LAB # 01

INTRODUCTION TO STRING POOL, LITERALS, AND WRAPPER CLASSES

OBJECTIVE: To study the concepts of String Constant Pool, String literals, String immutability and Wrapper classes.

LAB TASKS

1. Write a program that initialize five different strings using all the above mentioned ways, i.e.

   a) string literals

   b) new keyword

also use intern method and show string immutability.

```java
package lab.taskk;
public class LabTaskk {
    public static void main(String[] args) {
        String s1="python";
        String s2="java";
        String s3="c++";
        String s4="rust";
        String s5="go";
        String s6="javascript";
        String s7=new String("python");
        String s8=new String("java");
        String st9=new String("c++");
        String s10=new String("rust");
        String s11=new String("go");
        String s12=new String("javascript");
        String i1=s1.intern();
        System.out.println("i1="+i1);
        String i2=s2.intern();
        System.out.println("i2="+i2);
        String i3=s3.intern();
        System.out.println("i3="+i3);
        String i4=s4.intern();
        System.out.println("i4="+i4);
        String i5=s5.intern();
        System.out.println("i5="+i5);
        System.out.println("");
        String str="sara";
        str.concat("laiba");
        System.out.println("immutable="+str);
        String T="jav";
        T.concat("savera");
        System.out.println("immutable="+T);
        String S="AREESHA";
        S.concat("zainab");
        System.out.println("immutable="+S);
        String A="zainab";
        A.concat("sania");
        System.out.println("immutable="+A);
```

```
System.out.println("immutable="+str);
String T="jav";
T.concat("savera");
System.out.println("immutable="+T);
String S="AREESHA";
S.concat("zainab");
System.out.println("immutable="+S);
String A="zainab";
A.concat("sania");
System.out.println("immutable="+A);
String B="alice";
str.concat("jhon");
System.out.println("immutable="+B);
```

```
run:
i1=python
i2=java
i3=c++
i4=rust
i5=go

immutable=sara
immutable=jav
immutable=AREESHA
immutable=zainab
immutable=alice
```

2. Write a program to convert primitive data type Double into its respective wrapper object.

```
package lab.taskk;
public class LabTaskk {
    public static void main(String[] args) {
        double n1=50.5;
        Double n2=n1;
        System.out.println("pirmitive data type:"+n1);
        System.out.println("Wrapper object:"+n2);
```

```
run:
pirmitive data type:50.5
Wrapper object:50.5
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Write a program that initialize five different strings and perform the following operations.

a) Concatenate all five stings.

b) Convert fourth string to uppercase.

c) Find the substring from the concatenated string from 8 to onward

```java
package lab.taskk;
public class LabTaskk {
    public static void main(String[] args) {
        String s1="python";
        String s2="java";
        String s3="c++";
        String s4="rust";
        String s5="go";
        String str=s1 + s2 + s3 + s4 + s5;
        System.out.println("Concatination: "+str);
        System.out.println("Upper case:"+s1.toUpperCase());
        System.out.println("Substring:"+str.substring(8));
```

```
run:
Concatination: pythonjavac++rustgo
Upper case:PYTHON
Substring:vac++rustgo
BUILD SUCCESSFUL (total time: 0 seconds)
```

4.You are given two strings word1 and word2. Merge the strings by adding letters in alternating order, starting with word1. If a string is longer than the other, append the additional letters onto the end of the merged string. Return *the merged string.*

Example:

Input: word1 = "abc", word2 = "pqr"
Output: "apbqcr"
Explanation: The merged string will be merged as so:
word1: a b c word2:
p q r
merged: a p b q c r

```java
package lab.taskk;
public class LabTaskk {

        public static String mergeAlternately(String word1, String word2) {

        StringBuilder merged = new StringBuilder();
        int i = 0, j = 0;

        while (i < word1.length() && j < word2.length()) {
            merged.append(word1.charAt(i++));
            merged.append(word2.charAt(j++));
        }

        // Append remaining characters, if any
        if (i < word1.length()) {
            merged.append(word1.substring(i));
        }
        if (j < word2.length()) {
            merged.append(word2.substring(j));
        }

        return merged.toString();
    }

    public static void main(String[] args) {
        String word1 = "abc";
        String word2 = "pqr";
        System.out.println("Merged String: " + mergeAlternately(word1, word2));

    }
}
```

```
run:
Merged String: apbqcr
BUILD SUCCESSFUL (total time: 0 seconds)
```

5. Write a Java program to find the minimum and maximum values of Integer, Float, and Double using the respective wrapper class constants.

```
package lab.taskk;
public class LabTaskk {
    public static void main(String[] args) {
        System.out.println("Integer Max Valure:"+Integer.MAX_VALUE);
        System.out.println("Integer Min Valure:"+Integer.MIN_VALUE);
        System.out.println("");
        System.out.println("Float Max Valure:"+Float.MAX_VALUE);
        System.out.println("Float Min Valure:"+Float.MIN_VALUE);
        System.out.println("");
        System.out.println("Double Max Valure:"+Double.MAX_VALUE);
        System.out.println("Double Min Valure:"+Double.MIN_VALUE);


    }
}
```

```
run:
Integer Max Valure:2147483647
Integer Min Valure:-2147483648

Float Max Valure:3.4028235E38
Float Min Valure:1.4E-45

Double Max Valure:1.7976931348623157E308
Double Min Valure:4.9E-324
BUILD SUCCESSFUL (total time: 0 seconds)
```

HOME TASKS

1. Write a JAVA program to perform Autoboxing and also implement different methods of wrapper class.

```
package lab.tasks;

public class LabTasks {
    public static void main(String[] args) {
        byte b=10;
        Byte b1=b;
        System.out.println("Byte="+b1);
        short s=13;
        Short s1=s;
        System.out.println("Short="+s1);
        int i=1000;
        Integer i1=i;
        System.out.println("Integer="+i1);
        long l=34L;
        Long l1=l;
        System.out.println("Long="+l1);
        float f=20.5f;
        Float f1=f;
        System.out.println("Float="+f1);
        double d=58.5;
        Double d1=d;
        System.out.println("Double="+d1);
        char c = 'A';
        Character c1=c;
        System.out.println("Character="+c1);
        boolean bool= true;
        Boolean bol=bool;
        System.out.println("Boolean="+bol);
        System.out.println("");
         System.out.println("Integer as Byte: " + i1.byteValue());
        System.out.println("Double as Integer: " + d1.intValue());
        System.out.println("Character in Lowercase: " + Character.toLowerCase(c1));
```

```
run:
Byte=10
Short=13
Integer=1000
Long=34
Float=20.5
Double=58.5
Character=A
Boolean=true

Integer as Byte: -24
Double as Integer: 58
Character in Lowercase: a
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Write a Java program to count the number of even and odd digits in a given integer using Autoboxing and Unboxing.

```java
package lab.tasks;
import java.util.Scanner;
public class LabTasks {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();

        int evenCount = 0;
        int oddCount = 0;

        Integer num = number;

        while (num != 0) {
            int digit = num % 10;

            if (digit % 2 == 0) {
                evenCount++;
            } else {
                oddCount++;
            }

            num = num / 10;
        }

        System.out.println("Number of even digits: " + evenCount);
        System.out.println("Number of odd digits: " + oddCount);
```

```
run:
Enter an integer: 125
Number of even digits: 1
Number of odd digits: 2
BUILD SUCCESSFUL (total time: 8 seconds)
```

3. Write a Java program to find the absolute value, square root, and power of a number using Math class methods, while utilizing Autoboxing and Wrapper classes**.**

```java
package lab.tasks;
import java.util.Scanner;
public class LabTasks {
    public static void main(String[] args) {
     Scanner input = new Scanner(System.in);
     System.out.print("Enter a number: ");
     Double num=input.nextDouble();//Automatic autoboxing
        //Using Math class methods with Autoboxing and Wrapper classes
     System.out.println("Absoloute Value:"+(Math.abs(num))+"\n"+"Sqr root:"+(Math.sqrt(num))+"\nPower:"+(Math.pow(num, 2)));
```

```
run:
Enter a number: 4
Absoloute Value:4.0
Sqr root:2.0
Power:16.0
BUILD SUCCESSFUL (total time: 2 seconds)
```

4. Write a Java program to **reverse only the vowels** in a string.

```java
package lab.tasks;
import java.util.Scanner;
public class LabTasks {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();


        char[] chars = input.toCharArray();
        int left = 0, right = chars.length - 1;

        while (left < right) {
            char leftChar = Character.toLowerCase(chars[left]);
            char rightChar = Character.toLowerCase(chars[right]);

            if (!(leftChar == 'a' || leftChar == 'e' || leftChar == 'i' || leftChar == 'o' || leftChar == 'u')) {
                left++;
            } else if (!(rightChar == 'a' || rightChar == 'e' || rightChar == 'i' || rightChar == 'o' || rightChar == 'u'))
                right--;
            } else {
                char temp = chars[left];
                chars[left] = chars[right];
                chars[right] = temp;
                left++;
                right--;
            }
        }

        System.out.println("String after reversing vowels: " + new String(chars));
```

```
run:
Enter a string: hello world
String after reversing vowels: hollo werld
BUILD SUCCESSFUL (total time: 5 seconds)
```

5. Write a Java program to **find the longest word** in a sentence.

```java
package lab.tasks;
import java.util.Scanner;
public class LabTasks {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String input = scanner.nextLine();

        String longestWord = ""; // Variable to store the longest word
        String[] words = input.split(" "); // Split the input sentence into words

        for (String word : words) {
            // Check if the current word is longer than the longestWord found so far
            if (word.length() > longestWord.length()) {
                longestWord = word; // Update longestWord
            }
        }

        System.out.println("The longest word is: " + longestWord);
```

```
run:
Enter a sentence: programming fundamental is basic
The longest word is: programming
BUILD SUCCESSFUL (total time: 27 seconds)
```