

This is an individual assignment. Seeking direct help from students, tutors, and websites such as chegg or stack overflow will be construed as a violation of the honor code.

Semester Project Part 1: Implementing a basic Stack

Data Structures and Analysis of Algorithms, akk5

Objectives

- To strengthen student's knowledge of C++ programming
- To give student experience in writing Data Structures for data types

Instructions

For this assignment you must write a program that implements a stack of integers. The stack should be implemented using classes and must support the push, pop, and peek stack operations, the isEmpty method, and overload operator<<. Your stack implementation should also handle popping or peeking at data from an empty stack without generating a runtime error.

Your program should test the correctness of your implementation in function main. You should provide test case to verify your stack works for the following cases:

1. Pushing an element of data onto a newly created stack
2. Pushing an element of data onto a non-empty stack
3. Pushing an element of data onto an empty stack (Note: a stack is empty when it is newly created and after the last element is popped off the stack)
4. Popping a value from a newly created stack
5. Popping a value from a non-empty stack
6. Popping a value from an empty stack
7. Performing a sequence of consecutive pushes and pops
8. Peeking at the top of a newly create stack with peek
9. Peeking at the top of a non-empty stack
10. Peeking at the top of an empty stack.

This is an individual assignment. Seeking direct help from students, tutors, and websites such as chegg or stack overflow will be construed as a violation of the honor code.

Grading Breakdown

Point Breakdown	
Structure	12 pts
The program has a header comment with the required information.	3 pts
The overall readability of the program.	3 pts
Program uses separate files for main and class definitions	3 pts
Program includes meaningful comments	3 pts
Syntax	18 pts
Implements Class Node correctly	9 pts
Implements Class Stack correctly	9 pts
Behavior	70 pts
Program returns correct values for the following	
<ul style="list-style-type: none"> Pushing a value onto a newly created stack 	7 pts
<ul style="list-style-type: none"> Pushing a value onto a non-empty stack 	7 pts
<ul style="list-style-type: none"> Pushing a value onto an empty stack 	7 pts
<ul style="list-style-type: none"> Popping a value from a non-empty stack 	7 pts
<ul style="list-style-type: none"> Popping a value from a newly created stack 	7 pts
<ul style="list-style-type: none"> Popping a value from an empty stack 	7 pts
<ul style="list-style-type: none"> A sequence of multiple pushes and pops 	7 pts
<ul style="list-style-type: none"> Peeking at an empty stack 	7 pts
<ul style="list-style-type: none"> Peeking at a non-empty stack 	7 pts
<ul style="list-style-type: none"> Peeking at an empty stack 	7 pts
Total Possible Points	100pts
Penalties	
Program does NOT compile	-100
Late up to 24 hrs	-30
Late more than 24hrs	-100

This is an individual assignment. Seeking direct help from students, tutors, and websites such as chegg or stack overflow will be construed as a violation of the honor code.

Header Comment

At the top of each program, type in the following comment:

```
/*  
  
Student Name: <student name>  
  
Student NetID: <student NetID>  
  
Compiler Used: <Visual Studio, GCC, etc.>  
  
Program Description:  
<Write a short description of the program.>  
  
*/
```

Example:

```
/*  
  
Student Name: John Smith  
  
Student NetID: jjjs123  
  
Compiler Used: Eclipse using MinGW  
  
Program Description:  
This program prints lots and lots of strings!!  
  
*/
```

Assignment Information

Due Date: 9/09/2018 (Section 1), 9/08/2018 (Section 3)

Files Expected:

1. Main.cpp – File containing function main
2. Stack.h - File containing the Stack and Node class structures and definitions.