National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	Software Construction & Development	Course Code:	SE3001
	Degree Program:	BSSE	Semester:	Fall 2023
	Due Date:	Sunday, October 8, 2023 11:59 PM	Total Marks:	10 + 10 + 10 + 10 = 40
	Section:	BSE-5B	Page(s):	3
	Exam Type:	Assignment 2		

Instruction/Notes:

- Plagiarism is a serious offense and will not be tolerated.
- Any student found to have copied code from any online/offline source(ChatGPT) will receive a grade of zero for the assignment.
- No assignment will be accepted after the due date.
- Submit only .java files by zipping it and you must follow the criteria of submission by renaming your zip file as your RollNo.zip i.e 21L-1234.zip.
- Make your git repository public and insert the link in private comment

ANY FILES VIOLATING THE SUBMISSION FORMAT WILL NOT BE CONSIDERED

Feel free to contact TA in case of any questions or before taking any assumptions.

Using Github with REGULAR commits and PRs in the assignment is mandatory. Marks: [10]

Q1: You have been assigned the task of creating a data structure called **GenericStack** to handle elements of different data types. This will help in managing various types of data efficiently in different parts of your application. **[10]**

Requirements:

Create a generic class called **`GenericStack`** that can hold elements of any data type using Java Generics. Your class should support the following methods:

`push(T item)`: This method should push an item of type T onto the stack.

`pop()`: This method should remove and return the top item from the stack. If the stack is empty, throw an EmptyStackException.

`isEmpty()`: This method should return true if the stack is empty, false otherwise.

`size()`: This method should return the number of elements in the stack.

Also Implement all errors using the **`ExceptionHandling`** class which is to be thrown like when attempting to pop from an empty stack or invalid character input. Think of examples like this. Ensure that the exception message provides meaningful information.

Create a simple program that demonstrates the use of your GenericStack with both primitive and reference data types.

Sample Input	Sample Output

`Enter Stack Size: 3`
`Enter Stack Values`
 	
`1`
 	
`2`
 	
`3`
 	
`Perform Stack Operations`
 	
`pop`
 	
`pop`
 	
`size`
 	
`isEmpty`
 	
`Quit`
	`Stack size: 3`
	
	`Popped: 3`
	
	`Popped: 2`
	
	`Stack size: 1`
	
	`Stack is not empty.`
	
	`Program Exited!`

Q2: Implement a program to find the first non-repeating character in a character stream efficiently in a time complexity of O(1). **[10]**

Requirements:

1. Implement the **`FirstNonRepeatingStream`** class with the following methods:

- **`FirstNonRepeatingStream()`**: Constructor that initializes the data structure.
- **`void add(char c)`**: Add a character to the stream.
- **`char getFirstNonRepeating()`**: Retrieve the first non-repeating character from the stream. Return '-' if no such character exists.

Note: You can use any data structures/collections available in Java, but you must ensure that both add and getFirstNonRepeating operations are ***O(1)*** in time complexity.

Sample Input	Sample Output
`Stream: ['a','b', 'a']`	`b`
`Stream: ['a','b', 'a', 'b']`	`-`
`Stream: ['a','b', 'b']`	`a`

Q3: You are given a list of intervals, where each interval is represented as a pair of integers [start, end], your task is to merge any overlapping intervals to print a list of non-overlapping intervals. **[10]**

Write a function **`mergeIntervals(List<int[]> intervals)`** that accomplishes this task with the following constraints:

- You must solve the problem using O(N) time complexity.
- You are not allowed to create any new list.
- You can assume that the list will be provided in sorted order with respect to start time.

Sample Input	Sample Output
[**[1, 3],** **[2, 6],** **[8, 10],** **[15, 18]** **]**	**[** **[1, 6],** **[8, 10],** **[15, 18]** **]**

BEST OF LUCK