Lab Assignment-10

ROLL: 2005535 | NAME: SAHIL SINGH | DATE: 21/04/22

QUES 1: Write a program to perform following operations on user entered strings

i) Change the case of the string
ii) Reverse the string
iii) Compare two strings
iv) Insert one string into another string.

SOLUTION:

```
import java.util.*;
class Lab10Q1 {
    static String changeCase(String s) {
        String ans = "";
        int l = s.length();
        for (int i = 0; i < l; i++) {
            char c = s.charAt(i);
            if (c >= 97 && c <= 122) {
                ans += (char) (c - 32);
            } else if (c >= 65 && c <= 90) {
                ans += (char) (c + 32);
            } else {
                ans += (char) c;
            }
        }
        return ans;
    }
    static String Reverse(String s) {
        String ans = "";
        int l = s.length();
        for (int i = 0; i < l; i++) {
            ans = s.charAt(i) + ans;
        }
        return ans;
    }
    static String Compare(String s1, String s2) {
        String ans = (s1.compareTo(s2) > 0) ? s1 : s2;
        return ans;
    }
    static String Insert(String s1, String s2, int i) {
        String ans = s1.substring(0, i) + s2 + s1.substring(i, s1.length());
        return ans;
    }
    public static void main(String args[]) {
        Scanner inp = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String s = inp.nextLine();
        String changed = changeCase(s);
        System.out.println("Changed Case String: " + changed);
        String reversed = Reverse(s);
        System.out.println("Reversed String: " + reversed);
        System.out.print("Enter one more String to compare: ");
        String s1 = inp.nextLine();
        String greater = Compare(s, s1);
        System.out.println("Greater String: " + greater);
        System.out.print("Enter the index where you want to enter 2nd string in the first:
");
        int i = inp.nextInt();
        String inserted = Insert(s, s1, i);
        System.out.println("Inserted String at index" + i + " : " + inserted);
        inp.close();
    }
}
```

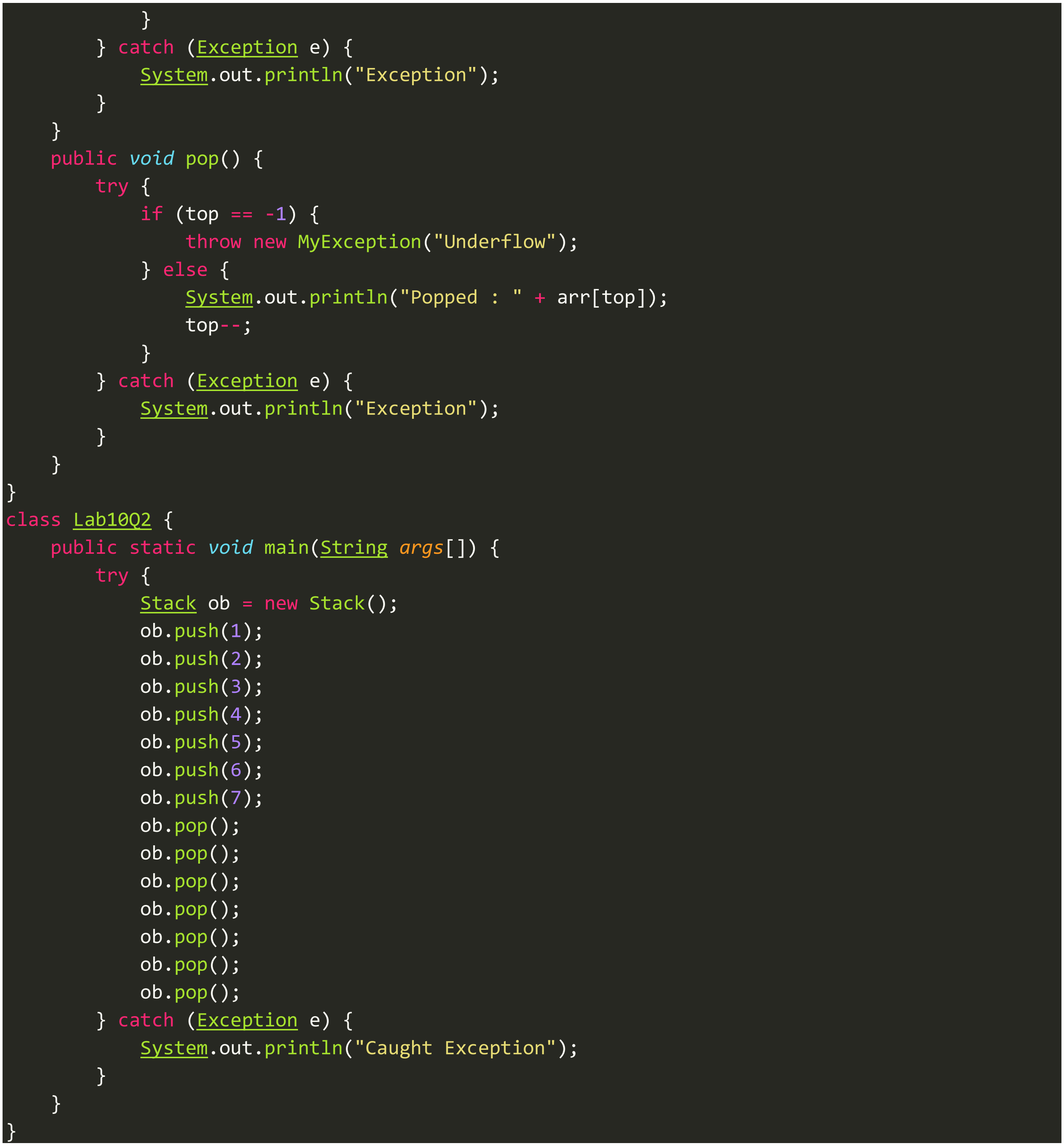
OUTPUT:

```
Enter a string: Sahil Singh
Changed Case String: sAHIL sINGH
Reversed String: hgniS lihaS
Enter one more String to compare: Huihuihui
Greater String: Sahil Singh
Enter the index where you want to enter 2nd string in the first: 6
Inserted String at index6 : Sahil HuihuihuiSingh
```

QUES 2: Write a java program to implement a stack class having methods push() and pop(). These methods must be designed to throw user defined exception when the stack is empty or full.

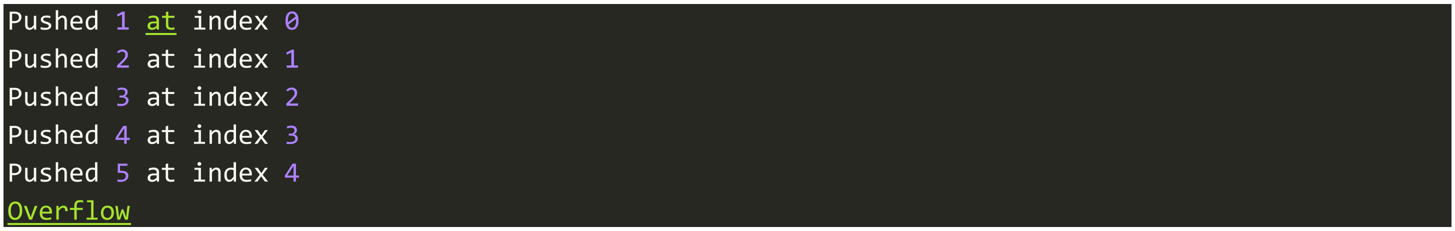
SOLUTION:

```
class MyException extends Exception {
    public MyException(String str) {
        System.out.println(str);
    }
}
class Stack {
    int arr[] = new int[5];
    int top;
    public Stack() {
        top = -1;
    }
    public void push(int a) {
        try {
            if (top == 4) {
                throw new MyException("Overflow");
            } else {
                top++;
                arr[top] = a;
                System.out.println("Pushed " + arr[top] + " at index " + top);
```



```
            }
        } catch (Exception e) {
            System.out.println("Exception");
        }
    }
    public void pop() {
        try {
            if (top == -1) {
                throw new MyException("Underflow");
            } else {
                System.out.println("Popped : " + arr[top]);
                top--;
            }
        } catch (Exception e) {
            System.out.println("Exception");
        }
    }
}
class Lab10Q2 {
    public static void main(String args[]) {
        try {
            Stack ob = new Stack();
            ob.push(1);
            ob.push(2);
            ob.push(3);
            ob.push(4);
            ob.push(5);
            ob.push(6);
            ob.push(7);
            ob.pop();
            ob.pop();
            ob.pop();
            ob.pop();
            ob.pop();
            ob.pop();
            ob.pop();
        } catch (Exception e) {
            System.out.println("Caught Exception");
        }
    }
}
```

OUTPUT:



```
Pushed 1 at index 0
Pushed 2 at index 1
Pushed 3 at index 2
Pushed 4 at index 3
Pushed 5 at index 4
Overflow
```

```
Exception
Overflow
Exception
Popped : 5
Popped : 4
Popped : 3
Popped : 2
Popped : 1
Underflow
Exception
Underflow
Exception
```

QUES 3: Write a java program to create Account with 500 rupee minimum balance, deposit amount, withdraw amount and also throws LessBalanceException which returns the statement that says withdraw amount is not valid.

SOLUTION:

```
class LessBalanceException extends Exception {
    public LessBalanceException(String str) {
        System.out.println(str);
    }
}
class Lab10Q3 {
    public static void main(String args[]) {
        int current = 1000;
        try {
            int withdraw = 700;
            if ((current - withdraw) < 500) {
                throw new LessBalanceException("Withdraw Not possible as amount is not 
sufficient");
            } else {
                current -= withdraw;
                System.out.println("Amount successfully withdrawn, remaining balance : " + 
current);
            }
        } catch (Exception e) {
            System.out.println("Caught Exception");
        }
    }
}
```

OUTPUT:

```
Withdraw Not possible as amount is not sufficient
Caught Exception
```

QUES 4: Create an user defined exception named Check Argument to check the number of arguments passed through command line. If the number of arguments is less than four, throw the Check Argument exception, else print the addition of squares of all the four elements.

SOLUTION:

```
class CheckArgumentException extends Exception {
    CheckArgumentException(String str) {
        System.out.println(str);
    }
}
public class Lab10Q4 {
    public static int calculateSum(int args, int[] argumentsArray) throws
CheckArgumentException {
        int sum = 0;
        if (args < 4)
            throw new CheckArgumentException("The Number of Arguments passed in the CLI is
less than 4");
        else {
            for (int i = 0; i < args; i++) {
                sum = sum + (argumentsArray[i] * argumentsArray[i]);
            }
        }
        return sum;
    }
    public static void main(String[] args) {

        int[] argumentsArray = new int[args.length];
        for (int i = 0; i < args.length; i++) {
            argumentsArray[i] = Integer.parseInt(args[i]);
        }
        try {
            int result = calculateSum(args.length, argumentsArray);
            System.out.println("The Sum of the Sqaures of the numbres is: " + result);
        } catch (CheckArgumentException e) {
            System.out.println(e);
        }
    }
}
```

OUTPUT:

```
PS D:\Labs\wt_lab\Lab 10> d:; cd 'd:\Labs\wt_lab\Lab 10'; & 'C:\Program Files\Java\jdk-
14.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
'C:\Users\KIIT\AppData\Roaming\Code\User\workspaceStorage\258a4bb52a5b50cd0afc0c8b79387a19\re
dhat.java\jdt_ws\Lab 10_8b3b0ba6\bin' 'Lab10Q4' 1 2 3 4
The Sum of the Sqaures of the numbres is: 30

PS D:\Labs\wt_lab\Lab 10>  d:; cd 'd:\Labs\wt_lab\Lab 10'; & 'C:\Program Files\Java\jdk-
14.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
'C:\Users\KIIT\AppData\Roaming\Code\User\workspaceStorage\258a4bb52a5b50cd0afc0c8b79387a19\re
dhat.java\jdt_ws\Lab 10_8b3b0ba6\bin' 'Lab10Q4' 1 2

The Number of Arguments passed in the CLI is less than 4

CheckArgumentException
```
