Department of Electrical & Computer Engineering

North South University

Assignment 1

Submitted by: Kazi Sakib Ahmad

ID: 1510702042

Course: CSE427

Section: 01

Semester: Spring 2019

Submitted to: Mr. Shaikh Shawon Arefin Shimon

Lecturer, Department of ECE,

North South University

Ans. to the Question (a)

I have tried to do implement the `GenericStack` class using Test Driven Development (TDD). In order to do TDD for implementing I had to use a wide range of variables to write test cases. As it was asked to implement generic stack so I have written test cases by following input space partition and used variables from following data types:

1. Positive integers
2. Negative integers
3. Character variables
4. Strings and set of strings
5. Positive floating point numbers
6. Negative floating point numbers

To implement `GenericStack` class I was asked to must implement assuming the following method signatures:

```
1. public GenericStack();
2. public void push (object X)
3. public Object pop();
4. public Boolean isEmpty();
```

Finally I ended up implementing the following methods also for implementing the usability of `GenericStack` class

```
1. public int size() &
2. public Object topItem()
```

I have tried Unit Testing and Input Space Partitioning as a part of TDD.

Ans. to the Question (b)

While implementing `GenericStack` class as I have to keep in mind the possibilities of different inputs variables from different domain of data types. So I have used inputs of different characteristics and also tried of making combinations of different input variables while testing test cases. I am defining the characteristics of all input variables I used in test cases below:

In the `GenericStack` I have assumed each input as a single object. So while pushing into the stack I assumed each input as Object regardless of the data type of any particular input.

1. Single character
2. Set of characters or single strings
3. Set of strings having whitespaces between two adjacent
4. Positive integer number
5. Negative integer number
6. Positive floating point number
7. Negative floating point number

Mainly I have input having these 7 kinds of characteristics.

Ans. to the Question (c)

As mentioned earlier, whatever the input is the GenericStack should assume it as an Object and then process further. But there are some common characteristics of the input variables. They are;

1. It can be either a valid input or an invalid input. But as the stack is defining each input as single object so every input should be stacked as when it is pushed.
2. An input may have wide range of other similar input. As an example, I have used integers as input. But as an input space partition the domain of integers is so big. So I divided it into 2 sub domains (i.e. positive integers and negative integers) and tested for both sub-domain input spaces.
3. In case of using multiple inputs as a set, the input set can integrity factor or even doesn't have it (using inputs from different domain as a set)

Ans. to the Question (d)

The characteristics of the inputs I used in the test cases are partitioned in blocks below:

Characteristic	**Input variable Used**
Single character	`'a', 'b', 'c'`
Set of characters or single string	`"push1", "push2", "push3", "Kazi", "Sakib", "My", "Name", "Dabbe", "Siccin", "Anabelle", "Ahmad", "student"`
Set of strings having whitespaces between two adjacent	`"pushing for pop", "push 2 for pop", "here is a push", "Kazi Sakib",`
Positive integer number	`100, 10, 1, 23, 100006, 12`
Negative integer number	`-152, -65`
Positive floating point number	`6.333, 3.2, 102.66`
Negative floating point number	`-602.33, -10025.6, -63.5`

Ans. to the Question (e)

Two values for each block is defined below:

Input Block	**Input variable Used**
Character block	`'a', 'b'`
String block	`"push1", "student"`
Integer block	`100006, -65`
Float block	`6.333, -10025.6`