Assignment 2 – Stacks

Comp310 – Object oriented data structures

# Topic(s)

Stack Applications and Implementations

# Readings & review

* [Carrano, Data Structures and Abstractions with Java, 3rd Edition:](http://instructors.coursesmart.com/9780136100997/id1) Chapters 5 & 6
* Stack ADT & implementation lectures

# Objectives

Be able to:

* Implement a Stack ADT
* Apply your implementation to a **Last-In-First-Out (LIFO)** based project.

# instructions

1. **(Optional) Select a Partner.** Please let instructor know **BEFORE YOU START** who you will work with.
2. Read over the assignment and ASK QUESTIONS about anything that you don’t understand (before you start).
3. Do the [problems](#_Problems) listed in the next section. .
4. Be sure to follow [Good Programming Practices](#_Good_programming_practices), also listed below.
5. Individually answer the [Analysis and Feedback](#_Analysis_/_Summary:) questions as a Journal entry on blackboard.
6. Follow instructions for [submitting your work](#_Submitting_your_work:).
7. **Demo your project to instructor** (required for grading).

# Problems

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ch | Project | | Page | Notes |
| 6 | **#1** | | 154 | Stack **implementation.** |
| 5 | **Select one** | **#6** | 137 | Stack **application**. You may also propose your own project as an application of Stacks. Start early, ask questions and have fun! |
| **#8** | 138 |

# Good programming practices

**For all classes**

* Make your instance variables **private**
* Include **constructors** to initialize your instance variables.
* Derived class constructors should **leave initialization of super class instance variables** to the super classes’ constructors:
  + Remember the call to the super classes constructor is: **super( <init1>, <init2>,..)**
* Include **Accessor** and **Mutator** methods for all instance variables *(please ask if you’re not sure what these are)*
* Include a **main** method for testing (unless it’s an abstract class) and test before you proceed
* Add comments to your code, not just so it’s easier for other readers, but also so it’s easier for you to remember your logic.

# Analysis / Summary:

**In your Journal on Blackboard, please answer the following questions:**

1. If you worked in pairs:
   1. How did you “divide up” the work so that each student still met the objectives for the assignment (i.e., learned, understood and applied the concepts).
   2. What was your contribution?
   3. How did you coordinate code changes/testing?
   4. Other observations about working with a partner?
2. Where did you have trouble with this assignment? How did you move forward? What topics still confuse you?
3. What did you learn from this assignment? (Please be specific)
4. How could this assignment be improved in the future?

# Submitting your work:

1. Make sure that your name(s) are in all of your project files. (Note: for students who work in pairs, each must include a summary file with just their name on it)
2. If you have more than one file for your solution, make a .zip file for your project
3. In Blackboard, attach your solution file to the submission for this assignment. (Note: for students who work in pairs, only one of you should submit to the assignment)