LAB 4

Solve below question:-

QUE 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

INPUT:-

**package** overloading;

//creating a main class

**class** Calcultor

{

//for addiding two integer number

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

{

**return** a+b;

}

//for addiding two double number

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

{

**return** a+b;

}

//for addiding three integer number

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

{

**return** a+b+c;

}

**public** **int** add(**int**...numbers)

{

**int** sum=0;

**for**(**int** number:numbers)

{

sum+=number;

}

**return** sum;

}

}

**public** **class** OverloadingMethod

{

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

{

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

//creating the object

Calcultor calc=**new** Calcultor();

//calculation

System.***out***.println("Sum of 2 and 3: " + calc.add(2, 3));

System.***out***.println("Sum of 2.5 and 3.5: " + calc.add(2.5, 3.5));

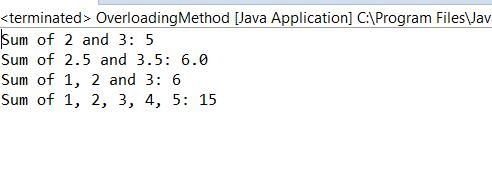
System.***out***.println("Sum of 1, 2 and 3: " + calc.add(1, 2, 3));

System.***out***.println("Sum of 1, 2, 3, 4, 5: " + calc.add(1, 2, 3, 4, 5));

}

}

OUTPUT:-



QUE 2:- uper 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.

INPUT:-

**package** SuperKeyWord;

// Define the Person class

**class** Person {

// Properties of the Person class

**private** String name;

**private** **int** age;

// Constructor of the Person class

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

**this**.name = name;

**this**.age = age;

}

// Getter method for name

**public** String getName() {

**return** name;

}

// Getter method for age

**public** **int** getAge() {

**return** age;

}

}

// Define the Student class which is a subclass of Person

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

// Additional property for the Student class

**private** String grade;

// Constructor of the Student class

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

// Call the constructor of the Person class using super

**super**(name, age);

// Initialize the grade property

**this**.grade = grade;

}

// Getter method for grade

**public** String getGrade() {

**return** grade;

}

// Method to display student details

**public** **void** displayStudentDetails() {

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

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

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

}

}

// Main class to demonstrate the creation of Student objects

**public** **class** SuperKeywords {

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

// Create a Student object

Student student1 = **new** Student("Alice", 20, "A");

// Display the details of the student

student1.displayStudentDetails();

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

// Create another Student object

Student student2 = **new** Student("Bob", 22, "B");

// Display the details of the student

student2.displayStudentDetails();

}

}

OUTPUT:-



QUE 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.

INPUT:-

**package** SuperKeyWord;

**class** Shape

{

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

{

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

}

}

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

{

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

{

**super**.draw();

System.***out***.println("Drawing the circle");

}

}

**public** **class** SuperKeyword

{

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

{

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

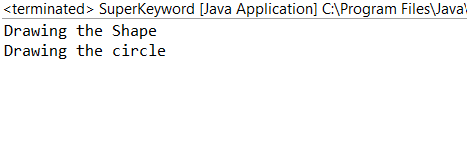
Shape s=**new** Circle();

s.draw();

}

}

OUTPUT:-



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

INPUT:- import java.util.Scanner;

public class WordCount {

public static void main(String[] args) {

// Create a Scanner object to read input from the user

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter the string

System.out.print("Enter a string: ");

// Read the entire line of input as the string

String inputString = scanner.nextLine();

// Call the countWords method to count the number of words in the input string

int wordCount = countWords(inputString);

// Display the number of words

System.out.println("The number of words in the string is: " + wordCount);

// Close the scanner to free up resources

scanner.close();

}

// Method to count the number of words in a string

public static int countWords(String str) {

// Initialize the word count to 0

int wordCount = 0;

// Get the length of the string

int length = str.length();

// Initialize a flag to indicate if we are inside a word

boolean isWord = false;

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

// Check if the current character is a letter or digit (part of a word)

if (Character.isLetterOrDigit(str.charAt(i))) {

if (!isWord) {

// We are entering a new word

wordCount++;

isWord = true;

}

} else {

// We are outside a word

isWord = false;

}

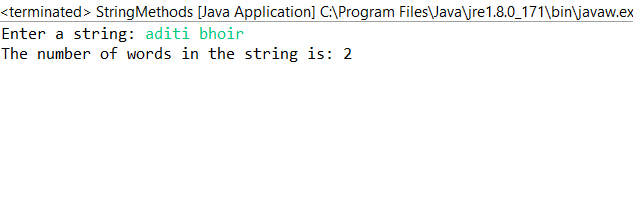
}

return wordCount;

}

}

OUTPUT:-



QUE 5:- Write a Java Program to remove all white spaces from a String?

INPUT:-

**package** StringFunction;

**public** **class** TrimFunction {

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

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

String name=" Audip Foundation ";

System.***out***.println(name.trim());

}

}

OUTPUT:-



QUE 6:- WAP to find occurrence of given in the given string.

INPUT:-

**package** String;

**import** java.util.Scanner;

**public** **class** StringOccurence {

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

// Create a Scanner object to read input from the user

Scanner scanner = **new** Scanner(System.***in***);

// Prompt the user to enter the main string

System.***out***.print("Enter the main string: ");

// Read the entire line of input as the main string

String mainString = scanner.nextLine();

// Prompt the user to enter the substring to find

System.***out***.print("Enter the substring to find: ");

// Read the entire line of input as the substring

String subString = scanner.nextLine();

// Call the findOccurrences method to count the occurrences of the substring

**int** occurrences = *findOccurrences*(mainString, subString);

// Display the number of occurrences

System.***out***.println("The substring '" + subString + "' occurred " + occurrences + " times in the main string.");

// Close the scanner to free up resources

scanner.close();

}

// Method to find the number of occurrences of subString in mainString

**public** **static** **int** findOccurrences(String mainString, String subString) {

// Initialize count of occurrences to 0

**int** count = 0;

// Start searching from the beginning of the main string

**int** fromIndex = 0;

// Loop to find all occurrences of the substring

**while** ((fromIndex = mainString.indexOf(subString, fromIndex)) != -1) {

// Increment the count for each occurrence found

count++;

// Move the start index past the current occurrence to search for further occurrences

fromIndex += subString.length();

}

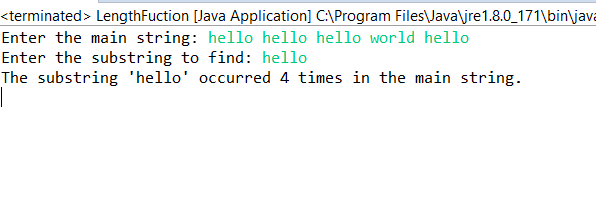
// Return the total count of occurrences

**return** count;

}

}

OUTPUT:-



QUE 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

INPUT:-

**package** String;

//importing the Scanner

**import** java.util.Scanner;

**public** **class** StringFunctions {

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

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

String name;

//initiating the scanner

Scanner sc= **new** Scanner(System.***in***);

//entering the string

System.***out***.println("Enter the String: ");

name=sc.next();

//closing the scanner

sc.close();

//for finding the length of the string

System.***out***.println("The length of the given string is: "+name.length());

//getting the Uppercase of the string

System.***out***.println("The UpperCase of the give String is: "+name.toUpperCase());

//getting the lowercase of the string

System.***out***.println("The LowerCase of the given String is: "+name.toLowerCase());

//for finding char at as in which number

System.***out***.println("char at: "+name.charAt(1));

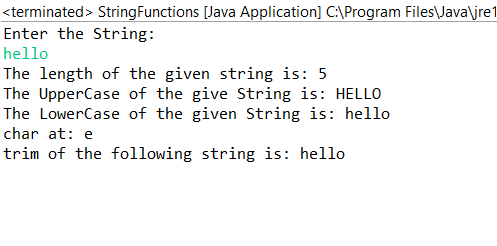
//removing whitespace of the string

System.***out***.println("trim of the following string is: "+name.trim());

}

}

OUTPUT:-



**INPUT:-**

**//Split()**

**package** String;

**public** **class** StringsFunctions {

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

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

String name="Anudip";

**// for splitting the function**

String[]nameArray=name.split("d");

**for**(String eachString:nameArray) {

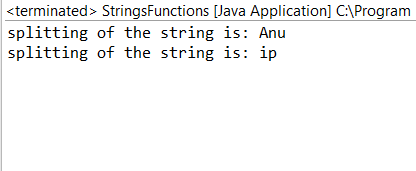
System.***out***.println(“splitting of the string is: “+eachString);

}

}

}

OUTPUT:-



INPUT:-

//ENDSWITH()

**package** String;

**public** **class** StringsFunctions {

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

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

String name="Anudip";

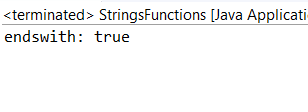
//checking that the given string is ending the same word or not that we have entered

System.***out***.println("endswith: "+name.endsWith("p"));

}

}

OUTPUT:-



INPUT:-

//Startswith()

**package** String;

**public** **class** StringsFunctions {

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

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

String name="Anudip";

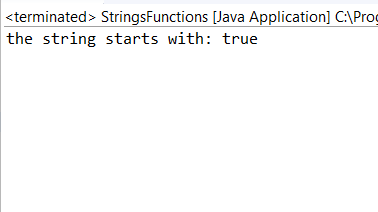
//checking that the given string is Starting the same word or not that we have entered

System.***out***.println("endswith: "+name.startsWith("A"));

}

}

OUTPUT:-



INPUT:-

//replace()

**package** String;

**public** **class** StringsFunctions {

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

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

String name="Anudip";

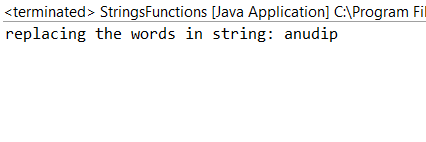
//checking that the given string is replacing the new word or not

System.***out***.println("endswith: "+name.replace("A", "a"));

}

}

OUTPUT:-



INPUT:-

//contain()

**package** String;

**public** **class** StringsFunctions {

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

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

String name="Anudip";

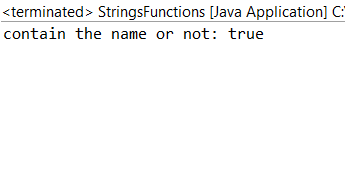
//checking that the given string is having the perticular word or not

System.***out***.println("contain the or not: "+name.contains("n"));

}

}

OUTPUT:-



INPUT:-

//replceAll

**package** String;

**public** **class** StringsFunctions {

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

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

String name="Anudip";

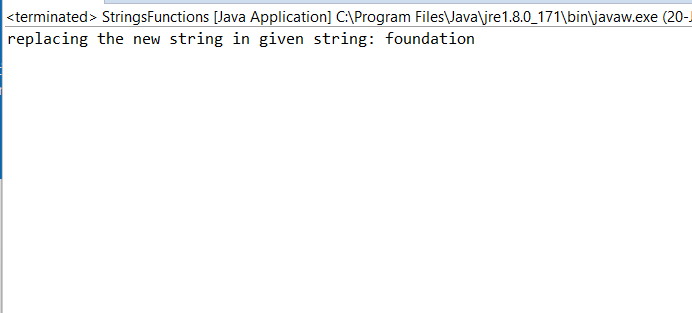
//checking that the given string is replacing the new string or not

System.***out***.println("replacing the new string in given string: "+name.replaceAll("Anudip", "foundation"));

}

}

OUTPUT:-



QUE 8:- Write a java program to implement string tokenizer.

INPUT:-

**package** String;

**import** java.util.StringTokenizer;

**public** **class** StringTokenize {

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

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

//assgining the string

StringTokenizer a=**new** StringTokenizer("hello good morning"," ");

**while**(a.hasMoreTokens()) {

//for getting the tokens for each word in the string

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

}

}

}

OUTPUT:-

