**LAB#01**

**Lab Task#01:**

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

a) string literals

b) new keyword

use intern method and show string immutability.

**Code: Output:**

public class Lab1 {

public static void main(String[] args) {

A screenshot of a computer program

Description automatically generated String str1 = "Hello";

System.out.println("str1: " + str1);

String str2 = new String("Hello");

System.out.println("str2: " + str2);

String str3 = "Hello".intern();

System.out.println("str3: " + str3);

String str4 = "World";

System.out.println("str4: " + str4);

String str5 = new String("Java");

System.out.println("str5: " + str5);

str1 = str1 + " World";

System.out.println("str1 after modification: " + str1);

System.out.println("Original str1: " + "Hello");

}

}

**Lab Task#02:**

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

**Code: Output:**

public class Lab1 {

public static void main(String[] args) {

A screenshot of a computer

Description automatically generated double n = 9.00;

System.out.println("Value of n: " + n);

Double m = n;

System.out.println("value of m: " + m);

}

}

**Lab Task#03:**

3. Write a program that initializes five different strings and performs the following operations.

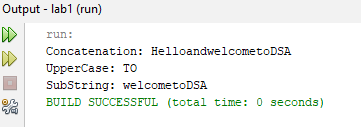
a. Concatenate all five stings.

b. Convert the fourth string to uppercase.

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

**Code: Output:**

public class Lab1 {

 public static void main(String[] args) {

String str1= "Hello";

String str2= "and";

String str3= "welcome";

String str4= "to";

String str5= "DSA";

String str6= str1+str2+str3+str4+str5;

System.out.println("Concatenation: "+ str6);

System.out.println("UpperCase: "+ str4.toUpperCase());

System.out.println("SubString: "+ str6.substring(8));

}

}

**Lab Task#04:**

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.

**Code: Output:**

public class MergeStrings {

public static void main(String[] args) {

String word1 = "abc";

A screenshot of a computer

Description automatically generated String word2 = "pqr";

mergeStrings(word1, word2);

}

public static void mergeStrings(String word1, String word2) {

String merged = "";

int length1 = word1.length();

int length2 = word2.length();

for (int i = 0; i < length1 || i < length2; i++) {

if (i < length1) {

merged += word1.charAt(i);

}

if (i < length2) {

merged += word2.charAt(i);

}

}

System.out.println("Merged String: " + merged);

}

}

**Lab Task#05:**

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

**Code:**

public class Lab1 {

public static void main(String[] args) {

System.out.println("Integer Min Value: " + Integer.MIN\_VALUE);

System.out.println("Integer Max Value: " + Integer.MAX\_VALUE);

System.out.println("Float Min Value: " + Float.MIN\_VALUE);

System.out.println("Float Max Value: " + Float.MAX\_VALUE);

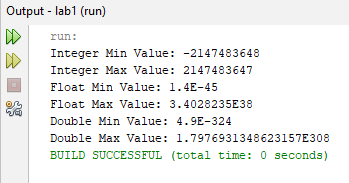
System.out.println("Double Min Value: " + Double.MIN\_VALUE);

System.out.println("Double Max Value: " + Double.MAX\_VALUE);

}

}

**Output:**



**Home Task#01:**

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

**Code:**

public class HomeTask1 {

public static void main(String[] args) {

//Autoboxing

Integer wrappedInt = 10;

int unboxedInt = wrappedInt;

System.out.println("Unboxed int: " + unboxedInt);

Double wrappedDouble = 5.67;

double unboxedDouble = wrappedDouble;

System.out.println("Unboxed double: " + unboxedDouble);

Boolean wrappedBool = true;

boolean unboxedBool = wrappedBool;

System.out.println("Unboxed boolean: " + unboxedBool);

//different methods for wrapper class

int parsedInt = Integer.parseInt("25");

double parsedDouble = Double.parseDouble("45.89");

boolean parsedBool = Boolean.parseBoolean("true");

System.out.println("Parsed int: " + parsedInt);

System.out.println("Parsed double: " + parsedDouble);

System.out.println("Parsed boolean: " + parsedBool);

System.out.println("Int to String: " + Integer.toString(parsedInt));

System.out.println("Double to String: " + Double.toString(parsedDouble));

System.out.println("Max Integer: " + Integer.MAX\_VALUE);

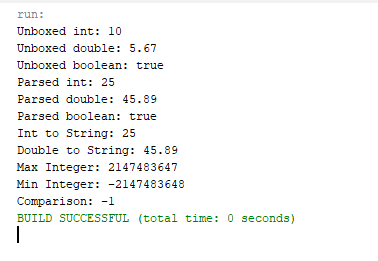
System.out.println("Min Integer: " + Integer.MIN\_VALUE);

System.out.println("Comparison: " + Integer.compare(100, 200));

}

}

**Output:**



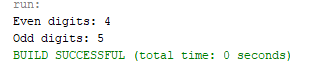
**Home Task#02:**

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

**Code: Output:**

public class HomeTask2 {

public static void main(String[] args) {

 int number = 123456789;

Integer wrappedNumber = number;

int evenCount = 0, oddCount = 0;

while (wrappedNumber > 0) {

int digit = wrappedNumber % 10;

if (digit % 2 == 0) {

evenCount++;

} else {

oddCount++;

}

wrappedNumber /= 10;

}

System.out.println("Even digits: " + evenCount);

System.out.println("Odd digits: " + oddCount);

}

}

**Home Task#03:**

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.

**Code:**

public class HomeTask2 {

public static void main(String[] args) {

Integer wrappedNumber = -16;

int absoluteValue = Math.abs(wrappedNumber);

System.out.println("Absolute Value: " + absoluteValue);

Double wrappedSquareRoot = Math.sqrt(Math.abs(wrappedNumber));

System.out.println("Square Root: " + wrappedSquareRoot);

Integer exponent = 2;

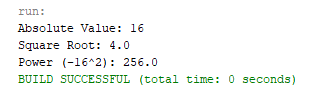
double power = Math.pow(wrappedNumber.doubleValue(), exponent);

System.out.println("Power (" + wrappedNumber + "^" + exponent + "): " + power);

}

}

**Output:**



**Home Task#04:**

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

**Code: Output:**

public class HomeTask2 {

public static void main(String[] args) {

String input = "hello world";

System.out.println("Reversed Vowels String: " + reverseVowels(input));

}

public static String reverseVowels(String s) {

String vowels = "aeiouAEIOU";

String reversedVowels = "";

A white background with black text

Description automatically generated for (int i = 0; i < s.length(); i++) {

char c = s.charAt(i);

if (vowels.indexOf(c) != -1) {

reversedVowels = c + reversedVowels;

}

}

String result = "";

int index = 0;

for (int i = 0; i < s.length(); i++) {

char c = s.charAt(i);

if (vowels.indexOf(c) != -1) {

result += reversedVowels.charAt(index++);

} else {

result += c;

}

}

return result;

}

}

**Home Task#05:**

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

**Code: Output:**

public class HomeTask2 {

public static void main(String[] args) {

String sentence = "This is a simple Java program to find the longest word.";

String longestWord = findLongestWord(sentence);

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

}

public static String findLongestWord(String sentence) {

String[] words = sentence.split(" ");

A white background with black text

Description automatically generated String longestWord = "";

for (String word : words) {

word = word.replaceAll("[^a-zA-Z]", "");

if (word.length() > longestWord.length()) {

longestWord = word;

}

}

return longestWord;

}

}