**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** myPackage;

**class** Calculator{

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

**return** a+b;

}

**public** **double** add(**double** a,**double** b) {

**return** a+b;

}

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

**return** a+b+c;

}

}

**public** **class** Calculator\_Demo {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Calculator cl=**new** Calculator();

System.***out***.println("Sum of two numbers :"+cl.add(4, 7));

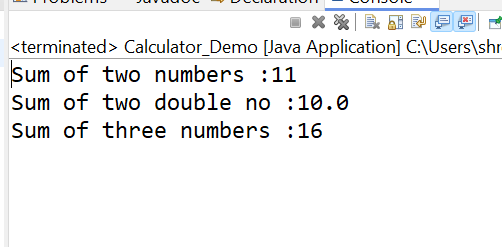
System.***out***.println("Sum of two double no :"+cl.add(6.3, 3.7));

System.***out***.println("Sum of three numbers :"+cl.add(5, 7, 4));

}

}

**Output:**



1. **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** myPackage;

**class** Person{

String name;

**int** age;

**public** Person(String name,**int** age) {

**this**.name=name;

**this**.age=age;

}

}

**class** Student **extends** Person{

**int** grade;

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

**super**(name,age);

**this**.grade=grade;

System.***out***.println("Name is :"+" "+ name+" "+"Age is :"+" "+age+"Grade is :"+grade);

}

}

**public** **class** SuperKeyword\_Demo {

**public** **static** **void** main(String[] args) {

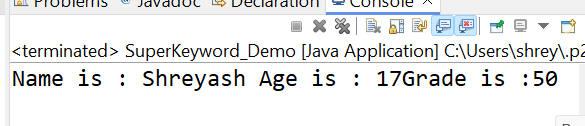
// **TODO** Auto-generated method stub

Student s=**new** Student("Shreyash", 17,50);

}

}

**Output:**



**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** myPackage;

//Shape class

**class** Shape {

**public** **void** draw() {

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

}

}

//Circle class, subclass of Shape

**class** Circle **extends** Shape {

@Override

**public** **void** draw() {

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

**super**.draw(); // calling draw() method of the superclass Shape

}

}

**public** **class** Super\_Keyword {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

// Creating an object of Circle

Circle circle = **new** Circle();

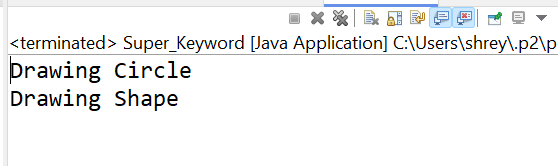
// Calling draw() method of Circle

circle.draw();

}

}

**Output:**



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

**Program:**

**package** myPackage;

**public** **class** Count\_Words\_in\_String {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String str = "Hello World, this is a sample string.";

**int** wordCount = *countWords*(str);

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

}

// Method to count words in a string without using predefined methods

**public** **static** **int** countWords(String str) {

**if** (str == **null** || str.isEmpty()) {

**return** 0;

}

**int** count = 0;

**boolean** word = **false**;

**int** endOfLine = str.length() - 1;

**for** (**int** i = 0; i <= endOfLine; i++) {

// If the character is a letter, it indicates start of a word

**if** (Character.*isLetter*(str.charAt(i)) && i != endOfLine) {

word = **true**;

}

// If the character is not a letter and it's not a continuation of a word

**else** **if** (!Character.*isLetter*(str.charAt(i)) && word) {

count++;

word = **false**;

}

// If the character is a letter at the end of the string

**else** **if** (Character.*isLetter*(str.charAt(i)) && i == endOfLine) {

count++;

}

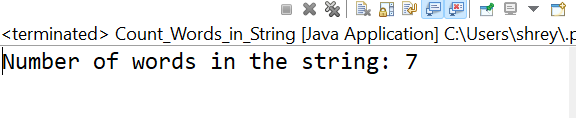
}

**return** count;

}

}

**Output:**



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

**Progaram:**

**package** myPackage;

**public** **class** Remove\_WhiteSpace {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String str = " Hello World ";

// Removing white spaces using regular expression

String trimmedString = *removeWhiteSpaces*(str);

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

System.***out***.println("String after removing white spaces: \"" + trimmedString + "\"");

}

**public** **static** String removeWhiteSpaces(String str) {

// Using regular expression to remove white spaces

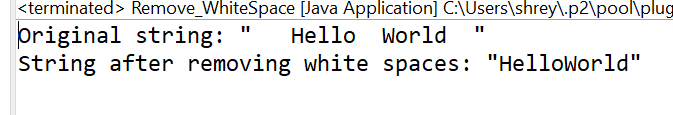
// \\s+ matches one or more white space characters

**return** str.replaceAll("\\s+", "");

}

}

**Output:**



1. **WAP to find occurrence of given in the given string.**

**Program:**

**package** myPackage;

**public** **class** Find\_WAP {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String mainString = "Java is a programming language. Java is widely used.";

String substringToFind = "Java";

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

System.***out***.println("Occurrences of \"" + substringToFind + "\" in the string: " + count);

}

**public** **static** **int** countOccurrences(String mainString, String substringToFind) {

**int** count = 0;

**int** index = mainString.indexOf(substringToFind);

// Iterate through the string and count occurrences using indexOf

**while** (index != -1) {

count++;

// Move to the next index after the last occurrence

index = mainString.indexOf(substringToFind, index + substringToFind.length());

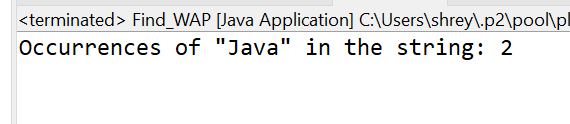
}

**return** count;

}

}

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** myPackage;

**public** **class** String\_Methods {

// **TODO** Auto-generated method stub

**public** **static** **void** main(String[] args) {

String str = "Hello, World!";

String str2 = "HELLO, MAAM!";

String str3 = "hello";

String str4 = "";

// replace method

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

// contains method

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

// replaceAll method

System.***out***.println("replaceAll: " + str.replaceAll("World", "Java"));

// indexOf method

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

// substring method

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

// equals method

System.***out***.println("equals: " + str.equals(str2));

// lastIndexOf method

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

// startsWith method

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

// endsWith method

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

// equalsIgnoreCase methid

System.***out***.println("equalsIgnoreCase: " + str.equalsIgnoreCase(str2));

// toLowerCase method

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

// toUpperCase method

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

// isEmpty method

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

// length method

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

// split method

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

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

**for** (String part : parts) {

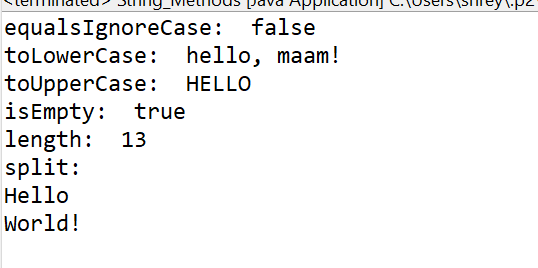
System.***out***.println(part);

}

}

}

**Output:**



**8. Write a java program to implement string tokenizer.**

**Program:**

**package** myPackage;

**import** java.util.StringTokenizer;

**public** **class** Java\_Tokenizers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String sentence = "Hello, how are you today?";

// Create a StringTokenizer object with comma and space as delimiters

StringTokenizer tokenizer = **new** StringTokenizer(sentence, ", ");

// Print all tokens

**while** (tokenizer.hasMoreTokens()) {

System.***out***.println(tokenizer.nextToken());

}

}

}

**Output:**

