# COE528 (W 2022)
# Lab3

**Duration: one week.**

**Note:**

1. Every lab assignment must be done individually.
2. When you name a folder or a file, you **should** avoid spaces in those names. For example, if you need to name a folder as **GreenApple**, you should name it as **GreenApple** instead of **Green Apple**

## Objective

- Provide and implement abstraction function and rep invariant.

## Queue of distinct strings

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

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

**Implementation of a queue of distinct strings:**

The following class, `QueueOfDistinctStrings` , represents a queue of distinct strings.

```
import java.util.ArrayList;
public class QueueOfDistinctStrings {

    // Overview: QueueOfDistinctStrings are mutable, bounded
    // collection of distinct strings that operate in
    // FIFO (First-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 QueueOfDistinctStrings() {
    // EFFECTS: Creates a new QueueOfDistinctStrings object
    items = new ArrayList<String>();
}


// MODIFIES: this
// EFFECTS: Appends the element at the end of the queue
//          if the element is not in the queue, otherwise
//          does nothing.
public void enqueue(String element) throws Exception {
    if(element == null) throw new Exception();
    if(false == items.contains(element))
        items.add(element);
}


public String dequeue() throws Exception {
    // MODIFIES: this
    // EFFECTS: Removes an element from the front of the queue
    if (items.size() == 0) throw new Exception();
    return items.remove(0);
}



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
        //          queue, the front element and the end element.
        //          Implements the abstraction function.
        // d) Write the code for the toString() here




    }
}
```

## Step 1: *Create a Netbeans project*

1. Create a Netbeans project called `coe528Lab3`
2. Create a Java class called `QueueOfDistinctStrings.` Set the package to `lab3.`
3. Copy the above code in it.
4. Do the following:
   (a) Write the abstraction function in the **Overview** clause.
   (b) Write the rep invariant in the **Overview** clause.
   (c) Fill in the body of the method `repOK`().
   (d) Fill in the body of the method `toString`().

## Step 2: Submit your lab

**Deadline: See announcement in D2L for deadline.**

Create a folder. Name it as YourLastname_YourFirstname_coe528_Labnumber.
Example: If student name is John Smith, the name of folder should be Smith_John_coe528_Lab3.

Copy your Netbeans project folder in the above folder.

The above folder must also contain a duly filled and signed standard cover page. The cover page can be found on the departmental web site:
Standard Assignment/Lab Cover Page

Compress the above folder as a single zip file that is named according to the following rule:
YourLastname_YourFirstname_coe528_Labnumber.zip.
Example: Smith_John_coe528_Lab3.zip.

**Upload the above zip file on D2L through the "Assessment" > "Assignments" link.**

**Within the deadline, you may re-submit (i.e. re-upload) the aforementioned zip file. However, note that your latest submission will always overwrite your previous submission.**

**Submission by email is NOT accepted.**

**Note:** If the code does not compile, the submission will receive a ZERO mark.