1.     Method Overloading: Write a class Calculator with overloaded methods add(). Implement add() methods that take:

     - Two integers

     - Two double values

     - Three integers

     - A variable number of integers

Program :

package Lab\_4;

public class Calculator {

// Method to add two integers

public int add(int a, int b) {

return a + b;

}

// Method to add two double values

public double add(double a, double b) {

return a + b;

}

// Method to add three integers

public int add(int a, int b, int c) {

return a + b + c;

}

// Method to add a variable number of integers using varargs

public int add(int... numbers) {

int sum = 0;

for (int num : numbers) {

sum += num;

}

return sum;

}

public static void main(String[] args) {

Calculator calc = new Calculator();

// Adding two integers

int sum1 = calc.add(5, 10);

System.***out***.println("Sum of 5 and 10 is: " + sum1);

// Adding two double values

double sum2 = calc.add(2.5, 3.5);

System.***out***.println("Sum of 2.5 and 3.5 is: " + sum2);

// Adding three integers

int sum3 = calc.add(3, 7, 11);

System.***out***.println("Sum of 3, 7, and 11 is: " + sum3);

// Adding variable number of integers

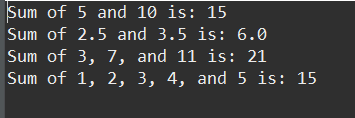
int sum4 = calc.add(1, 2, 3, 4, 5);

System.***out***.println("Sum of 1, 2, 3, 4, and 5 is: " + sum4);

}

}

Output :-



2. Super Keyword: Create a class Person with a constructor that accepts and sets name and age.

   - Create a subclass Student that adds a grade property and initializes name and age using the super keyword in its constructor.

   - Demonstrate the creation of Student objects and the usage of super to call the parent class constructor.

**Program :**

package Lab\_4;

//Person class

class Person {

private String name;

private int age;

// Constructor of Person class

public Person(String name, int age) {

this.name = name;

this.age = age;

}

// Getters and setters

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

}

//Student class, subclass of Person

class Student extends Person {

private int grade;

// Constructor of Student class using super to call Person's constructor

public Student(String name, int age, int grade) {

super(name, age); // Call to superclass constructor

this.grade = grade;

}

// Getter and setter for grade

public int getGrade() {

return grade;

}

public void setGrade(int grade) {

this.grade = grade;

}

// Main class to demonstrate the creation of Student objects

public static void main(String[] args) {

// Create a Student object using constructor with super keyword

Student student1 = new Student("Alice", 20, 85);

System.***out***.println("Student Name: " + student1.getName());

System.***out***.println("Student Age: " + student1.getAge());

System.***out***.println("Student Grade: " + student1.getGrade());

}

}

**Output :**

A black background with white text

Description automatically generated

3. Super Keyword: Create a base class Shape with a method draw() that prints "Drawing Shape".

   - Create a subclass Circle that overrides draw() to print "Drawing Circle".

   - Inside the draw() method of Circle, call the draw() method of the Shape class using super.draw().

   - Write a main method to demonstrate calling draw() on a Circle object.

Program:

package Lab\_4

// Shape class (base class)

class Shape {

// Method to draw shape

public void draw() {

System.out.println("Drawing Shape");

}

}

// Circle class (subclass of Shape)

class Circle extends Shape {

// Override draw method to draw circle

@Override

public void draw() {

super.draw(); // Call draw method of superclass Shape

System.out.println("Drawing Circle");

}

}

// Main class to demonstrate calling draw() on Circle object

public class Main {

public static void main(String[] args) {

// Create a Circle object

Circle circle = new Circle();

// Call draw method on Circle object

circle.draw();

}

}

Output :

A black background with white text

Description automatically generated

4. Write a Java Program to count the number of words in a String without using the Predefined method?

Program :

package Lab\_4;

import java.util.StringTokenizer;

public class WordCounts {

public static void main(String[] args) {

String text = "Hello World! this is my random string ";

int wordCount = *countWordsUsingTokenizer*(text);

System.***out***.println("Number of words in the string: " + wordCount);

}

public static int countWordsUsingTokenizer(String text) {

StringTokenizer tokenizer = new StringTokenizer(text);

return tokenizer.countTokens();

}

}

Output :



5. Write a Java Program to remove all white spaces from a String?

Program :

package Lab\_4;

public class RemoveWhiteSpaces {

public static void main(String[] args) {

String input = " Hello World! This is a test string ";

String output = *removeSpacesUsingReplaceAll*(input);

System.***out***.println("Original string: [" + input + "]");

System.***out***.println("String after removing spaces: [" + output + "]");

}

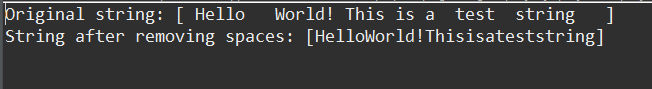
public static String removeSpacesUsingReplaceAll(String input) {

return input.replaceAll(" ", "");

}

}

Output :



6. WAP to find occurrence of given in the given string.

Program :

package Lab\_4

import java.util.StringTokenizer;

public class OccurrenceFinder {

public static void main(String[] args) {

String mainString = "hello , hello hello testing";

String substring = "hello";

int count = *countOccurrences*(mainString, substring);

System.***out***.println("Occurrences of '" + substring + "' in '" + mainString + "': " + count);

}

public static int countOccurrences(String mainString, String substring) {

StringTokenizer tokenizer = new StringTokenizer(mainString, substring);

// Subtract 1 because tokenizer includes

// empty token at the end if substring is at end

return tokenizer.countTokens() - 1;

}

}

Output :



7. Write a java class to implement any 10 string methods:

● replace ● contains ● replaceAll ● indexOf ● substring ● Equals ● lastIndexOf ● startsWith

● endsWith ● EqualsIgnoreCase ● toLowerCase ● toUpperCase ● isEmpty ● Length ● split

Program :

package Lab\_4;

public class StringMethodsExample {

public static void main(String[] args) {

// Sample string for testing

String str = "Hello, World!";

// Example usage of each method

System.***out***.println("Original string: " + str);

System.***out***.println("1. replace 'l' with 'X': " + str.replace('l', 'X'));

System.***out***.println("2. contains 'World': " + str.contains("World"));

System.***out***.println("3. replaceAll 'l' with 'X': " + str.replaceAll("l", "X"));

System.***out***.println("4. indexOf 'o': " + str.indexOf('o'));

System.***out***.println("5. substring from index 7: " + str.substring(7));

System.***out***.println("6. Equals 'Hello, World!': " + str.equals("Hello, World!"));

System.***out***.println("7. lastIndexOf 'l': " + str.lastIndexOf('l'));

System.***out***.println("8. startsWith 'Hello': " + str.startsWith("Hello"));

System.***out***.println("9. endsWith 'World!': " + str.endsWith("World!"));

System.***out***.println("10. EqualsIgnoreCase 'hello, world!': " + str.equalsIgnoreCase("hello, world!"));

System.***out***.println("11. toLowerCase: " + str.toLowerCase());

System.***out***.println("12. toUpperCase: " + str.toUpperCase());

System.***out***.println("13. isEmpty: " + str.isEmpty());

System.***out***.println("14. Length: " + str.length());

System.***out***.println("15. split by ',': ");

String[] parts = str.split(",");

for (String part : parts) {

System.***out***.println(" - " + part.trim());

}

}

}

Output :

A screen shot of a computer

Description automatically generated

8. Write a java program to implement string tokenizer.

Program :

package Lab\_4;

import java.util.StringTokenizer;

public class SimpleStringTokenizerExample {

public static void main(String[] args) {

String inputString = "Hello World, this is a simple example.";

// Using StringTokenizer to split by whitespace

StringTokenizer tokenizer = new StringTokenizer(inputString);

// Print tokens

System.***out***.println("Tokens:");

while (tokenizer.hasMoreTokens()) {

String token = tokenizer.nextToken();

System.***out***.println(token);

}

}

}

Output :

A screen shot of a computer

Description automatically generated