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

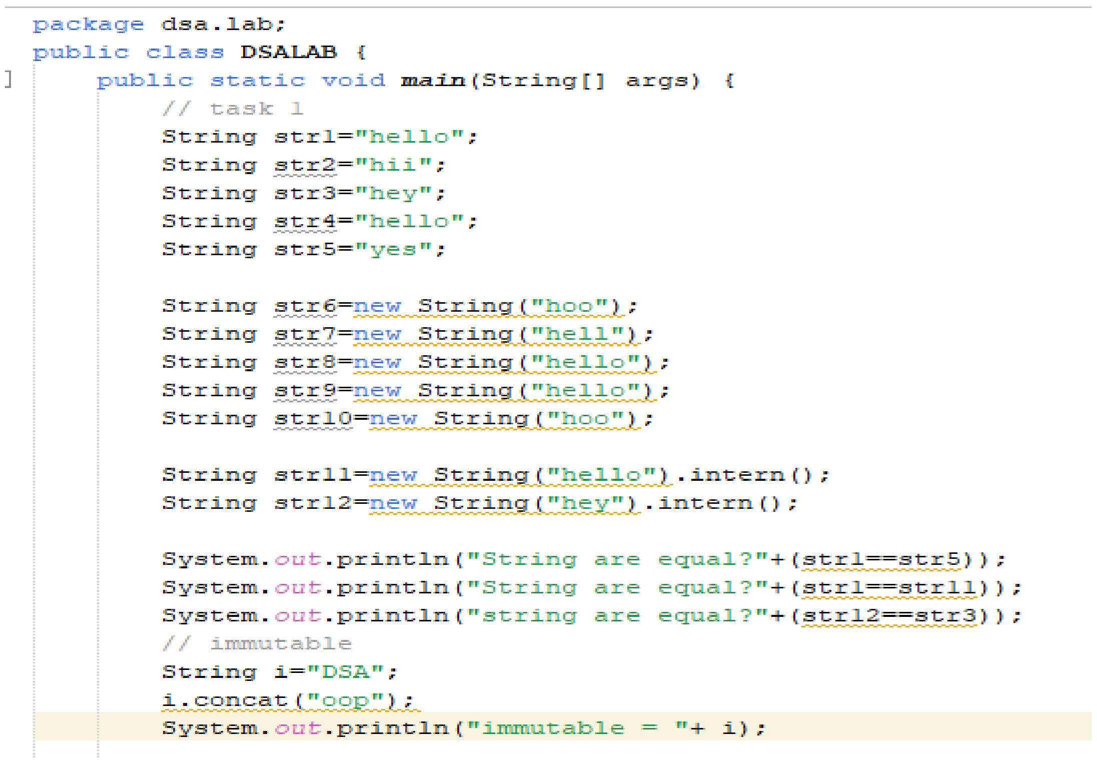
1. Write a program that initialize five different strings using all the above mentioned ways, i.e.,

   a) string literals

   b) new keyword

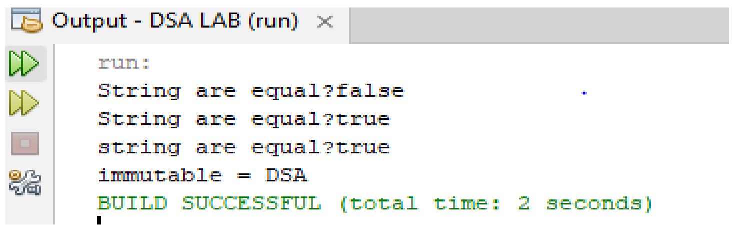
also use intern method and show string immutability.

CODE:



```
package dsa.lab;
public class DSALAB {
    public static void main(String[] args) {
        // task 1
        String str1="hello";
        String str2="hii";
        String str3="hey";
        String str4="hello";
        String str5="yes";

        String str6=new String("hoo");
        String str7=new String("hell");
        String str8=new String("hello");
        String str9=new String("hello");
        String str10=new String("hoo");

        String str11=new String("hello").intern();
        String str12=new String("hey").intern();

        System.out.println("String are equal?"+(str1==str5));
        System.out.println("String are equal?"+(str1==str11));
        System.out.println("string are equal?"+(str12==str3));
        // immutable
        String i="DSA";
        i.concat("oop");
        System.out.println("immutable = "+ i);
```

OUTPUT:

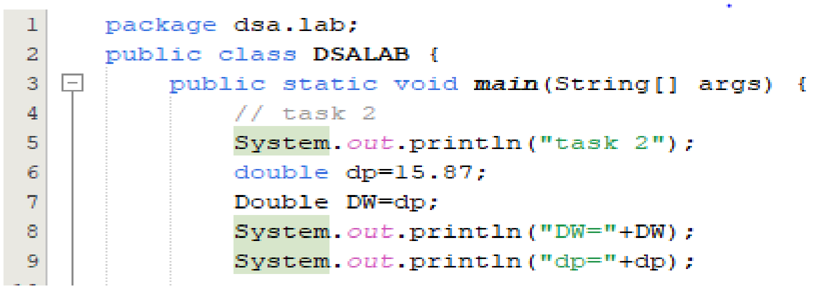


Output - DSA LAB (run)

```
run:
String are equal?false
String are equal?true
string are equal?true
immutable = DSA
BUILD SUCCESSFUL (total time: 2 seconds)
```

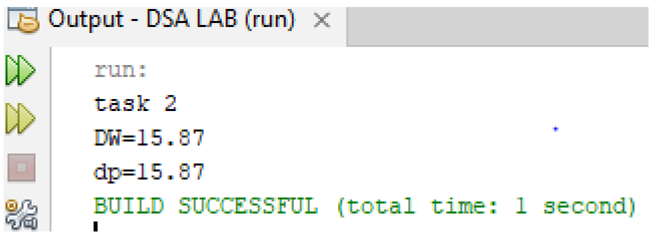
2. Write a program to convert primitive data type Double into its respective wrapper object.

CODE:



```java
package dsa.lab;
public class DSALAB {
    public static void main(String[] args) {
        // task 2
        System.out.println("task 2");
        double dp=15.87;
        Double DW=dp;
        System.out.println("DW="+DW);
        System.out.println("dp="+dp);
```

OUTPUT;

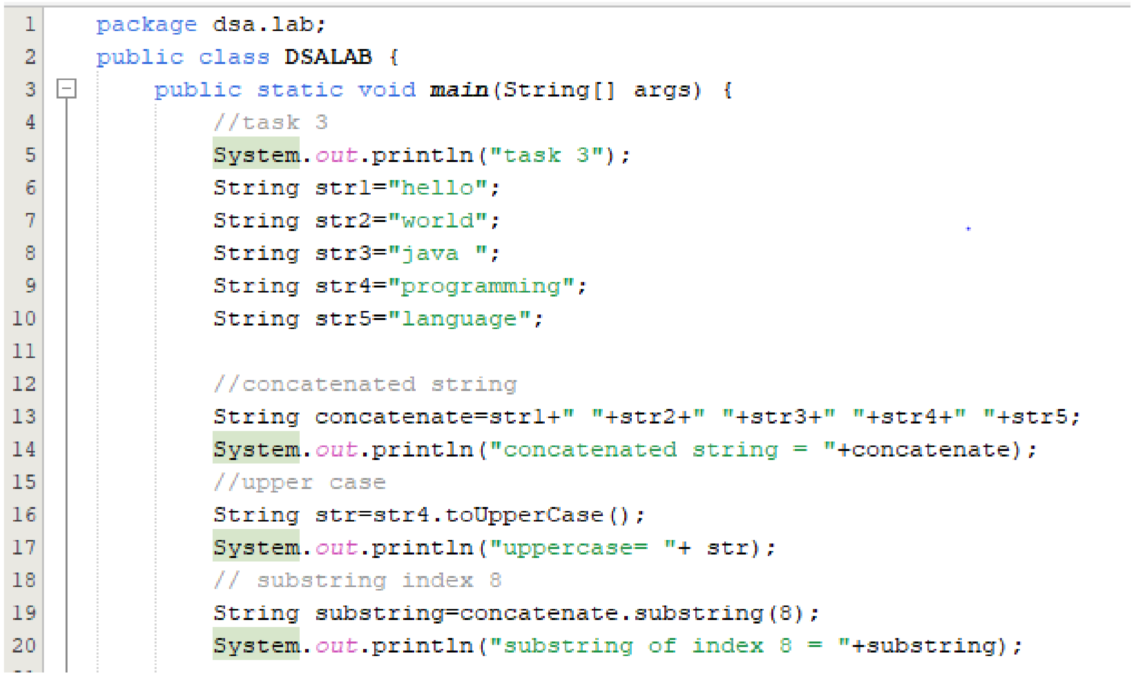


```
run:
task 2
DW=15.87
dp=15.87
BUILD SUCCESSFUL (total time: 1 second)
```

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

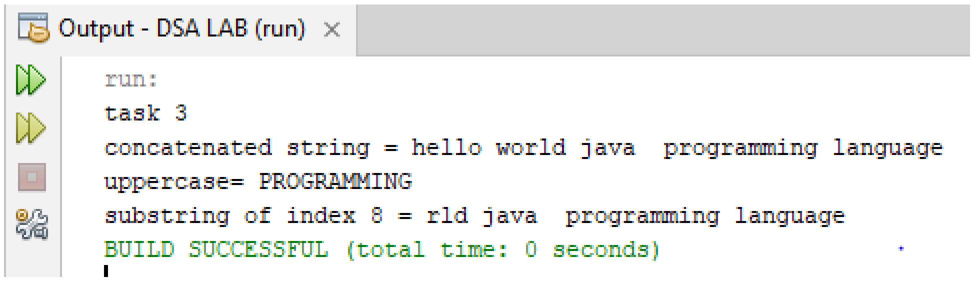
b) Convert fourth string to uppercase.
c) Find the substring from the concatenated string from 8 to onward

CODE:



```java
package dsa.lab;
public class DSALAB {
    public static void main(String[] args) {
        //task 3
        System.out.println("task 3");
        String str1="hello";
        String str2="world";
        String str3="java ";
        String str4="programming";
        String str5="language";

        //concatenated string
        String concatenate=str1+" "+str2+" "+str3+" "+str4+" "+str5;
        System.out.println("concatenated string = "+concatenate);
        //upper case
        String str=str4.toUpperCase();
        System.out.println("uppercase= "+ str);
        // substring index 8
        String substring=concatenate.substring(8);
        System.out.println("substring of index 8 = "+substring);
```

OUTPUT:



```
run:
task 3
concatenated string = hello world java  programming language
uppercase= PROGRAMMING
substring of index 8 = rld java  programming language
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"

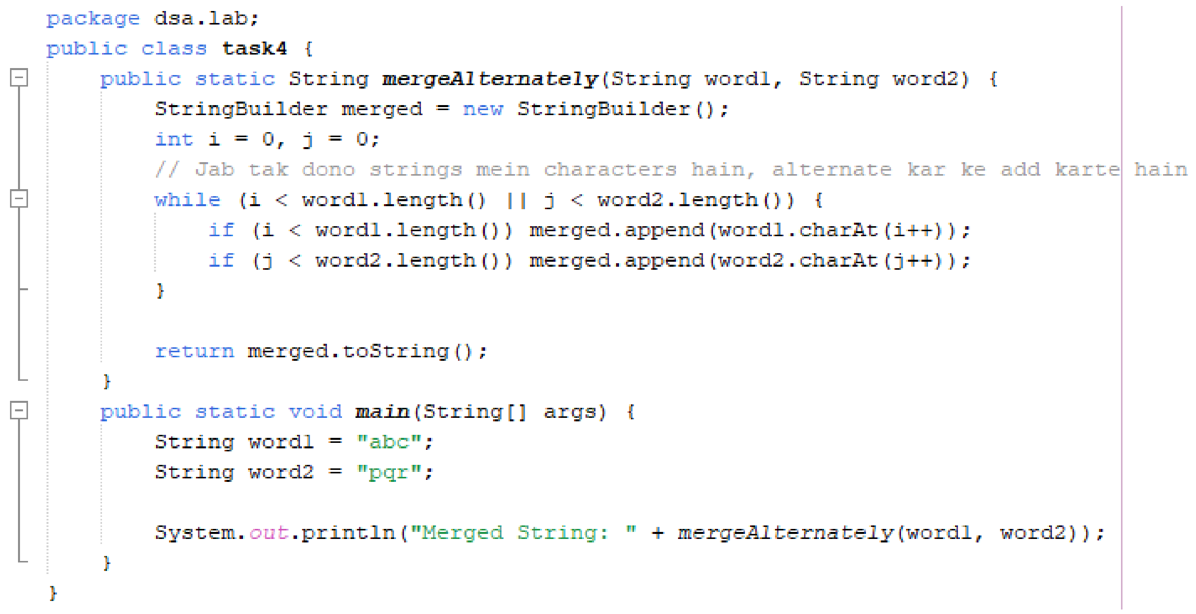
Explanation: The merged string will be merged as so:

word1: a b c word2:

p q r

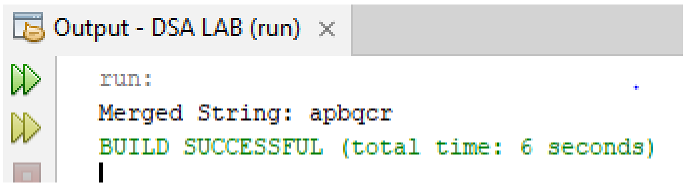
merged: a p b q c r

CODE:



```
package dsa.lab;
public class task4 {
    public static String mergeAlternately(String word1, String word2) {
        StringBuilder merged = new StringBuilder();
        int i = 0, j = 0;
        // Jab tak dono strings mein characters hain, alternate kar ke add karte hain
        while (i < word1.length() || j < word2.length()) {
            if (i < word1.length()) merged.append(word1.charAt(i++));
            if (j < word2.length()) merged.append(word2.charAt(j++));
        }

        return merged.toString();
    }
    public static void main(String[] args) {
        String word1 = "abc";
        String word2 = "pqr";

        System.out.println("Merged String: " + mergeAlternately(word1, word2));
    }
}
```

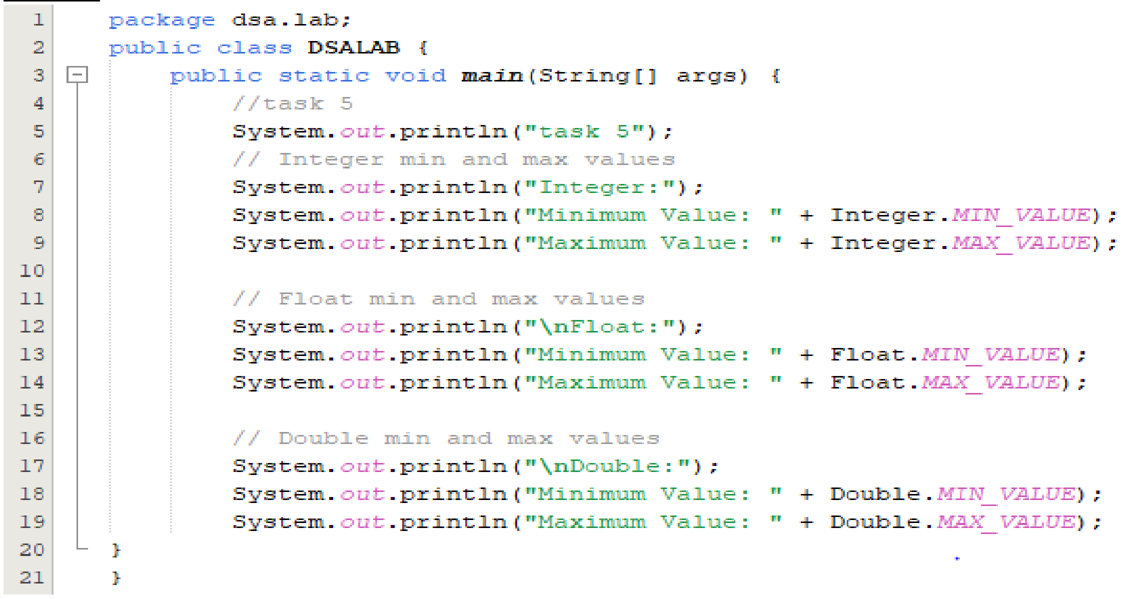
OUTPUT:



```
run:
Merged String: apbqcr
BUILD SUCCESSFUL (total time: 6 seconds)
```

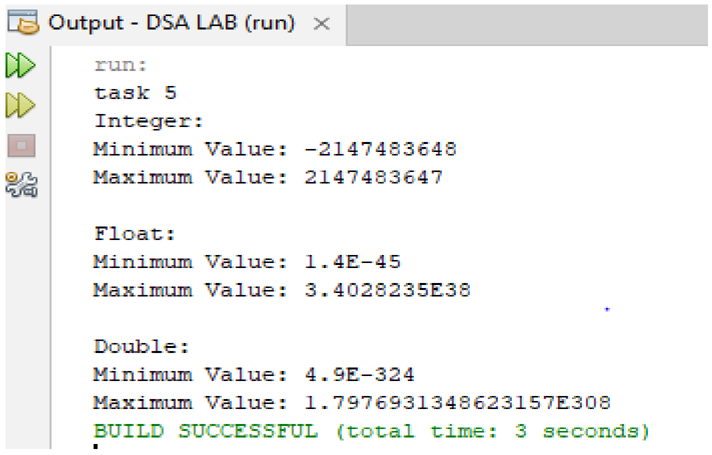
5. Write a Java program to find the minimum and maximum values of Integer, Float, and Double using the respective wrapper class constants.

CODE :



```
package dsa.lab;
public class DSALAB {
    public static void main(String[] args) {
        //task 5
        System.out.println("task 5");
        // Integer min and max values
        System.out.println("Integer:");
        System.out.println("Minimum Value: " + Integer.MIN_VALUE);
        System.out.println("Maximum Value: " + Integer.MAX_VALUE);

        // Float min and max values
        System.out.println("\nFloat:");
        System.out.println("Minimum Value: " + Float.MIN_VALUE);
        System.out.println("Maximum Value: " + Float.MAX_VALUE);

        // Double min and max values
        System.out.println("\nDouble:");
        System.out.println("Minimum Value: " + Double.MIN_VALUE);
        System.out.println("Maximum Value: " + Double.MAX_VALUE);
}
}
```

OUTPUT



Output - DSA LAB (run)

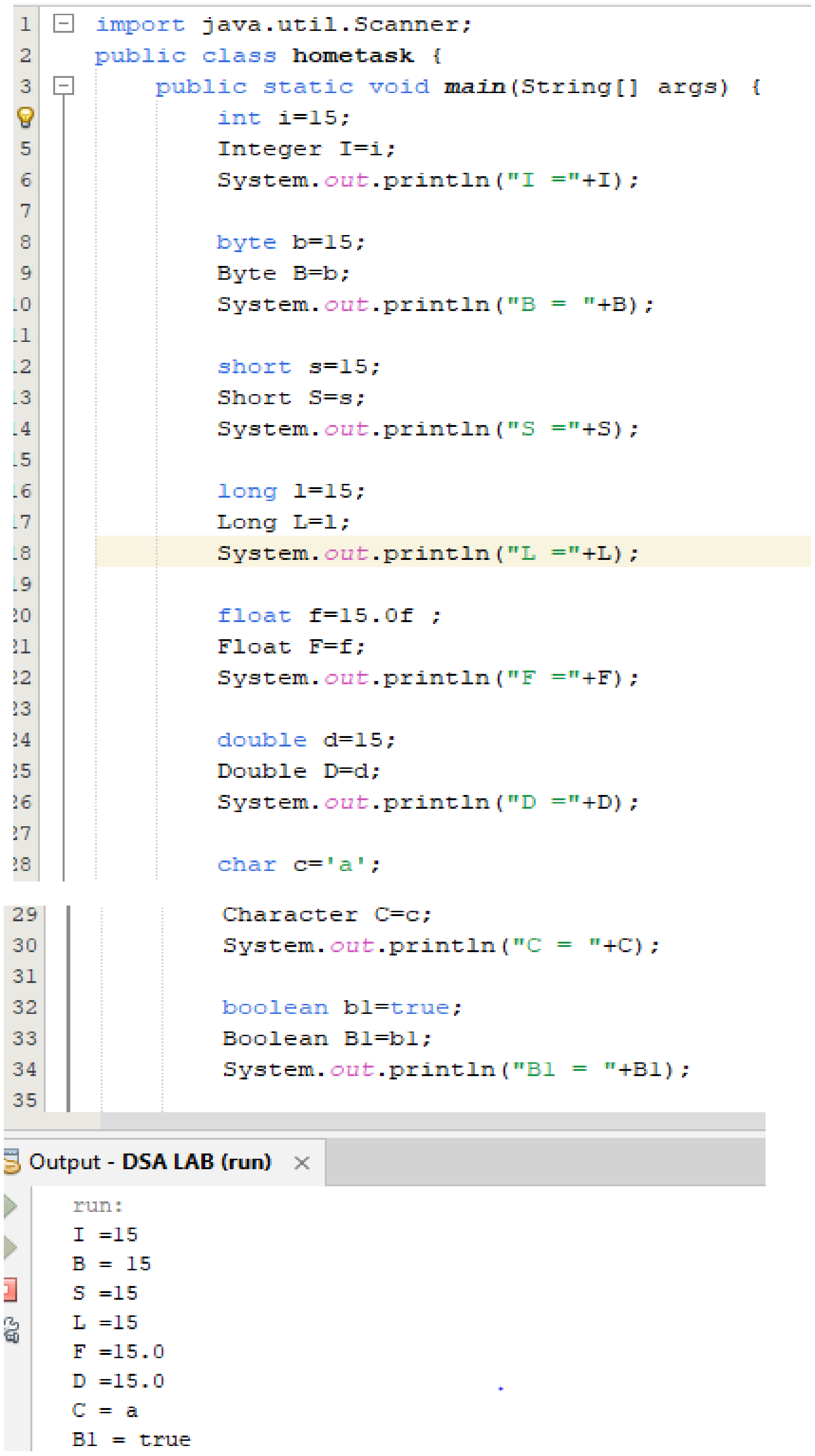
```
run:
task 5
Integer:
Minimum Value: -2147483648
Maximum Value: 2147483647

Float:
Minimum Value: 1.4E-45
Maximum Value: 3.4028235E38

Double:
Minimum Value: 4.9E-324
Maximum Value: 1.7976931348623157E308
BUILD SUCCESSFUL (total time: 3 seconds)
```

.

HOME TASKS

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

CODE:



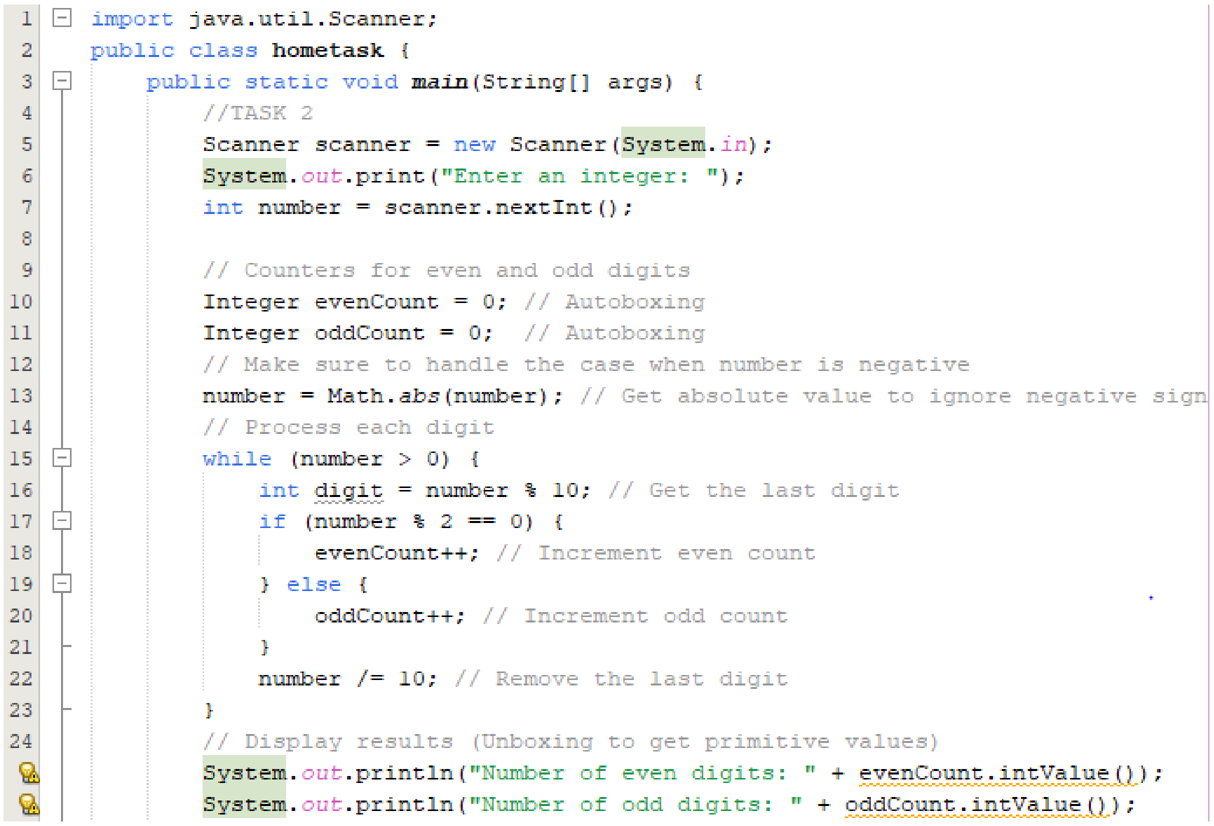
```
import java.util.Scanner;
public class hometask {
    public static void main(String[] args) {
        int i=15;
        Integer I=i;
        System.out.println("I ="+I);

        byte b=15;
        Byte B=b;
        System.out.println("B = "+B);

        short s=15;
        Short S=s;
        System.out.println("S ="+S);

        long l=15;
        Long L=l;
        System.out.println("L ="+L);

        float f=15.0f ;
        Float F=f;
        System.out.println("F ="+F);

        double d=15;
        Double D=d;
        System.out.println("D ="+D);

        char c='a';
        Character C=c;
        System.out.println("C = "+C);

        boolean bl=true;
        Boolean Bl=bl;
        System.out.println("Bl = "+Bl);

```

Output - DSA LAB (run)

```
run:
I =15
B = 15
S =15
L =15
F =15.0
D =15.0
C = a
Bl = true
```

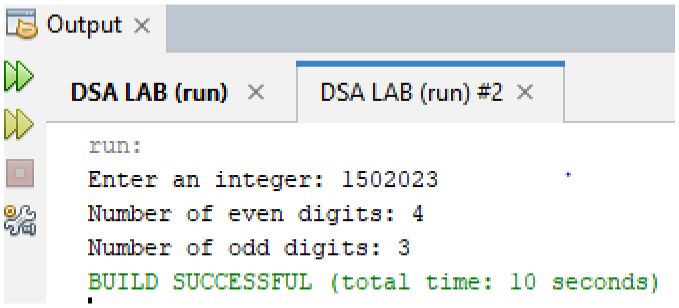
2. Write a Java program to count the number of even and odd digits in a given integer using Autoboxing and Unboxing.

CODE:



```java
import java.util.Scanner;
public class hometask {
    public static void main(String[] args) {
        //TASK 2
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();

        // Counters for even and odd digits
        Integer evenCount = 0; // Autoboxing
        Integer oddCount = 0;  // Autoboxing
        // Make sure to handle the case when number is negative
        number = Math.abs(number); // Get absolute value to ignore negative sign
        // Process each digit
        while (number > 0) {
            int digit = number % 10; // Get the last digit
            if (number % 2 == 0) {
                evenCount++; // Increment even count
            } else {
                oddCount++; // Increment odd count
            }
            number /= 10; // Remove the last digit
        }
        // Display results (Unboxing to get primitive values)
        System.out.println("Number of even digits: " + evenCount.intValue());
        System.out.println("Number of odd digits: " + oddCount.intValue());
```

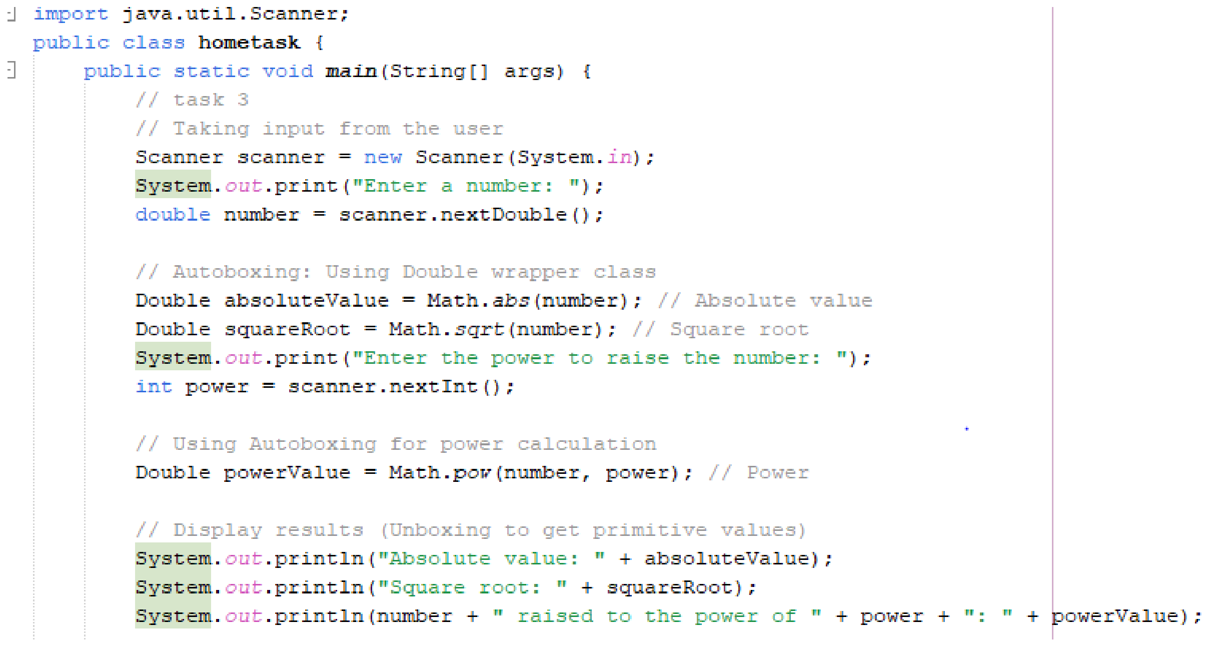
OUTPUT:



```
run:
Enter an integer: 1502023
Number of even digits: 4
Number of odd digits: 3
BUILD SUCCESSFUL (total time: 10 seconds)
```

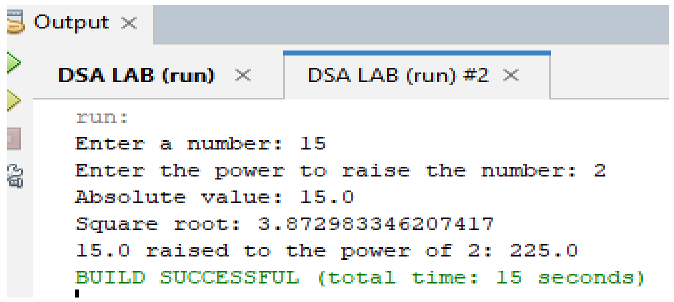
3. Write a Java program to find the absolute value, square root, and power of a number using Math class methods, while utilizing Autoboxing and Wrapper classes.

CODE:



```java
import java.util.Scanner;
public class hometask {
    public static void main(String[] args) {
        // task 3
        // Taking input from the user
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        double number = scanner.nextDouble();

        // Autoboxing: Using Double wrapper class
        Double absoluteValue = Math.abs(number); // Absolute value
        Double squareRoot = Math.sqrt(number); // Square root
        System.out.print("Enter the power to raise the number: ");
        int power = scanner.nextInt();

        // Using Autoboxing for power calculation
        Double powerValue = Math.pow(number, power); // Power

        // Display results (Unboxing to get primitive values)
        System.out.println("Absolute value: " + absoluteValue);
        System.out.println("Square root: " + squareRoot);
        System.out.println(number + " raised to the power of " + power + ": " + powerValue);
```

OUTPUT:



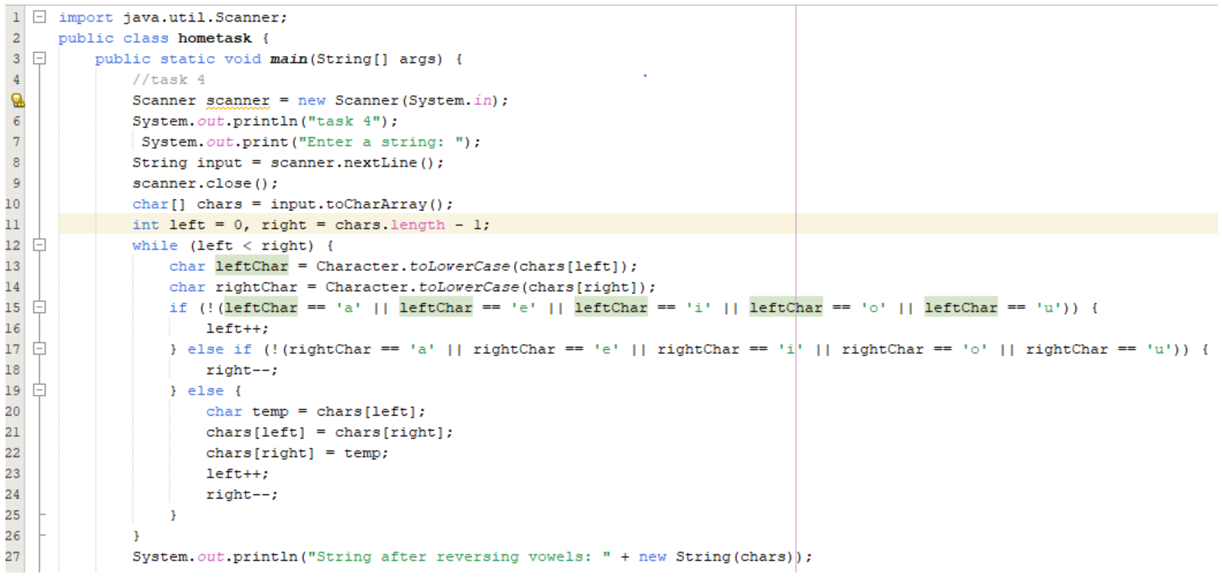
Output ×

DSA LAB (run) × DSA LAB (run) #2 ×

```
run:
Enter a number: 15
Enter the power to raise the number: 2
Absolute value: 15.0
Square root: 3.872983346207417
15.0 raised to the power of 2: 225.0
BUILD SUCCESSFUL (total time: 15 seconds)
```

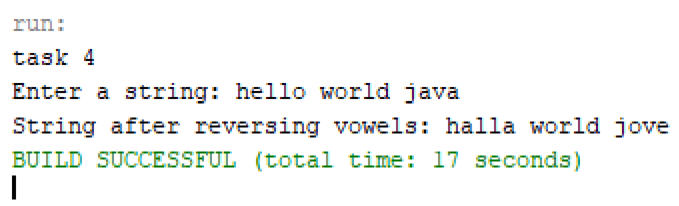
4. Write a Java program to **reverse only the vowels** in a string.

CODE:



```java
import java.util.Scanner;
public class hometask {
    public static void main(String[] args) {
        //task 4
        Scanner scanner = new Scanner(System.in);
        System.out.println("task 4");
         System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        scanner.close();
        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));
```

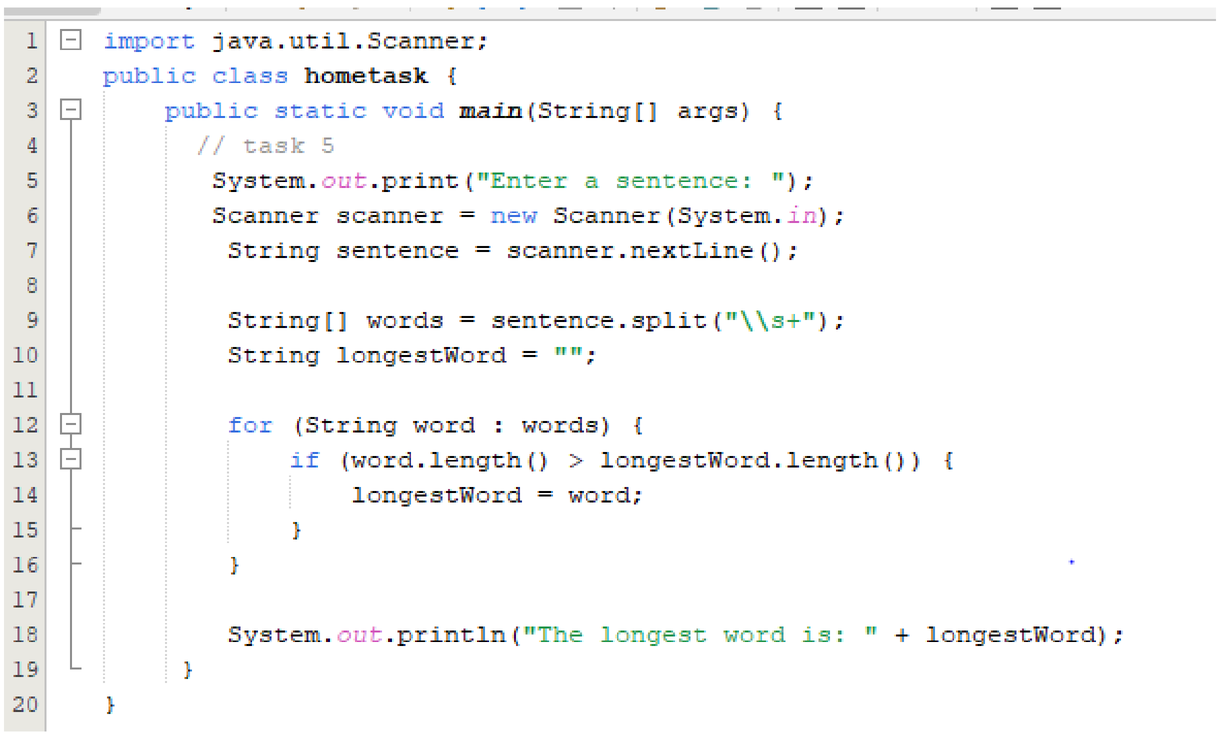
OUTPUT:



```
run:
task 4
Enter a string: hello world java
String after reversing vowels: halla world jove
BUILD SUCCESSFUL (total time: 17 seconds)
```

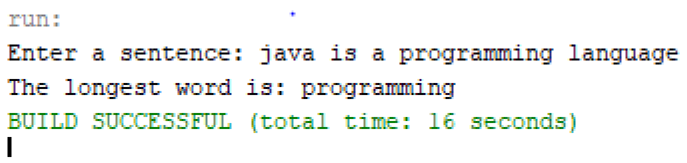
5. Write a Java program to **find the longest word** in a sentence.

CODE:



```java
import java.util.Scanner;
public class hometask {
    public static void main(String[] args) {
      // task 5
       System.out.print("Enter a sentence: ");
       Scanner scanner = new Scanner(System.in);
        String sentence = scanner.nextLine();

        String[] words = sentence.split("\\s+");
        String longestWord = "";

        for (String word : words) {
            if (word.length() > longestWord.length()) {
                longestWord = word;
            }
        }

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

OUTPUT:



```
run:
Enter a sentence: java is a programming language
The longest word is: programming
BUILD SUCCESSFUL (total time: 16 seconds)
```