

## 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 initializes 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.

#### INPUT:

```
package lab01;
public class Lab01 {
    public static void main(String[] args) {
        String s1 = new String (original: "Welcome");
        String s2 = new String (original: "To");
        String s3 = new String (original: "DSA");
        String s4 = "In";
        String s5 = "Java";

        System.out.println(x: s1.intern());
        System.out.println(x: s2.intern());
        System.out.println(x: s3.intern());
        System.out.println(x: s4.intern());
        System.out.println(x: s5.intern());
    }
}
```

#### OUTPUT:

```
run:
Welcome
To
DSA
In
Java
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

**INPUT:**

```
public class Lab01 {  
    public static void main(String[] args) {  
        double d1 = 10.5;  
        Double d2 = d1;  
  
        System.out.println(x: d1);  
    }  
}
```

**OUTPUT:**

```
run:  
10.5  
BUILD SUCCESSFUL (total time: 0 seconds)
```

- 3. Write a program that initialize five different strings and perform the following operations.**
- a. Concatenate all five strings.**
  - b. Convert fourth string to uppercase.**
  - c. Find the substring from the concatenated string from 8 to onward**

**INPUT:**

```
public class Lab01 {  
    public static void main(String[] args) {  
        String s1 = "Welcome";  
        String s2 = "To";  
        String s3 = "DSA";  
        String s4 = "In";  
        String s5 = "Java";  
  
        String con = s1 + s2 + s3 + s4 + s5;  
  
        System.out.println(x: con);  
        System.out.println(x: s4.toUpperCase());  
        System.out.println(x: con.substring(beginIndex: 8));  
    }  
}
```

**OUTPUT:**

```
run:
WelcomeToDSAINJava
IN
oDSAINJava
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.

**INPUT:**

```
public class Lab01 {
    public static String mergealternate(String word1, String word2) {
        String merged = "";
        int i = 0;

        while (i < word1.length() || i < word2.length()) {
            if (i < word1.length()) {
                merged += word1.charAt(index: i);
            }
            if (i < word2.length()) {
                merged += word2.charAt(index: i);
            }
            i++;
        }
        return merged;
    }
    public static void main(String[] args) {
        String word1 = "abc";
        String word2 = "vwxyz";
        System.out.println("Merged String: " + mergealternate(word1, word2));
    }
}
```

**OUTPUT:**

```
Merged String: avbwcxyz
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.**

**INPUT:**

```
public class Lab01{  
    public static void main(String[] args) {  
        System.out.println("Integer Minimum Value: " + Integer.MIN_VALUE);  
        System.out.println("Integer Maximum Value: " + Integer.MAX_VALUE);  
  
        System.out.println("Float Minimum Value: " + Float.MIN_VALUE);  
        System.out.println("Float Maximum Value: " + Float.MAX_VALUE);  
  
        System.out.println("Double Minimum Value: " + Double.MIN_VALUE);  
        System.out.println("Double Maximum Value: " + Double.MAX_VALUE);  
    }  
}
```

**OUTPUT:**

```
Integer Minimum Value: -2147483648  
Integer Maximum Value: 2147483647  
Float Minimum Value: 1.4E-45  
Float Maximum Value: 3.4028235E38  
Double Minimum Value: 4.9E-324  
Double Maximum Value: 1.7976931348623157E308  
BUILD SUCCESSFUL (total time: 0 seconds)
```

## HOME TASK

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

### INPUT:

```
public class ReverseVowel {  
    public static String reverseVowels(String str) {  
        char[] chars = str.toCharArray();  
        int left = 0, right = chars.length - 1;  
        String vowels = "AEIOUaeiou";  
  
        while (left < right) {  
            while (left < right && vowels.indexOf(chars[left]) == -1) left++;  
            while (left < right && vowels.indexOf(chars[right]) == -1) right--;  
  
            char temp = chars[left];  
            chars[left] = chars[right];  
            chars[right] = temp;  
            left++;  
            right--;  
        }  
        return new String(chars);  
    }  
  
    public static void main(String[] args) {  
        String input = "hello";  
        System.out.println("Reversed Vowels: " + reverseVowels(input));  
    }  
}
```

### OUTPUT:

```
Reversed Vowels: holle
```

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

### INPUT:

```
import java.util.Scanner;
public class Lab01 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print(s: "Enter an integer: ");
        int inputNumber = scanner.nextInt();
        Integer number = inputNumber;

        int evenCount = 0;
        int oddCount = 0;

        int num = number;

        while (num != 0) {
            int digit = num % 10;
            if (digit % 2 == 0) {
                evenCount++;
            } else {
                oddCount++;
            }
            num /= 10;
        }
        System.out.println("Number of even digits: " + evenCount);
        System.out.println("Number of odd digits: " + oddCount);
        scanner.close();
    }
}
```

### OUTPUT:

```
Enter an integer: 5468
Number of even digits: 3
Number of odd digits: 1
BUILD SUCCESSFUL (total time: 5 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.**

**INPUT:**

```
public class Lab01 {  
    public static void main(String[] args) {  
        double num = -25.0;  
        double exponent = 2.0;  
  
        Double number = num;  
        Double power = exponent;  
  
        Double absoluteValue = Math.abs(a: number);  
        Double squareRoot = Math.sqrt(a: Math.abs(a: number));  
        Double powerResult = Math.pow(a: number, b: power);  
  
        System.out.println("Absolute Value of " + number + " : " + absoluteValue);  
        System.out.println("Square Root of Absolute Value " + number + " : " + squareRoot);  
        System.out.println("Power of " + number + " raised to " + power + " : " + powerResult);  
    }  
}
```

**OUTPUT:**

```
Absolute Value of -25.0 : 25.0  
Square Root of Absolute Value -25.0 : 5.0  
Power of -25.0 raised to 2.0 : 625.0  
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

```
public class Lab01 {  
    public static void main(String[] args) {  
        int primitiveInt = 10;  
        Integer wrappedInt = primitiveInt;  
        System.out.println("Autoboxed Integer: " + wrappedInt);  
  
        double primitiveDouble = 20.5;  
        Double wrappedDouble = primitiveDouble;  
        System.out.println("Autoboxed Double: " + wrappedDouble);  
  
        float primitiveFloat = 15.5f;  
        Float wrappedFloat = primitiveFloat;  
        System.out.println("Autoboxed Float: " + wrappedFloat);  
  
        System.out.println(x: "Integer Methods:");  
        System.out.println("Integer to String: " + Integer.toString(i: wrappedInt));  
        System.out.println("String to Integer: " + Integer.parseInt(s: "100"));  
  
        System.out.println(x: "\nDouble Methods:");  
        System.out.println("Double to String: " + Double.toString(d: wrappedDouble));  
        System.out.println("String to Double: " + Double.parseDouble(s: "30.75"));  
  
        System.out.println(x: "\nFloat Methods:");  
        System.out.println("Float to String: " + Float.toString(f: wrappedFloat));  
        System.out.println("String to Float: " + Float.parseFloat(s: "25.5"));  
    }  
}
```

**OUTPUT:**

```
Autoboxed Integer: 10  
Autoboxed Double: 20.5  
Autoboxed Float: 15.5  
Integer Methods:  
Integer to String: 10  
String to Integer: 100  
  
Double Methods:  
Double to String: 20.5  
String to Double: 30.75  
  
Float Methods:  
Float to String: 15.5  
String to Float: 25.5
```



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

```
public class Lab01 {  
    public static void main(String[] args) {  
        String sentence = "Java is a programming language";  
        String longestWord = findLongestWord(sentence);  
        System.out.println("The longest word is: " + longestWord);  
    }  
    public static String findLongestWord(String sentence) {  
        String[] words = sentence.split(regex: " ");  
        String longestWord = "";  
  
        for (String word : words) {  
            if (word.length() > longestWord.length()) {  
                longestWord = word;  
            }  
        }  
        return longestWord;  
    }  
}
```

**OUTPUT:**

```
The longest word is: programming  
BUILD SUCCESSFUL (total time: 0 seconds)
```