 has been popped out.
Please enter 'p' for push, 'o' for pop, 'e' for exit: o
5 has been popped out.
Please enter 'p' for push, 'o' for pop, 'e' for exit: o
1 has been popped out.
Please enter 'p' for push, 'o' for pop, 'e' for exit: o
Nothing has been popped out since the stack is empty!
Please enter 'p' for push, 'o' for pop, 'e' for exit: e
Now, start to use a stack to evaluate postfix expressions...
Please enter the operands (integers 1~9) and operators (+, -, *, /) one by
one...
and enter '=' to indicate the end of the expression and to output the result.
2
5
7
+
*
1
-
=
The entered postfix expression results in 23
```

Test case 2 for stacktest:

```
Testing the basic functions of your stack...
Please enter the max size of your stack: 3
Please enter 'p' for push, 'o' for pop, 'e' for exit: e
Now, start to use a stack to evaluate postfix expressions...
Please enter the operands (integers 1~9) and operators (+, -, *, /) one by
one...
and enter '=' to indicate the end of the expression and to output the result.
4
5
+
-
Error! Invalid expression.
```

Test case 3 for `stacktest`:

```
Testing the basic functions of your stack...
Please enter the max size of your stack: 3
Please enter 'p' for push, 'o' for pop, 'e' for exit: e
Now, start to use a stack to evaluate postfix expressions...
Please enter the operands (integers 1~9) and operators (+, -, *, /) one by
one...
and enter '=' to indicate the end of the expression and to output the result.
7
2
+
4
5
-
=
Error! Invalid expression.
```