SE 2205: Algorithms and Data Structures for Object-Oriented Design
Lab 2

Assigned **Jan 29, 2019**.

1 Objectives

In this lab, you will implement a finite queue using two stacks.

Problem

Part A

Implement the generic array-based stack class called `ArrayStack<E>` based on the following interface:

```
public interface Stack<E>{
  int size();
  boolean isEmpty();
  void push(E e);
  E top();
  E pop();
}
```

where the first function returns the number of elements in the stack, the second checks whether the stack is empty, the third inserts a new node into the stack, the fourth reads the value in the top-most node and returns this and the final function removes a node from the Stack. Remember that the Stack is based on the LIFO principle.

Part B

Next, you will implement the Queue interface defined below **using only two stacks** based on the class you have implemented in the above by defining the class `SQueue<E>`:

```
public interface Queue<E>{
  int size();
  boolean isEmpty();
  E first();
  void enqueue(E node);
  E dequeue();
}
```

where the fist function returns the number of elements in the queue, the second checks whether the queue is empty, the third reads the first element in the queue and returns this, the fourth inserts a new node into the queue and the final function removes a node in the queue. Remember that the Queue is based on the FIFO principle. The size of the queue will be supplied in the argument of the constructor.

Part C

You will test your implementation by implementing the `Test` class consisting of the `main` function. In this function, you will `enqueue` five elements $\{1, 2, 3, 4, 5\}$ and print the values as you `dequeue` these. The order of these elements being printed must be the same as the insertion process.