**Experiment No. 1 (a)**

**Aim**

Java program to print whether the year entered by the user is Leap Year or Not.

**Source code**

package java\_file;

import java.util.Scanner;

public class \_1a\_Leap\_Year {

public static void main(String[] args) {

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

System.**out**.println("Enter the Year to Check for leap Year ???");

int year=input.nextInt();

if(((year%4==0)&&(year%100!=0))||(year%400==0))

{

System.**out**.println("\n\n"+year+" is a LEAP YEAR");

}

else

{

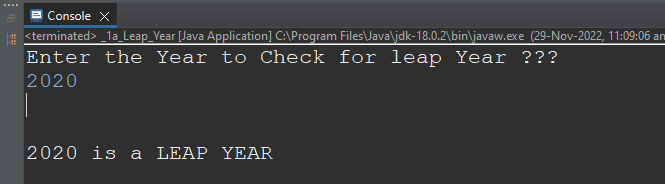
System.**out**.println("\n\n"+year+" is a COMMON YEAR");

}

}

}

**Output**



**Experiment No. 1 (b)**

**Aim**

Java program to Swap 2 Numbers without using Third Variable.

**Source code**

package java\_file;

import java.util.Scanner;

public class \_1b\_Swap {

public static void main(String[] args) {

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

System.**out**.println("Enter 2 Values???");

int a=input.nextInt();

int b=input.nextInt();

System.**out**.print("Before Swapping....\n"+"A="+a+"\nB="+b);

a=a+b;

b=a-b;

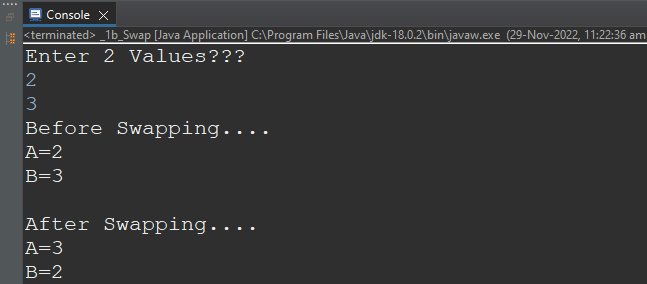
a=a-b;

System.**out**.print("\n\nAfter Swapping....\n"+"A="+a+"\nB="+b);

}

}

**Output**



**Experiment No. 2 (a)**

**Aim**

Java program to Print the Fibonacci Series.

**Source code**

package java\_file;

import java.util.Scanner;

public class \_2a\_Fibonacci {

public static void main(String[] args) {

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

System.**out**.println("How many Elements of Fibonacci Series to Print from 0 ???");

int a=input.nextInt();

int full=10;

int fib[]=new int[a];

fib[0]=0;

fib[1]=1;

System.**out**.println("\n\nFibonacci Series\n");

System.**out**.print("0\t1\t");

for(int i=2;i<a;i++)

{

fib[i]=fib[i-1]+fib[i-2];

System.**out**.print(fib[i]+"\t");

if(i==full)

{

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

full+=10;

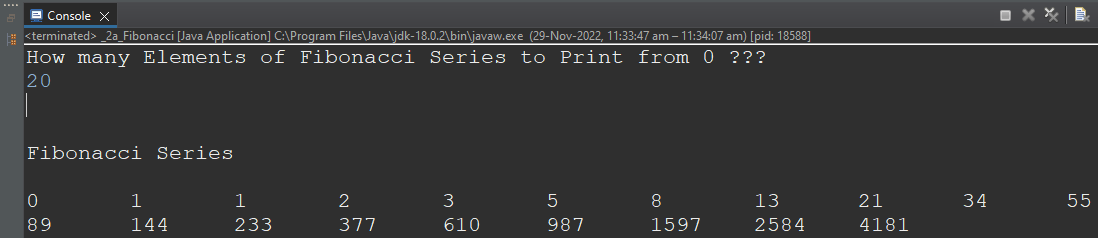
}

}

}

}

**Output**

****

**Experiment No. 2 (b)**

**Aim**

Java program to Print the Pyramid Star Pattern (Ask the user the Number of Rows using the Scanner class).

**Source code**

package java\_file;

import java.util.Scanner;

public class \_2b\_Pyramid {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.println("How Many Rows ???");

int z=input.nextInt();

int y=1;

System.out.println("\n\nSTAR PYRAMID\n");

for(int i=z;i>=1;i--)

{

for(int j=1;j<=i;j++)

System.out.print(" ");

int k=1;

while(k<=y)

{

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

k++;

}

y+=2;

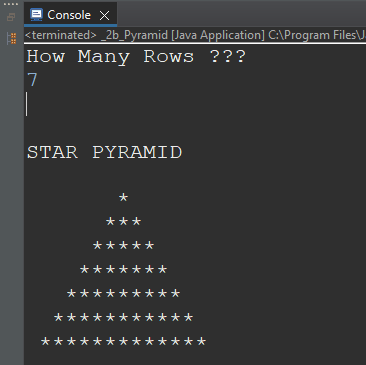
System.out.println("");

}

}

}

**Output**



**Experiment No. 3**

**Aim**

Java program to print Calculate Average of numbers using Arrays.

**Source code**

package java\_file;

public class \_3\_Average {

public static void main(String[] args) {

float arr[]= {27,71,39,85,13,69,77};

float temp=0;

System.out.println("Calculating AVERAGE.......");

for(int i=0;i<7;i++)

{

temp=temp+arr[i];

}

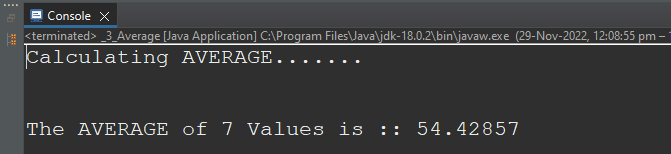
float avg=temp/7f;

System.out.println("\n\nThe AVERAGE of 7 Values is :: "+avg);

}

}

**Output**



**Experiment No. 4**

**Aim**

Create a class with 2 methods input() and sum()

Input(): take user input, and call the method sum() then print the sum

Sum(); calculate the sum and return the value to the input method.

**Source code**

package java\_file;

import java.util.Scanner;

public class \_4\_Sum {

float sum[]=new float[5];

public static void main(String[] args) {

\_4\_Sum obj=new \_4\_Sum();

obj.input();

}

void input() {

Scanner read=new Scanner(System.in);

System.out.println("Enter 5 Values ???");

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

sum[i]=read.nextFloat();

}

System.out.println("\n\nThe SUM of 5 elements is :: "+sum());

}

float sum() {

float temp=0;

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

temp+=sum[i];

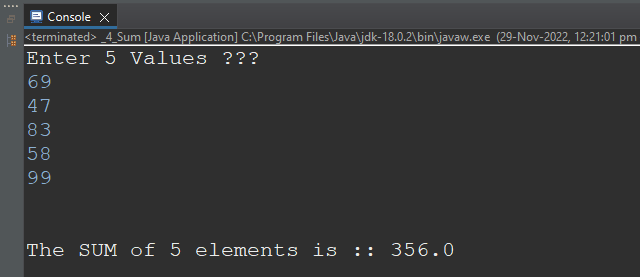
}

return temp;

}

}

**Output**



**Experiment No. 5**

**Aim**

Define a class Employee with the following specifications:

Data Member:

empno, ename, basic, hra, da, netpay

Member Methods:

haveData() method to accept values for empno, ename, basic, hra, da & invoke the method

calculate() for netpay.

dispData() method to display all the data members on the screen.

**Source code**

package java\_file;

public class \_5\_Employee {

int empno;

String ename;

float basic;

float hra;

float da;

float netpay;

public static void main(String[] args) {

\_5\_Employee obj=new \_5\_Employee();

obj.haveData(420,"Abhinay",80000,15000,5000);

obj.dispData();

}

void haveData(int a, String b, float c, float d, float e) {

empno=a;

ename=b;

basic=c;

hra=d;

da=e;

calculate();

}

void calculate() {

netpay=basic+hra+da;

System.out.println("The Net Pay is "+netpay);

}

void dispData() {

System.out.println("\n\nYour Salary Details....");

System.out.println("Employee No. \t:: "+empno);

System.out.println("Employee Name\t:: "+ename);

System.out.println("Basic Salary \t:: "+basic);

System.out.println("HRA \t:: "+hra);

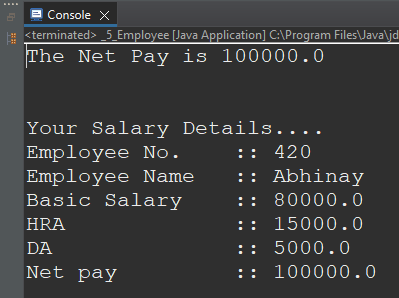
System.out.println("DA \t:: "+da);

System.out.println("Net pay \t:: "+netpay);

}

}

**Output**



**Experiment No. 6**

**Aim**

Program to find area of Square, Rectangle and Circle using Method Overloading.

**Source code**

package java\_file;

public class \_6\_Area {

float pi=3.14f;

public static void main(String args[]) {

\_6\_Area myobj=new \_6\_Area();

System.out.println("Values are in CM");

System.out.println("\n\nSide = "+20);

myobj.findArea(20);

System.out.println("\n\nLength = 30\nBreadth = 15");

myobj.findArea(30, 15);

System.out.println("\n\nRadius = "+10.5f);

myobj.findArea(10.5f);

}

public void findArea(int a) {

System.out.println("Area of Square = "+(a\*a));

}

public void findArea(int len, int wid) {

System.out.println("Area of Rectangle = "+(len\*wid));

}

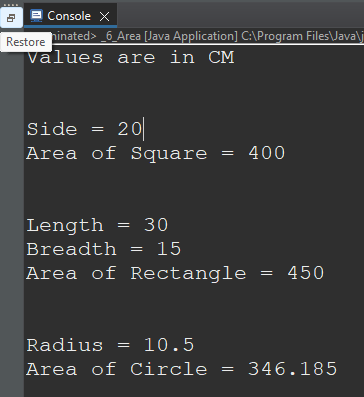
public void findArea(float rad) {

System.out.println("Area of Circle = "+(pi\*rad\*rad));

}

}

**Output**



**Experiment No. 7**

**Aim**

Construct a class named product with data members PNumber, PName, PPrice and PQuantity, TotalPrice. Write three functions:

product(): to initialize the value of the data members PNumber, PName, PPrice and PQuantity (using constructor concept).

calculate Total(): to calculate the TotalPrice (Total = price\* quantity).

display(): to print the values of the data members on the screen.

**Source code**

package java\_file;

public class \_7\_Product {

int PNumber;

String PName;

float PPrice;

int PQuantity;

float TotalPrice;

public static void main(String[] args) {

\_7\_Product obj=new \_7\_Product(777,"Laptop",80799.99f,2);

obj.calculateTotal();

obj.display();

}

\_7\_Product(int a, String b, float c, int d){

PNumber=a;

PName=b;

PPrice=c;

PQuantity=d;

}

void calculateTotal(){

TotalPrice=PPrice\*PQuantity;

System.out.println("Total Amount Payable is :: "+TotalPrice);

}

void display() {

System.out.println("\n\nYour Product Details....");

System.out.println("Product No. \t:: "+PNumber);

System.out.println("Product Name \t:: "+PName);

System.out.println("Product Price \t:: "+PPrice);

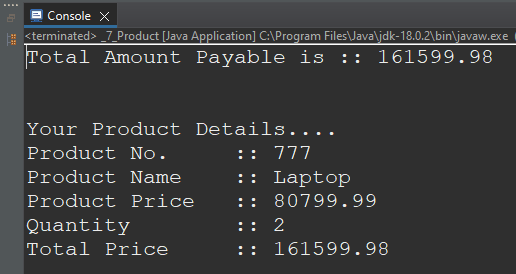
System.out.println("Quantity \t:: "+PQuantity);

System.out.println("Total Price \t:: "+TotalPrice);

}

}

**Output**



**Experiment No. 8 (a)**

**Aim**

Demonstrate the Concept of Multilevel Inheritance in Java.

**Source code**

package java\_file;

class Calculation {

void add(float x, float y) {

System.out.println("\nThe Addition of "+x+" and "+y+" is "+(x+y));

}

void sub(float x, float y) {

System.out.println("\nThe Subtraction of "+x+" and "+y+" is "+(x-y));

}

}

class NewCalculation1 extends Calculation{

void mul(float x, float y) {

System.out.println("\nThe Multiplication of "+x+" and "+y+" is "+(x\*y));

}

void div(float x, float y) {

System.out.println("\nThe Division of "+x+" and "+y+" is "+(x/y));

}

}

class NewCalculation2 extends NewCalculation1{

void mod(float x, float y) {

System.out.println("\nThe Moduls of "+x+" and "+y+" is "+(x+y));

}

}

public class \_8a\_Multilevel\_Inheritance {

public static void main(String[] args) {

NewCalculation2 obj=new NewCalculation2();

obj.add(12, 8);

obj.sub(45, 13);

obj.mul(13, 67);

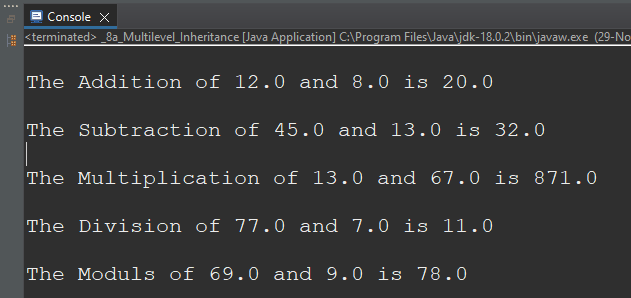
obj.div(77, 7);

obj.mod(69, 9);

}

}

**Output**



**Experiment No. 8 (b)**

**Aim**

Demonstrate the Concept of Hierarchical Inheritance in java.

**Source code**

package java\_file;

class Mobile\_Phone {

int model\_no;

String brand;

String colour;

float price;

int battery;

String type;

void call() {

System.out.println(brand+" "+type+" Making a Call !!!");

}

void message() {

System.out.println(brand+" "+type+" Sending Message !!!");

}

void camera() {

System.out.println(brand+" "+type+" Clicked Photo !!!");

}

void speaker() {

System.out.println(brand+" "+type+" Playing Sound !!!");

}

void radio() {

System.out.println(brand+" "+type+" Playing FM Radio !!!");

}

}

class KeypadPhone extends Mobile\_Phone {

float keypad\_size;

KeypadPhone(){

type="Keypad Phone";

brand="Nokia";

}

}

class SmartPhone extends Mobile\_Phone {

float display\_size;

String Stylus;

SmartPhone() {

type="Smart Phone";

brand="Apple";

}

void finger\_print\_scanner() {

System.out.println(brand+" "+type+" Scanning Finger !!!");

}

void face\_recognition() {

System.out.println(brand+" "+type+" Scanning Face !!!");

}

void games () {

System.out.println(brand+" "+type+" Playing Video Game !!!");

}

}

class FoldingPhone extends SmartPhone {

FoldingPhone() {

type="Folding Phone";

brand="Samsung";

}

void desktop\_mode() {

System.out.println(brand+" "+type+" in Desktop Mode !!!");

}

void Multiwindow() {

System.out.println(brand+" "+type+" using Multi-Window Function !!!");

}

}

public class \_8b\_Hierarchical\_Inheritance {

public static void main(String[] args) {

FoldingPhone samsung=new FoldingPhone();

SmartPhone apple=new SmartPhone();

KeypadPhone nokia=new KeypadPhone();

samsung.call();

samsung.desktop\_mode();

System.out.println();

apple.call();

apple.games();

System.out.println();

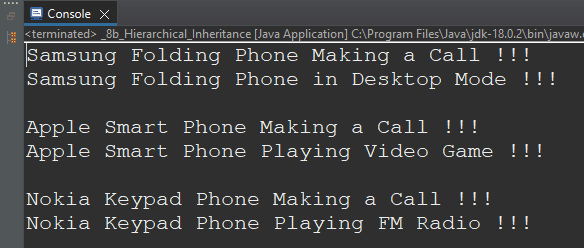
nokia.call();

nokia.radio();

}

}

**Output**



**Experiment No. 9**

**Aim**

Create a class 'Degree' having a method 'getDegree' that prints "I got a degree". It has two subclasses namely 'Undergraduate' and 'Postgraduate each having a method with the same name that prints "I am an Undergraduate" and "I am a Postgraduate" respectively. Call the 3 method by creating an object of the two child classes.

**Source code**

package java\_file;

class Degree {

void getDegree() {

System.out.println("I got a Degree.");

}

}

class Undergraduate extends Degree {

@Override

void getDegree() {

super.getDegree();

System.out.println("I am an Undergraduate.");

}

}

class Postgraduate extends Degree {

void getDegree() {

System.out.println("I am a Postgraduate.");

}

}

public class \_9\_Method\_overriding {

public static void main(String[] args) {

Undergraduate student=new Undergraduate();

Postgraduate student2=new Postgraduate();

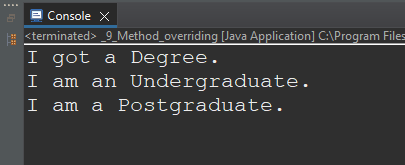
student.getDegree();

student2.getDegree();

}

}

**Output**



**Experiment No. 10**

**Aim**

Demonstrate the Concept of Abstraction in Java.

**Source code**

package java\_file;

abstract class Result {

abstract float percentage(float x[]);

abstract float sum(float x[]);

void display(float a[]) {

System.out.println("Your Scores...");

System.out.println("OOC \t::\t"+a[0]);

System.out.println("DBMS \t::\t"+a[1]);

System.out.println("IMP \t::\t"+a[2]);

System.out.println("DMF \t::\t"+a[3]);

System.out.println("CPS \t::\t"+a[4]);

System.out.println("\nYour Percentage is "+percentage(a));

}

}

class Working extends Result {

float percentage(float x[]) {

float per=(sum(x)/500)\*100;

return per;

}

float sum(float x[]) {

float temp=0;

for(int i=0;i<x.length;i++)

temp+=x[i];

return temp;

}

}

public class \_10\_Abstraction {

public static void main(String[] args) {

float arr[]= {91,84,65,50,77};

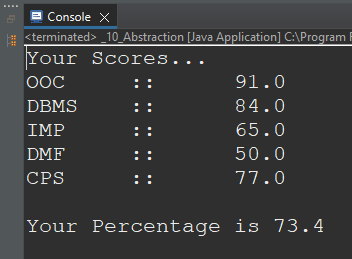
Working obj=new Working();

obj.display(arr);

}

}

**Output**



**Experiment No. 11**

**Aim**

Java program to calculate the area of rectangle using the concept of Encapsulation.

**Source code**

package java\_file;

class Area\_Rectangle {

private float length;

private float breadth;

private float Area;

private void Cal\_Area() {

Area=length\*breadth;

System.out.println("The Area of Rectangle is :: "+Area+" CM");

}

public void setLength\_Breadth(float length, float breadth) {

this.length = length;

this.breadth = breadth;

}

public float getLength() {

return length;

}

public float getBreadth() {

return breadth;

}

public void show\_Area() {

Cal\_Area();

}

}

public class \_11\_Encapsulation {

public static void main(String[] args) {

Area\_Rectangle obj=new Area\_Rectangle();

obj.setLength\_Breadth(17, 10);

System.out.println("Rectangle Details in CMs ...");

System.out.println("Length :: "+obj.getLength());

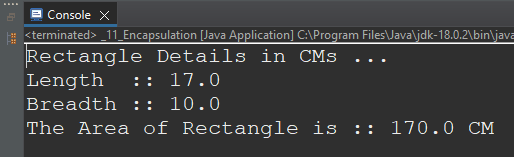
System.out.println("Breadth :: "+obj.getBreadth());

obj.show\_Area();

}

}

**Output**



**Experiment No. 12**

**Aim**

Java Program to demonstrate the concept of try, catch, and finally block in Exception Handling.

**Source code**

package java\_file;

import java.util.Scanner;

public class \_12\_Excepion\_Handling {

public static void main (String[] args) {

Scanner input=new Scanner (System.in);

System.out.println("Enter 2 numbers to perform Division ???");

int a=input.nextInt();

int b=input.nextInt();

try

{

int c=a/b;

System.out.println("The Quotient of division is :: "+c);

}

catch (ArithmeticException e)

{

System.out.println("Division by Zero is not Possible!!!!");

}

finally

{

System.out.println("\nRun the Code Again if Needed...");

System.out.println("The Program has Ended!!!");

}

}

}

**Output**

