C S 272/463 Introduction to data structures
Fall 2019
Lab 8: Generic Programming, Implement and Use Stacks

1 Learning objectives

Objective 1 (stack), Objective 5 (generic programming), Objective 6, Objective 7

2 Requirements

1 (5 pts) Write a generic interface for stack and put the code in **StackInterface.java**. This interface should include five functions: push, pop, top, size, and isEmpty.

2 (5 pts) Write a generic class for the node in singly linked lists and put the code in **SNode.java**.

3 (35 pts) Implement **LinkStack.java** with the following detailed requirements.

   (1) (5 pts) It has ONLY one instance variable, which is a generic node of type `SNode`.

   (2) (30 pts) LinkStack should implement the `StackInterface` interface and implement all the methods declared in ths interface. (Each method carries 7pts)

4 (35 pts) Implement **ArraylistStack.java** with the following detailed requirements.

   (1) (5 pts) It has one instance variable: an *arraylist* with a generic data type.

   (2) (30 pts) `ArrayListStack` should implement `StackInterface` interface and implement all the methods declared in this interface. (Each method carries 7pts)

5 (10 pts) You need to design test cases to test your functios in **ArraylistStack.java** and **LinkStack.java** thoroughly. If your test cases cannot cover some important conditions, points may be deducted. Please put your test case files to **StackTest.java**.

6 (10 pts) **NQueen.java**: Use either **stack that you implemented** to solve the N-queen problem. Your design needs to follow the logic in the lecture notes. You can also use the program project 10 at page 358 as reference. The parameter should be N (in the range of [1, 16]). The result should print queens at proper positions. For example the solution at page 358 should be printed as

```
Q - - - -
- - Q - -
- - - - Q
- Q - - -
- - - Q -
```

3 Note

- **Specifications** for all your classes and methods:
  Please properly explain (1) the functionality of the methods, (2) the parameters, (3) the return values, (4) the pre-conditions if there is any;
  Please use inline comments, meaningful variable names, indentation, formatting, and whitespace throughout your program to improve its readability.
- You can (but are not required to) design and implement other facilitating methods (E.g., other get and set methods, toString method) to finish the implementation of the required methods.

4 Submission

Submit through canvas a zipped file containing your java file(s) (not `.class` files).

5 Grading Criteria

(1) The score allocation is beside the questions.

(2) Please make sure that you test your code **thoroughly** by considering all possible test cases. Your code may be tested using more test cases.