7.1 Stack on the Heap

Due 21 Oct 2019 by 23:59 **Points** 0 **Submitting** an external tool

In this assignment, start with your code from Assignment 6.1 (Implementing a Stack). Just keep your *main()* function. Here, you are going to replace the implementation of your own *Stack* class with an implementation that stores the data in a linked list (compare zyBook, Section 12.4). For each *push* operation, a new element should be created and linked at the beginning of the linked list. For each *pop* operation, the element at the beginning of the list should be removed from the list.

Your new class *Stack* should look like the following one. Please note that your class needs to implement a default constructor, a copy constructor, and a destructor in order to work properly. (In the unlikely case that your code from 6.1 explicitly copies one *Stack* object onto another, instead of passing a *Stack* by-value to the *list* function, you will also have to implement a copy assignment operator; see zyBook, Section 12.10 in this case.)

```
class Stack {
    private:
        // your linked list goes here
    public:
        Stack();
        Stack(const Stack& original);
        ~Stack();
        bool isEmpty() const;
        int top() const;
        int pop();
        void push(int);
};
```

Use of arrays, a built-in stack class (or other container classes from std::) is not allowed.

Besides the new implementation of Stack, your program should behave exactly as with Assignment 6.1. The runtime tests are the same as well.

This tool needs to be loaded in a new browser window

Load 7.1 Stack on the Heap in a new window