# COE528 (Fall 2019)

# Lab4

**General Lab Rule**

* **Lab attendance is mandatory and will be recorded weekly**. You lose 10% of the lab mark if you do not attend the lab sessions. (For a lab with two sessions, it means that if you do not attend both sessions and for a lab with one session, it means that if you do not attend the only one session).
* **Due date: 11:59pm, the day before you scheduled lab 5 session.**
  + The penalty for up to 8 hours delay in submission is 20% of your mark
  + No acceptance after 8 hours delay
* You must include the duly filled and signed standard cover page with your submission. The cover page can be found on the departmental web site: [Standard Assignment/Lab Cover Page](http://www.ee.ryerson.ca/guides/Standard_Cover_Page_Assignments.pdf)

All the java files in this lab should have the following package declaration:

package coe528.lab4;

**Duration: one week.**

## Objectives

* Provide and implement abstraction function and rep invariant.

### Exercise 1:

**Abstract Concept of a stack of distinct strings:**

A stack of distinct strings, *p*, is a collection of strings with a *top* where a string cannot exist more than once in the collection. For example, the collection of strings{“ab”, “cd”, “ae”, “bd”} is a stack with *top* = “bd”. The only operations supported are addition of a string at the *top* known as *push*, and deletion of a string from the *top* known as *pop*.

**Implementation of a stack of distinct strings:**

The following class, StackOfDistinctStrings , represents a stack of distinct strings.

In the Netbeans program, click on Project > New Project and save it as "Ex1" on your lab4 directory.

Create a new class called “StackOfDistinctStrings” and copy the code in it. For this class:

a) Write the abstraction function in the Overview clause.

b) Write the rep invariant in the Overview clause.

b) Fill in the body of the method repOK().

b) Fill in the body of the method toString ().

import java.util.ArrayList;

public class StackOfDistinctStrings {

// Overview: StacksOfDistinctStrings are mutable, bounded

// collection of distinct strings that operate in

// LIFO (Last-In-First-Out) order.

//

// **The abstraction function is:**

// **a)** **Write the abstraction function here**

//

//

//

//

// **The rep invariant is:**

// **b)** **Write the rep invariant here**

//

//

//

//

//the rep

private ArrayList<String> items;

// constructor

public StackOfDistinctStrings() {

// EFFECTS: Creates a new StackOfDistinctStrings object

items = new ArrayList<String>();

}

public void push(String element) throws Exception {

// MODIFIES: this

// EFFECTS: Appends the element at the top of the stack

// if the element is not in the stack, otherwise

// does nothing.

if(element == null) throw new Exception();

if(false == items.contains(element))

items.add(element);

}

public String pop() throws Exception {

// MODIFIES: this

// EFFECTS: Removes an element from the top of the stack

if (items.size() == 0) throw new Exception();

return items.remove(items.size()-1);

}

public boolean repOK() {

// EFFECTS: Returns true if the rep invariant holds for this

// object; otherwise returns false

**// c) Write the code for the repOK() here**

}

public String toString() {

// EFFECTS: Returns a string that contains the strings in the

// stack and the top element. Implements the

// abstraction function.

**// d) Write the code for the toString() here**

}

}