**JAVA**

**ANVI DHIMAN**

**PRACTICAL NO. – 01**

|  |
| --- |
| **Problem Statements:**  Q1. Write a program that takes two integers (values to given within the program) and display the output of the following operators: addition, subtraction, multiplication, division and modulus.  DoD 1: Assign two numbers to two variables.  DoD 2: Use separate variable to store the results.  Q2. Write a Java program to convert minutes into the number of years, months and days.  Definition of Done :  DoD 1: Ask the user to input the minutes  DoD 2: Display years and months and days in the sequence yy-mm-dd. |
| **Question Bank:**  1. Why Java is considered dynamic?  **Answer:**  Java is considered as dynamic because of Byte code [a class file]. A source code written in one platform, that can be executed in any platform.  2. What is Java Virtual Machine and how it is considered in context of Java’s platform independent feature?  **Answer:**  JVM is a specification that provide the **runtime environment** in which java bytecode can be executed. JVM submit a .class file to any operating system, JVM interprets bytecode into machine language.  3. List two Java IDE’s? List some Java keywords (unlike C, C++ keywords)?  **Answer:**  Java IDE – Netbeans, Eclipse, etc.  Java keywords – import, super, finally, etc.  4. Consider the following class:  public class IdentifyMyParts {  public static int x = 7;  public int y = 3;  }   1. What are the class variables?   **Ans:**  A class variable is an important part of object – oriented programming (OOP) that defines a specific attribute or property for a class and may be referred to as a member variable or static member variable.   1. What are the instance variables?   **Answer:**  Instance variables in Java are non-static variables which are defined in a class outside any method, [constructor](https://www.edureka.co/blog/constructor-in-java/) or a block. Each instantiated object of the class has a separate copy or instance of that variable. An instance variable belongs to a class. |
| **Flipped Practical’s**  1. What is the output from the following code:  IdentifyMyParts a = new IdentifyMyParts();  IdentifyMyParts b = new IdentifyMyParts();  a.y = 5;  b.y = 6;  a.x = 1;  b.x = 2;  System.out.println("a.y = " + a.y);  System.out.println("b.y = " + b.y);  System.out.println("a.x = " + a.x);  System.out.println("b.x = " + b.x);  System.out.println("IdentifyMyParts.x = " + IdentifyMyParts.x);  **Ans:**    **2.** What's wrong with the following program?  public class SomethingIsWrong {  public static void main(String[] args) {  Rectangle myRect;  myRect.width = 40;  myRect.height = 50;  System.out.println("myRect's area is " + myRect.area());  }  }  **Ans:**  **Output:-**  The issue of the given program is that the “myRect” object of the ‘Rectangle’ properly initialize before accessing the properties and the methods. |

**PROGRAM NO. - 01**

**Source Code -**

//taking input from user

//show use of arithmetic operator

import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

int n1, n2, add, subtract, multiply, mod;

float divide;

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Two Numbers : ");

n1 = scanner.nextInt();

n2 = scanner.nextInt();

add = n1 + n2;

subtract = n1 - n2;

multiply = n1 \* n2;

divide = n1 / n2;

mod = n1%n2;

System.out.println("Sum = " + add);

System.out.println("Difference = " + subtract);

System.out.println("Multiplication = " + multiply);

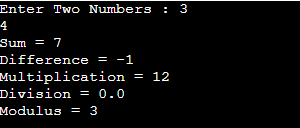
System.out.println("Division = " + divide);

System.out.println("Modulus = "+mod);

}

}

**Output –**

****

**PROGRAM N0. - 02**

**Source Code -**

import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

double minutesInYear = 60 \* 24 \* 365;

Scanner input = new Scanner(System.in);

System.out.print("Input the number of minutes: ");

double min = input.nextDouble();

long years = (long) (min / minutesInYear);

int days = (int) (min / 60 / 24) % 365;

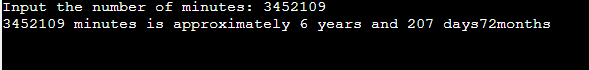
long months = years\*12;

System.out.println((int) min + " minutes is approximately " + years + " years and " + days + " days" + months +"months");

}

}

**Output –**

****

**PRACTICAL NO. – 02**

|  |
| --- |
| **Problem Statement**   1. Write a Java program to solve quadratic equations (use if, else if and else).   *Definition of Done:*  DoD 1: The program asks the values of coefficients of a quadratic equation.  DoD 2: The program should display the roots or an appropriate message.   1. Write a Java program that asks the user to provide a single character from the alphabet. Print Vowel or Consonant, depending on the user input. If the user input is not a letter (between a and z or A and Z), or is a string of length > 1, print an error message.   *Definition of Done:*  DoD 1: The program asks an input from the user.  DoD 2: A single character input is taken from the user or an error message is  generated.  DoD 3: The program should print “Vowel” if the entered character is a vowel and  “Consonant” if the entered character is a consonant.   1. Write a Java program to print following structure:   \*  \*\*  \*\*\*  \*\*\*\*  \*\*\*\*\*  \*\*\*\*\*\* |
| **Question Bank**  1. The most basic control flow statement supported by the Java programming language is the **if** statement.  2. The **switch** statement allows for any number of possible execution paths.  3. The **do-while** statement is similar to the while statement, but evaluates its expression at the **bottom** of the loop.  4. How do you write an infinite loop using the for statement?  **Answer:**  for(; ;){  }  5. How do you write an infinite loop using the while statement?  **Answer:**  while(true) {  }  6. Which looping process checks the test condition at the end of the loop?  **Answer:**  do-while loop  7. Why do we use continue statement?  **Answer:**  The continue keyword is used to end the current iteration in a for loop or while loop and continues to the next iteration.    8. What is the size of boolean variable?  **Answer:**  The size is 1-bit  9. Which looping process is best used when the number of iterations is known?  **Answer:**  for-loops are counter-controlled, meaning that they are normally used whenever the number of iterations is known in advance. |
| **Flipped Practical’s**  1. Consider the following code snippet.  if (aNumber >= 0)  if (aNumber == 0)  System.out.println("first string");  else System.out.println("second string");  System.out.println("third string");  a) What output do you think the code will produce if aNumber is 3?  **Answer:**  second string  third string    b) Write a test program containing the previous code snippet; make aNumber 3. What is the output of the program? Is it what you predicted? Explain why the output is what it is; in other words, what is the control flow for the code snippet?  **Answer:**  second string  third string  3 is greater than or equal to 0, so execution progresses to the second if statement. The second if statement's test fails because 3 is not equal to 0. Thus, the else clause executes (since it's attached to the second if statement). Thus, second string is displayed. The final println is completely outside of any if statement, so it always gets executed, and thus third string is always displayed.  c) Using only spaces and line breaks, reformat the code snippet to make the control flow easier to understand.  **Answer:**  if (aNumber >= 0)  if (aNumber == 0)  System.out.println("first string");  else  System.out.println("second string");  System.out.println("third string");  d) Use braces, { and }, to further clarify the code.  **Answer:**  if (aNumber >= 0) {  if (aNumber == 0) {  System.out.println("first string");  } else {  System.out.println("second string");  }  }  System.out.println("third string");  2. What's wrong? for (int k = 2, k <= 12, k++)  **Answer:**  There is always used semicolons ; at the end of the any statement for terminate the line.  3.If there is more than one statement in the block of a for loop, what must be placed at the beginning and the ending of the loop block?  **Answer:**  braces { }  4. What value is stored in num at the end of this looping?  for (num = 1; num <= 5; num++)  **Answer:**  Store in num vale is 5  The loop be start at the 1 and end the 5, so the answer is 5.  For loop are the loop used to repeat any string or any output number of times or display a series of number or alphabet. |

**PROGRAM N0. - 01**

**Source Code -**

// print the star pattern

import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

//i for the row and j for the columns

// row denote the number of rows you want to print

int i, j, row=6;

for(i=0; i<row; i++)

{

for(j=0; j<=i; j++)

{

//print the star

System.out.print("\*");

}

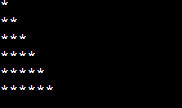
System.out.println();

}

}

}

**Output –**



**PROGRAM N0. - 02**

**Source Code -**

//Write a program to solve the quadratic equation (use if, else if and else)

//Definition done:

//DoD1: The program asks the value of coefficients of a quadratic equation.

//DoD2: The program should display the roots or an appropriate message

import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Input a: ");

double a = input.nextDouble();

System.out.print("Input b: ");

double b = input.nextDouble();

System.out.print("Input c: ");

double c = input.nextDouble();

double result = b \* b - 4.0 \* a \* c;

if (result > 0.0) {

double r1 = (-b + Math.pow(result, 0.5)) / (2.0 \* a);

double r2 = (-b - Math.pow(result, 0.5)) / (2.0 \* a);

System.out.println("The roots are " + r1 + " and " + r2);

} else if (result == 0.0) {

double r1 = -b / (2.0 \* a);

System.out.println("The root is " + r1);

} else {

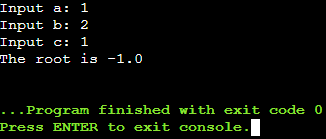
System.out.println("The equation has no real roots.");

}

}

}

**Output –**

****

**PROGRAM N0. - 03**

**Source Code –**

import java.util.Scanner;

public class Main

{

public static void main (String[]args){

Scanner in = new Scanner (System.in);

System.out.print ("Input an alphabet: ");

String input = in.next ().toLowerCase ();

boolean uppercase = input.charAt (0) >= 65 && input.charAt (0) <= 90;

boolean lowercase = input.charAt (0) >= 97 && input.charAt (0) <= 122;

boolean vowels = input.equals ("a") || input.equals ("e")

|| input.equals ("i") || input.equals ("o") || input.equals ("u");

if (input.length () > 1)

{

System.out.println ("Error. Not a single character.");

}

else if (!(uppercase || lowercase))

{

System.out.println

("Error. Not a letter. Enter uppercase or lowercase letter.");

}

else if (vowels)

{

System.out.println ("Input letter is Vowel");

}

else

{

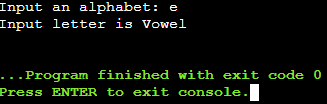
System.out.println ("Input letter is Consonant");

}

}

}

**Output –**

****

**PRACTICAL NO. 03**

|  |
| --- |
| **Problem Statement**  1. Write a Java program to find the maximum and minimum value of an array.  Definition of Done:  DoD 1: The program should ask the user to enter the elements of the array.  DoD 2: The program should display the maximum and minimum elements of the array.  2. Write a Java program to find the index of an array element in an array of size 10. The program should not use any function other than main ( ) functions.  Definition of Done:  DoD 1: The program should ask the user to enter the elements of the array.  DoD 2: The program should ask the user to enter a number to search.  DoD 3: The program should display the elements of the array entered.  DoD 4: The program should display the index of the number if the item is present or display -1 of the elements is not present.    3. Write a Java Program to count even and odd numbers in an array.  Definition of Done  DoD 1: The program should ask the user to enter the elements of the array.  DoD 2: Even elements will be stored in EvenArray[] and odd elements will be stored in oddArray[].  DoD 3: Display all three arrays along with their length.  4. Write a Java program to read numbers in an integer array of size 5 and display the following (using functions for each functionality):  i) Sum of all the elements  ii) Sum of alternate elements in the array.  Definition of Done  DoD 1: The program should ask the user to enter the elements of the array.  DoD 2: The program should display a menu with the above choices and ask the user to choose one of the choices. |
| **Question Bank:**  1. Can you pass the negative number as an array size?  **Answer:**  No, you cannot use a negative integer as size, the size of an array represents the number of elements in it, –ve number of elements in an array makes no sense.  2. Can you change the size of the array once you define it?  **Answer:**  You can’t change the size of array after it’s constructed. However, you can change the number of elements in an array list whenever you want.  3. What is an anonymous array?  **Answer:**  An array in java without any name is known as anonymous array. It is an array just for creating and using instantly. Using an anonymous array, we can pass any array with user value without the referenced variable.  4. What is the difference between int[] a and int a[] ?  **Answer:**  There is no difference and any of mentioned syntaxes can be used to declare the array.  5. What are jagged arrays in java? Give example?  **Answer:**  Each row is two-dimensional array is itself an array. Thus, the row can have different lengths. An array is this find known as Jagged Array.  **For Example:**  int[] [] angleArray = new int[5][];  angleArray[0] = new int[4];  angleArray[0] = new int[3];  angleArray[0] = new int[1];  angleArray[0] = new int[2]; |
| **Flipped Practical’s**  1.Which of these is an incorrect array declaration? a) int arr[] = new int[5] b) int [] arr = new int[5] c) int arr[] = new int[5]  d) int arr[] = int [5] new  **Answer:**  Option D is incorrect array declaration. Operator new should preceded by array size(int[5]).  2. What will be the output of the following program?  public class MyFirst {        public static void main(String[] args) {           MyFirst obj = new MyFirst(n);   }   static int a = 10;   static int n;   int b = 5;   int c;   public MyFirst(int m) {         System.out.println(a + ", " + b + ", " + c + ", " + n + ", " + m);     }  // Instance Block    {       b = 30;       n = 20;    }  // Static Block    static  {            a = 60;       }   }  **Output:** |

**Program No. – 01**

**Source Code :-**

import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

int count, max, min, i;

int[] inp = new int[500];

Scanner in = new Scanner(System.in);

//take the input of array size by the user

System.out.println("Enter number of elements");

count = in.nextInt();

//enter the array elements by the user

System.out.println("Enter " + count + " elements");

for(i = 0; i < count; i++) {

inp[i] = in.nextInt();

}

max = min = inp[0];

for(i = 1; i < count; i++) {

if(inp[i] > max)

max = inp[i];

else if (inp[i] < min)

min = inp[i];

}

//print the largest and smallest number of array

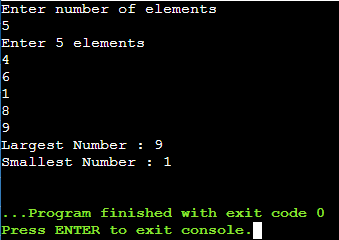
System.out.println("Largest Number : " + max);

System.out.println("Smallest Number : " + min);

}

}

**Output :-**

****

**Program No. :- 02**

**Source Code:-**

import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

int n, x, flag = 0, i=0;

Scanner sc = new Scanner(System.in);

//take input from by the array size System.out.print("System.out.println("Enet);

System.out.println("Enter no. of elements you want in array : ");

n = sc.nextInt();

int a[] = new int [n];

//enter the all elements of array by the user input

System.out.println("Enter the array elements : ");

for(i=0;i<n;i++)

{

a[i] = sc.nextInt();

}

System.out.println("Enter the elements you want to find : ");

x = sc.nextInt();

for(i=0; i<n; i++)

{

if(a[i] == x)

{

flag = 1;

//System.out.println("elements found at position : "+(i+1));

break;

}

else

{

flag = 0;

}

}

if(flag == 1)

{

System.out.println("Elements found at position :"+(i+1));

}

else{

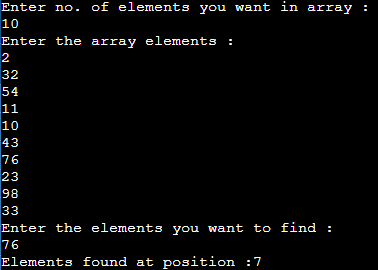
System.out.println("-1");

}

}

}

**Output:-**

****

**Program No. – 03**

**Source Code:-**

// Java Program to Count Even and Odd Numbers in an Array using For Loop

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

int i,even=0,odd=0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of array");

int size = sc.nextInt();

int[] elements = new int[size];

System.out.println("Enter the elements of array");

for(i=0;i<size;i++)

{

elements[i]=sc.nextInt();

}

for(i=0;i<size;i++)

{

if(elements[i]%2==0)

even++;

else

odd++;

}

System.out.println("Number of even elements: " +even);

System.out.println("Number of odd elements: " +odd);

int[] evenArray = new int[even];

int[] oddArray = new int[odd];

int j=0,k=0;

for(i=0;i<size;i++)

{

if(elements[i]%2==0)

evenArray[j++]=elements[i];

else

oddArray[k++]=elements[i];

}

System.out.println("Even Array");

for(i=0;i<even;i++)

System.out.print(" " +evenArray[i]);

System.out.println("\nodd Array");

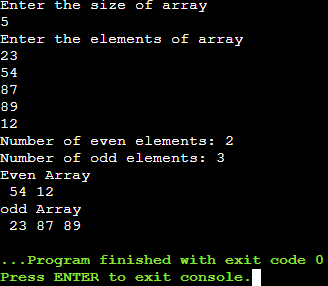
for(i=0;i<odd;i++)

System.out.print(" " +oddArray[i]);

}

}

**Output –**



**Q.04.** Write a Java program to read numbers in an integer array of size 5 and display the following (using functions for each functionality):

i) Sum of all the elements

ii) Sum of alternate elements in the array.

Definition of Done

DoD 1: The program should ask the user to enter the elements of the array.

DoD 2: The program should display a menu with the above choices and ask the user to choose one of the choices.

**Program No – 04**

**Source code –**

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

int i,even=0,odd=0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of array");

int size = sc.nextInt();

int[] elements = new int[size];

System.out.println("Enter the elements of array");

for(i=0;i<size;i++)

{

elements[i]=sc.nextInt();

}

System.out.println("Enter the choice 1 or 2");

System.out.println("1. Sum of all the elements");

System.out.println("2. Sum of alternative elements");

int sum=0;

int total=0;

int choice = sc.nextInt();

switch(choice)

{

case 1:

for(i=0;i<size;i++)

{

sum=sum+elements[i];

}

System.out.println("Sum of all the elements: " +sum);

break;

case 2:

for(i=0;i<size;i++)

{

total=total+elements[i];

i=i+1;

}

System.out.println("Sum of alternative elements: " +total);

break;

default:

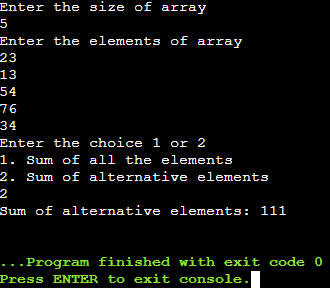
System.out.println("Error");

}

}

}

**Output –**



**PRACTICAL NO. 04**

|  |
| --- |
| **Problem Statements**   1. Create a class named 'Student' with String variable 'name' and integer variable 'roll\_no'. Assign the value of roll\_no as '2' and that of name as "John" by creating an object of the class Student. 2. Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' without any parameter in its constructor. 3. Write a program to print the area of a rectangle by creating a class named 'Area' taking the values of its length and breadth as parameters of its constructor and having a method named 'returnArea' which returns the area of the rectangle. Length and breadth of the rectangle are entered through the keyboard. 4. Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by the user. 5. Write a program to calculate the distance between two points (x1, y1) and (x2, y2). All numbers and return values should be of type double.   Definition of Done:  DoD 1: Two java files to be defined. One for class definitions and another for the application  DoD 2: A class point is defined with two float variables for x1 and x2 and the following functionality:  i. Non-parametrized and parameterized constructors are defined.  ii. Get and set methods are defined for all the instance variables.  iii. Distance function is defined to calculate the distance between two points.  iv. Display function is defined with width of 7 and precision of 2.  Write this program with a static method definition for calculating the distance between two points. |
| **Question Bank**  1. What is the difference between class and object?  **Ans:-** Class is a blue print or templet from which object are created and object is a instance of a class. Object is a real word entity and the class is a group of similar object.    2. What is constructor chaining?  **Ans:-** Constructor chaining is a sequence of invoking constructor upon ab object. It is used when we invoke a number of constructor, one after another by using only the instance.    3. What is No-arg constructor?  **Ans:-** A constructor that does not accept any argument.  4. What happens if you keep return type for a constructor?  **Ans:-** If we add a return type to a constructor, then it will become a method of the class.  5. What is the use of private constructor?  **Ans:-** The private constructor in Java is used to create a singleton class. A private constructor in Java ensure that only one object is created at a time.    6. Can we use this() in a method?  **Ans:-** The “this” keyword in java is used as a reference to the current object, within an instance method or constructor. Yes, you can call method using it.  7. Can we define a method with same name of class?  **Ans:-** Yes, It is allow to define the method with the same name as that of a class. |
| **Flipped Practicals**  **1.** What will be the output of the following Java program?  **class** A  {  **int** i;  **int** j;  A()  {  i = 1;  j = 2;  }  }  **class** Output  {  **public** **static** **void** main(String args[])  {  A obj1 = **new** A();  A obj2 = **new** A();  System.out.print(obj1.equals(obj2));  }  }  a) false b) true c) 1 d) Compilation Error  **Ans:-** false |

**Program No – 1**

**Source code –**

public class ComplexNumber{

//for real and imaginary parts of complex numbers

double real, img;

//constructor to initialize the complex number

ComplexNumber(double r, double i){

this.real = r;

this.img = i;

}

public static ComplexNumber sum(ComplexNumber c1, ComplexNumber c2)

{

//creating a temporary complex number to hold the sum of two numbers

ComplexNumber temp = new ComplexNumber(0, 0);

temp.real = c1.real + c2.real;

temp.img = c1.img + c2.img;

//returning the output complex number

return temp;

}

public static void main(String args[])

{

ComplexNumber c1 = new ComplexNumber(5.5, 4);

ComplexNumber c2 = new ComplexNumber(1.2, 3.5);

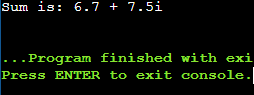
ComplexNumber temp = sum(c1, c2);

System.out.println("Sum is: "+ temp.real+" + "+ temp.img +"i");

}

}

**Output –**

****

**Program No – 2**

**Source code –**

import java.util.Scanner;

class area{

int length;

int breadth;

area(int a, int b)

{

length = a;

breadth = b;

}

public int areareturn()

{

return length \* breadth;

}

}

public class Rectangle4 {

public static void main(String[] args) {

Scanner obj = new Scanner(System.in);

int a,b;

System.out.println("Enter the lenght of Recatangle");

a = obj.nextInt();

obj.nextLine();

System.out.println("Enter the breadth of Recatangle");

b = obj.nextInt();

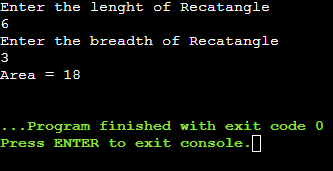
area ob1 = new area(a,b);

System.out.println("Area = "+ ob1.areareturn());

}

}

**Output –**



**Program No – 3**

**Source code –**

import static java.lang.Math.sqrt;

class Triangle

{

int a,b,c;

public double getArea()

{

double p=(a+b+c)/2.0;

return Math.sqrt(p\*(p-a)\*(p-b)\*(p-c))

}

public double getPerimeter()

{

return (a+b+c)/2.0;

}

}

class Perimeter

{

public static void main(String args[])

{

Triangle t=new Triangle();

t.a=3;

t.b=4;

t.c=5;

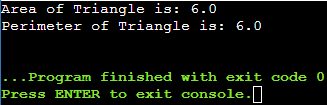
System.out.println("Area of Triangle is: " +t.getArea());

System.out.println("Perimeter of Triangle is: " +t.getPerimeter());

}

}

**Output –**



**Program No – 4**

**Source Code –**

import java.util.Scanner;

class CalDis

{

double dis;

CalDis(int x1,int y1,int x2,int y2)

{

dis=(Math.sqrt((x2-x1)\*(x2-x1) + (y2-y1)\*(y2-y1)));

}

}

class DistanceBwPoint

{

public static void main(String args[])

{

int x1,x2,y1,y2;

Scanner sc=new Scanner(System.in);

System.out.println("enter x1 point");

x1=sc.nextInt();

System.out.println("enter y1 point");

y1=sc.nextInt();

System.out.println("enter x2point");

x2=sc.nextInt();

System.out.println("enter y2 point")

y2=sc.nextInt();

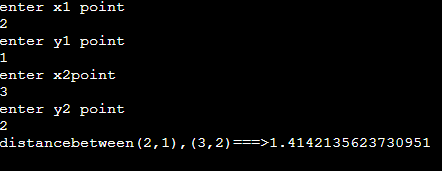
CalDis cd=new CalDis(x1,y1,x2,y2);

System.out.println("distancebetween"+"("+x1+","+y1+"),"+"("+x2+","+y2+")===>"+cd.dis);

}

}

**OUTPUT –**



**Program No – 5**

**Source Code –**

import java.util.Scanner;

public class Student {

String name;

int roll\_no;

public Student(String name, int roll\_no) {

this.name = name;

this.roll\_no = roll\_no;

}

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter student name: ");

String name = input.nextLine();

System.out.print("Enter student roll number: ");

int roll\_no = input.nextInt();

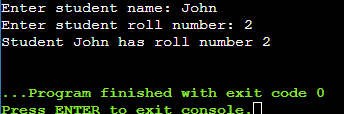
Student student = new Student(name, roll\_no);

System.out.printf("Student %s has roll number %d\n", student.name, student.roll\_no);

}

}

**OUTPUT –**

****

**PRACTICAL NO: 5**

|  |
| --- |
| **Program Statement**   1. Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.     2. Create three classes:  Class Vehicle:  Vehicle Class will contain a display() function, which will say "This is a Vehicle".  Class Car:  Car Class will derive the Vehicle Class and overwrite its display() function. it will say "This is a Car".  Class Bike:  Bike Class will derive the Vehicle Class and overwrite its display() function. it will say "This is a Bike".  Write an application that reads an Integer N, which will denote the number of tyres in the vehicle. You have to create an object of the appropriate class according to the value of N and use it display() function.  If N = 2, Create a Bike Object.  If N = 4, Create a Car Object.  Create a Vehicle Object, otherwise.  Definition of Done:  DoD 1: Each class definition is stored in its own .java file.  DoD 2: Switch statement is used for identifying the appropriate class for which the object is to be invoked.    3. Define a class Box with the following instance variables: width, height and depth, all of type float. Create a new class BoxWeigth that extends Box to include weight as an instance variable. Write an application that tests the functionalities of both these classes.  Definition of Done:  DoD 1: Three java files to be defined. One for each class definition: Box, BoxWeight and BoxWeightDemo.  DoD 2: Box and BoxWeight should have three types of constructors defined: clone of an object, all dimensions specified as arguments, no argument.  DoD 3: Super is used to call base class constructors in derived class  DoD 4: Get and set functions defined as applicable in Box and BoxWeight classes.  DoD 5: Function to display volume in Box class and weight in BoxWeigth class |
| **Question Bank**  1. What is the use of super keyword?  **Ans:-** The super keywordrefers to super class objects. It is used to call superclass method and to access the superclass constructor. The **super** keyword in Java is a reference variable which is used to refer immediate parent class object.  2. What is multi-level inheritance?  **Ans:-** The multi-level inheritance include the involvement of at least two or two more classes. One class inherits the features from a parent class and newly created the sub-class becomes the base class for another new class.  3. What is the usage of inheritance?  **Ans:-** The most important use of the inheritance in Java is code reusability. |
| **Flipped Questions**  Q1. What is the output of the following?  class A  {      {          System.out.println(1);      }  }    class B extends A  {      {          System.out.println(2);      }  }    class C extends B  {      {          System.out.println(3);      }  }    public class MainClass  {      public static void main(String[] args)      {          C c = new C();      }  }  **Output:-**    Q2. What is the output of the following?  class A  {      public A()      {          System.out.println("Class A Constructor");      }  }    class B extends A  {      public B()      {          System.out.println("Class B Constructor");      }  }    class C extends B  {      public C()      {          System.out.println("Class C Constructor");      }  }    public class MainClass  {      public static void main(String[] args)      {          C c = new C();  **Output:-** |

**Program No – 01**

**Source code –**

class Shape {

void sh()

{

System.out.println("This is shape ");

}

}

class Rectangle extends Shape {

void rec()

{

System.out.println("This is rectangle ");

}

}

class Circel extends Shape{

void ci()

{

System.out.println("This is circel");

}

}

class Square extends Rectangle{

void sq()

{

System.out.println("Square is Rectangle");

}

}

class TestInheritance3{

public static void main(String args[])

{

Square obj = new Square();

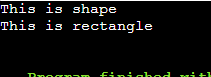
obj.sh();

obj.rec();

}

}

**OUTPUT -**

****

**Program No – 02**

**Source code –**

import java.util.\*;

class Vehicle{

public void Display()

{

System.out.println("this is a vehical");

}

}

class Car extends Vehicle

{

public void Display()

{

System.out.println("this is car");

}

}

class Bike extends Vehicle

{

public void Display()

{

System.out.println("this is bike");

}

}

class MakeObject

{

public static void main(String [] args){

Scanner sc=new Scanner(System.in);

System.out.println("enter the number of wheels in vehicle ");

int N = sc.nextInt();

if(N==2)

{

Bike ob = new Bike();

ob.Display();

}

else if(N==4)

{

Car ob1=new Car();

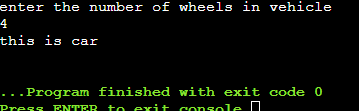
ob1.Display();

}

}

}

**OUTPUT –**

****

**Program No – 03**

**Source code –**

import java.util.Scanner;

class Box {

float width, height, depth;

public Box(float w, float h, float d) {

width = w;

height = h;

depth = d;

}

public void setWidth(float w) {

width = w;

}

public void setHeight(float h) {

height = h;

}

public void setDepth(float d) {

depth = d;

}

public float getWidth() {

return width;

}

public float getHeight() {

return height;

}

public float getDepth() {

return depth;

}

public void displayVolume() {

System.out.println("Volume of Box: " + width \* height \* depth);

}

}

class BoxWeight extends Box {

private float weight;

public BoxWeight(float w, float h, float d, float wt) {

super(w, h, d);

weight = wt;

}

public void setWeight(float wt) {

weight = wt;

}

public float getWeight() {

return weight;

}

public void displayWeight() {

System.out.println("Weight of Box: " + weight);

}

}

public class BoxWeightDemo{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter width: ");

float width = sc.nextFloat();

System.out.print("Enter height: ");

float height = sc.nextFloat();

System.out.print("Enter depth: ");

float depth = sc.nextFloat();

System.out.print("Enter weight: ");

float weight = sc.nextFloat();

BoxWeight bw = new BoxWeight(width, height, depth, weight);

bw.displayVolume();

bw.displayWeight();

sc.close();

}

}

**OUTPUT –**

