***WINTER 17***

***NORMAL CODE***

1. Default access specifier

class test

{

int data=40; //default access

void show() // default access

{

System.out.println("Hello java");

}

}

public class test1{

public static void main(String args[]){

testobj=new test();

System.out.println(obj.data);

obj.show();

}

}

1. Write a program to accept number from command line and print

square root of the number.

class test1

{

public static void main(String args[])

{

intnum;

num= Integer.parseInt(args[0]);

doublesq=Math.sqrt(num);

System.out.println("square root of "+ num +" is +sq);

}}

1. Example (Program to raise an exception if the passwords do not match)

import java.io.\*;

classPasswordException extends Exception

{

PasswordException(String msg)

{

super(msg);

}

}

class PassCheck

{

public static void main(String args[]) throws IOException

{

BufferedReader bin=new BufferedReader(new

InputStreamReader(System.in));

try

{

System.out.println("Enter Password : ");

if(bin.readLine().equals("abc123"))

{

System.out.println("Authenticated ");

}

else

{

throw new PasswordException("Authentication failure");

}

}

catch(PasswordException e)

{

System.out.println(e);

}

}

}

1. Write a single program to implement inheritance and

polymorphism in java.

class Employee {

String name;

String address;

int number;

Employee(String name, String address, int number) {

System.out.println("Constructing an Employee");

this.name = name;

this.address = address;

this.number = number;

}

public void mailCheck() {

System.out.println("Mailing a check to " + this.name + " " +

this.address);

}

public String toString() {

return name + " " + address + " " + number;

}

public String getName() {

return name;

}

public String getAddress() {

return address;

}

public void setAddress(String newAddress) {

address = newAddress;

}

public intgetNumber() {

return number;

}

}

class Salary extends Employee {

private double salary;

Salary(String name, String address, int number, double salary) {

super(name, address, number);

setSalary(salary);

}

public void mailCheck() {

System.out.println("Within mailCheck of Salary class ");

System.out.println("Mailing check to " + getName()

+ " with salary " + salary);

}

public double getSalary() {

return salary;

}

public void setSalary(double newSalary) {

if(newSalary>= 0.0) {

salary = newSalary;

}

}

public double computePay() {

System.out.println("Computing salary pay for " + getName());

return salary/52;

}

}

public class Demo {

public static void main(String [] args) {

Salary s = new Salary("RAM", "Dadar", 3, 3600.00);

Employee e = new Salary("John ", "Thane", 2, 2400.00);

System.out.println("Call mailCheck using Salary reference --");

s.mailCheck();

System.out.println("\n Call mailCheck using Employee reference--");

e.mailCheck();

}

}

1. Super

Class Box

{

Box()

{

}

void show()

{

//definition of show

}

} //end of Box class

Class BoxWeight extends Box

{

BoxWeight()

{

}

void show() // method is overridden in derived

{

Super.show() // will call base class method

}

}

ORRRR

class A

{

int i;

A(int a, iont b)

{

i=a+b;

}

void add()

{

System.out.println(“sum of a and b=”+i);

}

}

class B extends A

{

int j;

B(int a,int b, int c)

{

super(a,b);

j=a+b+c;

}

void add()

{

super.add();

System.out.println(“Sum of a, b and c is:” +j);

}

}

1. This

Box(double width, double height, double depth)

{

this.width = width;

this.height = height;

this.depth = depth;

}

1. Write a program to print the following output:

1 1 1 1 1

2 2 2 2

3 3 3

4 4

5

class Sample

{

public static void main(String args[]){

inti,j,a=1;

for(i=5;i>=1;i--)

{

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

{

System.out.println(a+" ");

}

System.out.println(“\n”);

a++;

}

}

1. Method overriding

class Vehicle{

void run(){System.out.println("Vehicle is running");}

}

class Bike2 extends Vehicle{

void run()

{

System.out.println("Bike is running safely");

}

public static void main(String args[]){

Bike2 obj = new Bike2();

obj.run();

}

1. compareTo

String str1 = "Strings are immutable";

String str2 = "Strings are immutable";

String str3 = "Integers are not immutable";

int result = str1.compareTo( str2 );

System.out.println(result);

result = str2.compareTo( str3 );

System.out.println(result);

1. equalsIgoneCase

String str1 = "Strings are immutable";

String str2 = "Strings are immutable";

String str3 = "Integers are not immutable";

int result = str1.compareTo( str2 );

System.out.println(result);

result = str2.compareTo( str3 );

System.out.println(result);

1. create a package (user defined )

package myPack;

importjava.util.\*;

public class Myclass {

public void myMethod() {

System.out.println("Inside package");

}

}

Import it

import myPack.\*;

public class MyClassExample{

public static void main(String a[]) {

Myclass c= new Myclass();

c.myMethod();

}

}

1. Define a class person with data member as Aadharno, name, Panno implement concept of constructor overloading. Accept data for 5 object and print it.

import java.io.\*;

class Person {

intAadharno;

String name;

String Panno;

Person(intAadharno, String name, String Panno) {

this.Aadharno = Aadharno;

this.name = name;

this.Panno = Panno;

}

Person(intAadharno, String name) {

this.Aadharno = Aadharno;

this.name = name;

Panno = "Not Applicable";

}

void display() {

System.out.println("Aadharno is :"+Aadharno);

System.out.println("Name is: "+name);

System.out.println("Panno is :"+Panno);

}

public static void main(String ar[]) {

BufferedReaderbr = new

BufferedReader(newInputStreamReader(System.in));

Person p, p1, p2, p3, p4;

int a;

String n, pno;

try {

System.out.println("Enter Aadhar no");

a = Integer.parseInt(br.readLine());

System.out.println("Enter name");

n = br.readLine();

System.out.println("Enter panno");

pno = br.readLine();

p = new Person(a,n,pno);

System.out.println("Enter Aadhar no");

a = Integer.parseInt(br.readLine());

System.out.println("Enter name");

n = br.readLine();

System.out.println("Enter panno");

pno = br.readLine();

p1 = new Person(a,n,pno);

System.out.println("Enter Aadhar no");

a = Integer.parseInt(br.readLine());

System.out.println("Enter name");

n = br.readLine();

p2 = new Person(a,n);

System.out.println("Enter Aadhar no");

a = Integer.parseInt(br.readLine());

System.out.println("Enter name");

n = br.readLine();

p3 = new Person(a,n);

System.out.println("Enter Aadhar no");

a = Integer.parseInt(br.readLine());

n = br.readLine();

System.out.println("Enter panno");

pno = br.readLine();

p4 = new Person(a,n,pno);

p.display();

p1.display();

p2.display();

p3.display();

p4.display();

} catch(Exception e) {

System.out.println("Exception caught"+e);

}

}

}

1. Multiple inheritance

Eg: interface MyInterface{

int strength=60;

void method1();

}

class MyBaseClass {

String str;

MyBaseClass(String str) {

this.str = str;

}

public void display() {

System.out.println("Class: "+str);

}

}

public class MyClass extends MyBaseClass implements MyInterface

{

float total;

MyClass(String str, float t) {

super(str);

total = t;

}

public void method1() {

float avg = total/strength;

System.out.println("Avg is "+avg);

}

public static void main(String a[]) {

MyClass c = new MyClass("Fifth Sem",1300.0f);

c.display();

c.method1();

}

}

***APPLET***

1. Attempt any ONE of the following: Design an applet which display equals size three rectangle one below the other and fill them with orange, white and green color respectively.

import java.awt.\*;

importjava.applet.\*;

/\*

<applet code = DisplayRectangle.class height = 300 width =

300></applet>

\*/

public class DisplayRectangle extends Applet {

public void init() {

setBackground(Color.PINK);

}

public void paint(Graphics g) {

g.setColor(Color.ORANGE);

g.fillRect(40,40,40,30);

g.setColor(Color.WHITE);

g.fillRect(40,90,40,30);

g.setColor(Color.GREEN);

g.fillRect(40, 140,40,30);

}

}

1. Write an applet to accept Account No and balance in form of parameter and print message “low balance” if the balance is less than 500.

Program :

importjava.awt.\*;

importjava.applet.\*;

public class applet1 extends Applet

{

String accno="";

int balance=0;

public void init()

{

accno=getParameter("acno");

balance=Integer.parseInt(getParameter("bal"));

}

public void paint(Graphics g)

{

if(balance<500)

g.drawString(accno+": Low balance...",100,100);

else

g.drawString(accno+":sufficient balance...",100,100);

}

}

/\*<applet code=applet1 width=200 height=200>

<param name="acno" value="1001">

<param name="bal" value="200">

</applet>\*/

1. Write a applet program to set background with red colour and fore ground with blue colour.

import java.awt.\*;

import java.applet.\*;

public class applet2 extends Applet

{

String str="java programming";

public void init()

{

setBackground(Color.red);

setForeground(Color.blue);

}

public void paint(Graphics g)

{

g.drawString("welcome",100,100);

}

}

/\*<applet code=applet2 width=200 height=200>

</applet>\*/

***WINTER 18***

***NORMAL CODE***

1. Write a program to find greater number among two numbers using conditional operator.

class greater

{

public static void main(String args[])

{

int a,b;

int bignum;

a=100;

b=150;

bignum=(a>b?a:b); //conditional operator

System.out.println("Greater number between "+a+ " and "+b +" is =

"+bignum);

}

}

1. Write a program to print sum of even numbers from 1 to 20.

public class sum\_of\_even

{

public static void main(String[] args)

{

int sum=0;

for (int i=0; i<=20;i=i+2)

{

sum = sum+i;

}

System.out.println("Sum of these numbers: "+sum);

}

}

1. Final

class XYZ

{

final void demo(){

System.out.println("XYZ Class Method");

}

}

class ABC extends XYZ{

void demo()

{

System.out.println("ABC Class Method");

}

public static void main(String args[])

{

ABC obj= new ABC();

obj.demo();

}

}

The above program would throw a compilation error, however we can use the

parent class final method in sub class without any issues.

class XYZ

{

final void demo(){

System.out.println("XYZ Class Method");

}

}

class ABC extends XYZ

{

public static void main(String args[])

{

ABC obj= new ABC();

obj.demo();

}

}

FINAL CLASS

final class XYZ

{

}

class ABC extends XYZ

{

void demo()

{

System.out.println("My Method");

}

public static void main(String args[])

{

ABC obj= new ABC();

obj.demo();

}

}

1. Write a program to calculating area and perimeter of rectangle.

import java.io.\*;

import java.lang.\*;

class rect

{

public static void main(String args[]) throws Exception

{

DataInputStream dr=new DataInputStream(System.in);

int l,w;

System.out.println(“Enter length and width:”);

l=Integer.parseInt(dr.readLine());

w=Integer.parseInt(dr.readLine());

int a=l\*w;

System.out.println(“Area is=”+a);

int p=2(l+w);

System.out.println(“Area is=”+p);

}

}

1. Write a program to create a vector with seven elements as (10,30,50,20,40,10,20). Remove elements 3rd and 4th position. Insert new elements at 3rd position. Display original and current size of vector.

import java.util.\*;

public class VectorDemo

{

public static void main(String args[])

{

Vector v = new Vector();

v.addElement(new Integer(10));

v.addElement(new Integer(30));

v.addElement(new Integer(50));

v.addElement(new Integer(20));

v.addElement(new Integer(40));

v.addElement(new Integer(10));

v.addElement(new Integer(20));

System.out println(v.size()); // display original size

v.removeElementAt(2); // remove 3rd element

v.removeElementAt(3); // remove 4th element

v.insertElementAt(11,2) // new element inserted at 3rd position

System.out.println("Size of vector after insert delete operations: " + v.size());

}

}

1. Define a class ‘Book’ with data members bookid, bookname and price.Accept data for seven objects using Array of objects and display it.

import java.lang.\*;

import java.io.\*;

class Book

{

String bookname;

int bookid;

int price;

BufferedReader br=new BufferedReader(new InputStreamReader

(System.in));

void getdata()

{

try

{

System.out.println("Enter Book ID=");

bookid=Integer.parseInt(br.readLine());

System.out.println("Enter Book Name=");

bookname=br.readLine();

System.out.println("Enter Price=");

price=Integer.parseInt(br.readLine());

}

catch(Exception e)

{

System.out.println("Error");

}

}

void display()

{

System.out.println("Book ID="+bookid);

System.out.println("Book Name="+bookname);

System.out.println("Price="+price);

}

}

class bookdata

{

public static void main(String args[])

{

Book b[]=new Book[7];

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

{

b[i]=new Book();

}

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

{

b[i].getdata();

}

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

{

b[i].display();

}

}

}

1. Logical operator

public class Test

{

public static void main(String args[])

{

boolean a = true;

boolean b = false;

System.out.println("a && b = " + (a&&b));

System.out.println("a || b = " + (a||b) );

System.out.println("!(a && b) = " + !(a && b));

}

}

Output:

a && b = false

a || b = true

!(a && b) = true

***APPLET CODE***

1. Write a program to create an applet for displaying circle, rectangle and triangle one below the other and filled them with red, green and yellow respectively.

import java.awt.\*;

import java.applet.\*;

/\* <applet code="test.class" width=200 height=200>

</applet> \*/

public class test extends Applet

{

public void paint(Graphics g)

{

g.setColor(Color.RED);

g.fillOval(80,50,50,50);

g.setColor(Color.GREEN);

g.fillRect(50,120,100,50);

g.setColor(Color.YELLOW);

int x1[]={50, 100, 150, 50};

int y1[]={250, 200, 250, 250};

int n1=4;

g.fillPolygon(x1, y1, n1);

}

}

1. Design an applet which displays rectangle filled with blue colour and display message as “MSBTE EXAM” in red colour below it.

import java.awt.\*;

import java.applet.\*;

public class myapplet extends Applet

{

public void paint(Graphics g)

{

g.setColor(Color.blue);

g.fillRect(50,50,100,50);

g.setColor(Color.red);

g.drawString("MSBTE EXAM",60,120);

1}

}

/\*<applet code=myapplet height=300 width=300>

</applet>\*/

***SUMMER 18***

***NORMAL CODE***

1. Copy const

class Student6{

int id;

String name;

Student6(int i,String n)

{

id = i;

name = n;

}

Student6(Student6 s){

id = s.id;

name =s.name;

}

void display(){System.out.println(id+" "+name);}

public static void main(String args[]){

Student6 s1 = new Student6(111,"Karan");

Student6 s2 = new Student6(s1); //copy constructor called

s1.display();

s2.display();

}

}

1. Parameterized

class Student4{

int id;

String name;

Student4(int i,String n){

id = i;

name = n;

}

void display()

{ System.out.println(id+" "+name);}

public static void main(String args[])

{

Student4 s1 = new Student4(111,"Karan");

Student4 s2 = new Student4(222,"Aryan");

s1.display();

s2.display();

}

}

1. Default

class Bike1{

Bike1()

{

System.out.println("Bike is created");

}

public static void main(String args[]){

Bike1 b=new Bike1();

}

}

Const with no args

Such constructors does not have

any parameters. All the objects created using this type of constructors

has the same values for its datamembers.

Eg:

class Student {

int roll\_no;

String name;

Student() {

roll\_no = 50;

name="ABC";

}

void display() {

System.out.println("Roll no is: "+roll\_no);

System.out.println("Name is : "+name);

}

public static void main(String a[]) {

Student s = new Student();

s.display();

}

}

1. Write a java program to display all the odd numbers between 1 to 30 using for loop & if statement.

class OddNum

{

public static void main(String args[])

{

for(int i=1;i<=30;i++)

{

if(i%2 ==1)

{

System.out.print("Odd number :"+i +"\n");

}

}

}

}

1. Write a java program to extend interface assuming suitable data.

interface A

{

int x=20;

void display();

}

interface B extends A

{

int y=30;

void show();

}

class C implements B

{

public void display()

{

System.out.println("A's X="+x);

}

public void show()

{

System.out.println("A's X="+x);

System.out.println("B's Y="+y);

}

public static void main(String args[])

{

C obj=new C();

obj.display();

obj.show();

}

}

1. Write a java program to implement following functions of string:

(1) Calculate length of string

(2) Compare between strings

(3) Concatenating strings

class StringDemo

{

public static void main(String args[])

{

String str1="INDIA";

String str2="India";

String str3="My India";

String str4="India";

System.out.println("The length of string INDIA is "+str1.length()); //Length of string

System.out.println("Comparing String India and India is "+str2.equals(str4));

System.out.println("Comparing String INDIA and India is "+str1.equals(str2));

System.out.println("Comparing String INDIA and India with equalsIgnoreCase is"+str1.equalsIgnoreCase(str2));

String str5="I Love";

System.out.println("Result of concatinating of string is "+str5.concat(str3));

}

}

1. Notify,wait,resume,sleep

class sus extends Thread implements Runnable

{

static Thread th;

float rad,r;

public sus()

{

th= new Thread();

th.start();

}

public void op()

{

System.out.println("\nThis is OP");

if(rad==0)

{

System.out.println("Waiting for input radius");

try

{

wait();

}

catch(Exception ex)

{

}

}

}

public void ip()

{

System.out.println("\nThis is IP");

r=7;

rad= r;

System.out.println(rad);

System.out.println("Area = "+3.14\*rad\*rad);

notify();

}

public static void main(String arp[])

{

try{

sus s1 = new sus();

System.out.println("\nReady to go");

Thread.sleep(2000);

System.out.println("\nI am resuming");

th.suspend();

Thread.sleep(2000);

th.resume();

System.out.println("\nI am resumed once again");

s1.op();

s1.ip();

s1.op();

}

catch(Exception e)

{}

}

}

1. Write a java program to implement multilevel inheritance with 4 levels of hierarchy.

class emp

{

int empid;

String ename;

emp(int id, String nm)

{

empid=id;

ename=nm;

}

}

class work\_profile extends emp

{

String dept;

String job;

work\_profile(int id, String nm, String dpt, String j1)

{

super(id,nm);

dept=dpt;

job=j1;

}

}

class salary\_details extends work\_profile

{

int basic\_salary;

salary\_details(int id, String nm, String dpt, String j1,int bs)

{

super(id,nm,dpt,j1);

basic\_salary=bs;

}

double calc()

{

double gs;

gs=basic\_salary+(basic\_salary\*0.4)+(basic\_salary\*0.1);

return(gs);

}

}

class salary\_calc extends salary\_details

{

salary\_calc(int id, String nm, String dpt, String j1,int bs)

{

super(id,nm,dpt,j1,bs);

}

public static void main(String args[])

{

salary\_calc e1=new salary\_calc(101,"abc","Sales","clerk",5000);

double gross\_salary=e1.calc();

System.out.println("Empid :"+e1.empid);

System.out.println("Emp name :"+e1.ename);

System.out.println("Department :"+e1.dept);

System.out.println("Job :"+e1.job);

System.out.println("BAsic Salary :"+e1.basic\_salary);

System.out.println("Gross salary :"+gross\_salary);

}

}

1. Write a java program to copy the content of the file “file1.txt” into new file “file2.txt”.

import java.io.\*;

class filecopy

{

public static void main(String args[]) throws IOException

{

FileReader fr=new FileReader("file1.txt");

FileWriter fo=new FileWriter("file2.txt");

int ch;

try

{

while((ch=fr.read())!=-1)

{

fo.write(ch);

}

fr.close();

fo.close();

}

finally

{

if(fr!=null)

fr.close();

if(fo!=null)

fo.close();

}

}

}

1. Public access specifier

class Hello

{

public int a=20;

public void show()

{

System.out.println("Hello java");

}

}

public class Demo

{

public static void main(String args[])

{

Hello obj=new Hello();

System.out.println(obj.a);

obj.show();

}

}

private access specifier :

class Hello

{

private int a=20;

private void show()

{

System.out.println("Hello java");

}

}

public class Demo

{

public static void main(String args[])

{

Hello obj=new Hello();

System.out.println(obj.a); //Compile Time Error, you can't access private data

obj.show(); //Compile Time Error, you can't access private methods

}

}

protected access specifier :

// save A.java

package pack1;

public class A

{

protected void show()

{

System.out.println("Hello Java");

}

}

//save B.java

package pack2;

import pack1.\*;

class B extends A

{

public static void main(String args[]){

B obj = new B();

obj.show();

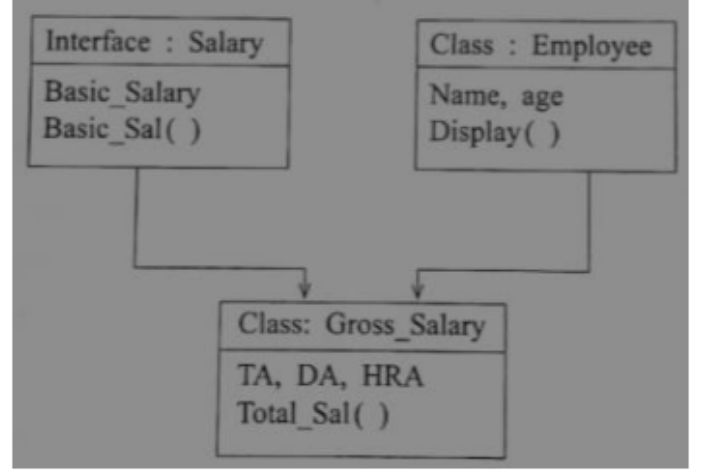
}

}

***APPLET CODE***

***WINTER 19***

***NORMAL CODE***



interface Salary

{

double Basic Salary=10000.0;

void Basic Sal();

}

class Employee

{

String Name;

int age;

Employee(String n, int b)

{

Name=n;

age=b;

}

void Display()

{

System.out.println("Name of Employee

:"+Name);

System.out.println("Age of Employee :"+age);

}

}

class Gross\_Salary extends Employee implements Salary

{

double HRA,TA,DA;

Gross\_Salary(String n, int b, double h,double t,double d)

{

super(n,b);

HRA=h;

TA=t;

DA=d;

}

public void Basic\_Sal()

{

System.out.println("Basic Salary

:"+Basic\_Salary);

}

void Total\_Sal()

{

Display();

Basic\_Sal();

double Total\_Sal=Basic\_Salary + TA + DA +

HRA;

System.out.println("Total Salary :"+Total\_Sal);

}

}

class EmpDetails

{ public static void main(String args[])

{ Gross\_Salary s=new

Gross\_Salary("Sachin",20,1000,2000,7000);

s.Total\_Sal();

}

}

2. Write a program to perform following task

(i) Create a text file and store data in it.

(ii) Count number of lines and words in that file.

import java.util.\*;

import java.io.\*;

class Model6B

{

public static void main(String[] args) throws Exception

{

int lineCount=0, wordCount=0;

String line = "";

BufferedReader br1 = new BufferedReader(new

InputStreamReader(System.in));

FileWriter fw = new FileWriter("Sample.txt");

//create text file for writing

System.out.println("Enter data to be inserted in

file: ");

String fileData = br1.readLine();

fw.write(fileData);

fw.close();

BufferedReader br = new BufferedReader(new

FileReader("Sample.txt"));

while ((line = br.readLine()) != null)

{

lineCount++; // no of lines count

String[] words = line.split(" ");

wordCount = wordCount + words.length;

// no of words count

}

System.out.println("Number of lines is : " +

lineCount);

System.out.println("Number of words is : " +

wordCount);

}

3.. Write a program to create a vector with five elements as (5,

15, 25, 35, 45). Insert new element at 2nd position. Remove 1st

and 4th element from vector.

import java.util.\*;

class VectorDemo

{

public static void main(String[] args)

{

Vector v = new Vector();

v.addElement(new Integer(5));

v.addElement(new Integer(15));

v.addElement(new Integer(25));

v.addElement(new Integer(35));

v.addElement(new Integer(45));

System.out.println("Original array elements are

");

for(int i=0;i<v.size();i++)

{

System.out.println(v.elementAt(i));

}

v.insertElementAt(new Integer(20),1); // insert

new element at 2nd position

v.removeElementAt(0);

//remove first element

v.removeElementAt(3);

//remove fourth element

System.out.println("Array elements after insert

and remove operation ");

for(int i=0;i<v.size();i++)

{

System.out.println(v.elementAt(i));

}}}

4.. Define a class circle having data members pi and radius.

Initialize and display values of data members also calculate

area of circle and display it.

Class abc

{

float pi,radius;

abc(float p, float r)

{

pi=p;

radius=r;

}

void area()

{

float ar=pi\*radius\*radius;

System.out.println("Area="+ar);

}

void display()

{

System.out.println("Pi="+pi);

System.out.println("Radius="+radius);

} }

class area

{

public static void main(String args[])

{

abc a=new abc(3.14f,5.0f);

a.display();

a.area();

}

}

***APPLET CODE***

***SUMMER 19***

***NORMAL CODE***

1. Dynamic method dispatch

class A

{

void m1()

{

System.out.println("Inside A's m1 method");

}

}

class B extends A

{

// overriding m1()

void m1()

{

System.out.println("Inside B's m1 method");

}

}

class C extends A

{

// overriding m1()

void m1()

{

System.out.println("Inside C's m1 method");

}

}

// Driver class

class Dispatch

{

public static void main(String args[])

{

// object of type A

A a = new A();

// object of type B

B b = new B();

// object of type C

C c = new C();

// obtain a reference of type A

A ref;

// ref refers to an A object

ref = a;

// calling A's version of m1()

ref.m1();

// now ref refers to a B object

ref = b;

// calling B's version of m1()

ref.m1();

// now ref refers to a C object

ref = c;

// calling C's version of m1()

ref.m1();

}

}

1. instanceOf

class Simple1{

public static void main(String args[]){

Simple1 s=new Simple1();

System.out.println(sinstanceofSimple1);//true

}

}

1. dot

this.name=”john”; where name is a instance variable referenced by

‘this’ keyword

c.getdata(); where getdata() is a method invoked on object ‘c’.

4. Define a class student with int id and string name as data

members and a method void SetData ( ). Accept and display the

data for five students.

import java.io.\*;

class student

{

int id;

String name;

BufferedReader br = new BufferedReader(new

InputStreamReader(System.in));

void SetData()

{

try

{

System.out.println("enter id and name for student");

id=Integer.parseInt(br.readLine());

name=br.readLine();

}

catch(Exception ex)

{}

}

void display()

{

System.out.println("The id is " + id + " and the name is "+ name);

}

public static void main(String are[])

{

student[] arr;

arr = new student[5];

int i;

for(i=0;i<5;i++)

{

arr[i] = new student();

}

for(i=0;i<5;i++)

{

arr[i].SetData();

}

for(i=0;i<5;i++)

{

arr[i].display();

}

}

}

***APPLET CODE***

***WINTER 22***

***NORMAL CODE***

1.. Design an applet to perform all arithmetic operations and display the result by using labels. textboxes and buttons.

import java.awt.\*;

import java.awt.event.\*;

public class sample extends Frame implements ActionListener {

Label l1, l2,l3;

TextField tf1, tf2, tf3;

Button b1, b2, b3, b4;

sample() {

l1=new Lable(“First No.”);

l1.setBounds(10, 10, 50, 20);

tf1 = new TextField();

tf1.setBounds(50, 50, 150, 20);

l2=new Lable(“Second No.”);

l2.setBounds(10, 60, 50, 20);

tf2 = new TextField();

tf2.setBounds(50, 100, 150, 20);

l3=new Lable(“Result”);

l3.setBounds(10, 110, 150, 20);

tf3 = new TextField();

tf3.setBounds(50, 150, 150, 20);

tf3.setEditable(false);

b1 = new Button("+");

b1.setBounds(50, 200, 50, 50);

b2 = new Button("-");

b2.setBounds(120,200,50,50);

b3 = new Button("\*");

b3.setBounds(220, 200, 50, 50);

b4 = new Button("/");

b4.setBounds(320,200,50,50);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

add(tf1);

add(tf2);

add(tf3);

add(b1);

add(b2);

add(b3);

add(b4);

setSize(400,400);

setLayout(null);

setVisible(true);

}

public void actionPerformed(ActionEvent e) {

String s1 = tf1.getText();

String s2 = tf2.getText();

int a = Integer.parseInt(s1);

int b = Integer.parseInt(s2);

int c = 0;

if (e.getSource() == b1){

c = a + b;

}

else if (e.getSource() == b2){

c = a - b;

else if (e.getSource() == b3){

c = a \* b;

else if (e.getSource() == b4){

c = a / b;

}

String result = String.valueOf(c);

tf3.setText(result);

}

public static void main(String[] args) {

new sample();

}

}

2.. Write a program to check whether the string provided by the user is palindrome or not.

1. **public** **class** PalindromeString
2. {
3. **public** **static** **void** main(String[] args) {
4. String string = "Kayak";
5. **boolean** flag = **true**;
7. //Converts the given string into lowercase
8. string = string.toLowerCase();
10. //Iterate the string forward and backward, compare one character at a time
11. //till middle of the string is reached
12. **for**(**int** i = 0; i < string.length()/2; i++){
13. **if**(string.charAt(i) != string.charAt(string.length()-i-1)){
14. flag = **false**;
15. **break**;
16. }
17. }
18. **if**(flag)
19. System.out.println("Given string is palindrome");
20. **else**
21. System.out.println("Given string is not a palindrome");
22. }
23. }

3.. Define an exception called 'No Match Exception' that is thrown when the

passward accepted is not equal to "MSBTE'. Write the program.

import java.io.\*;

class NoMatchException extends Exception

{

NoMatchException(String s)

{

super(s);

}

}

class test1

{

public static void main(String args[]) throws IOException

{

BufferedReader br= new BufferedReader(new InputStreamReader(System.in) );

System.out.println("Enter a word:");

String str= br.readLine();

try

{

import java.io.\*;

class NoMatchException extends Exception

{

NoMatchException(String s)

{

super(s);

}

}

class test1

{

public static void main(String args[]) throws IOException

{

BufferedReader br= new BufferedReader(new InputStreamReader(System.in) );

System.out.println("Enter a word:");

String str= br.readLine();

try

{

4.. Write a program to create a class 'salary with data members empid', ‘name'

and ‘basicsalary'. Write an interface 'Allowance’ which stores rates of

calculation for da as 90% of basic salary, hra as 10% of basic salary and pf as

8.33% of basic salary. Include a method to calculate net salary and display it.

interface allowance

{

double da=0.9\*basicsalary;

double hra=0.1\*basicsalary;

double pf=0.0833\*basicsalary;

void netSalary();

}

class Salary

{

int empid;

String name;

float basicsalary;

Salary(int i, String n, float b)

{

empid=I;

name=n;

basicsalary =b;

}

void display()

{

System.out.println("Empid of Emplyee="+empid);

System.out.println("Name of Employee="+name);

System.out.println("Basic Salary of Employee="+ basicsalary);

}

}

class net\_salary extends salary implements allowance

{

float ta;

net\_salary(int i, String n, float b, float t)

{

super(i,n,b);

ta=t;

}

void disp()

{

display();

System.out.println("da of Employee="+da);

}

public void netsalary()

{

double net\_sal=basicsalary+ta+hra+da;

System.out.println("netSalary of Employee="+net\_sal);

}

}

class Empdetail

{

public static void main(String args[])

{

net\_salary s=new net\_salary(11, “abcd”, 50000);

s.disp();

s.netsalary();

}

}

5.. Develop a program to create a class ‘Book’ having data members author, title

and price. Derive a class 'Booklnfo' having data member 'stock position’ andmethod to initialize and display the information for three objects.

class Book

{

String author, title, publisher;

Book(String a, String t, String p)

{

author = a;

title = t;

publisher = p;

}

}

class BookInfo extends Book

{

float price;

int stock\_position;

BookInfo(String a, String t, String p, float amt, int s)

{

super(a, t, p);

price = amt;

stock\_position = s;

}

void show()

{

System.out.println("Book Details:");

System.out.println("Title: " + title);

System.out.println("Author: " + author);

System.out.println("Publisher: " + publisher);

System.out.println("Price: " + price);

System.out.println("Stock Available: " + stock\_position);

}

}

class Exp6\_1

{

public static void main(String[] args)

{

BookInfo ob1 = new BookInfo("Herbert Schildt", "Complete Reference", "ABC

Publication", 359.50F,10);

BookInfo ob2 = new BookInfo("Ulman", "system programming", "XYZ Publication",

359.50F, 20);

BookInfo ob3 = new BookInfo("Pressman", "Software Engg", "Pearson Publication",

879.50F, 15);

ob1.show();

ob2.show();

ob3.show();

}

}

6.. Write a program to add 2 integer, 2 string and 2 float values in a vector.

Remove the element specified by the user and display the list.

import java.io.\*;

import java.lang.\*;

import java.util.\*;

class vector2

{

public static void main(String args[])

{

vector v=new vector();

Integer s1=new Integer(1);

Integer s2=new Integer(2);

String s3=new String("fy");

String s4=new String("sy");

Float s7=new Float(1.1f);

Float s8=new Float(1.2f);

v.addElement(s1);

v.addElement(s2);

v.addElement(s3);

v.addElement(s4);

v.addElement(s7);

v.addElement(s8);

System.out.println(v);

v.removeElement(s2);

v.removeElementAt(4);

System.out.println(v);

}

}

7..reverse

public class ReverseNumberExample1

{ public static void main(String[] args)

{

int number = 987654, reverse =0;

while(number !=0)

{

int remainder = number % 10;

reverse = reverse \* 10 + remainder;

number = number/10;

}

System.out.printtln(“The reverse of the given number is: “ + reverse);

} }

8.. Define a class employee with data members 'empid , name and salary.

Accept data for three objects and display it

class employee

{

int empid;

String name;

double salary;

void getdata()

{

BufferedReader obj = new BufferedReader (new InputStreamReader(System.in));

System.out.print("Enter Emp number : ");

empid=Integer.parseInt(obj.readLine());

System.out.print("Enter Emp Name : ");

name=obj.readLine();

System.out.print("Enter Emp Salary : ");

salary=Double.parseDouble(obj.readLine());

}

void show()

{

System.out.println("Emp ID : " + empid);

System.out.println(“Name : " + name);

System.out.println(“Salary : " + salary);

}

}

classEmpDetails

{

public static void main(String args[])

{

employee e[] = new employee[3];

for(inti=0; i<3; i++)

{

e[i] = new employee(); e[i].getdata();

}

System.out.println(" Employee Details are : ");

for(inti=0; i<3; i++)

e[i].show();

}

}

8..prime

class PrimeExample

{

public static void main(String args[]){

int i,m=0,flag=0;

int n=7;//it is the number to be checked

m=n/2;

if(n==0||n==1){

System.out.println(n+" is not prime number");

}else{

for(i=2;i<=m;i++){

if(n%i==0){

System.out.println(n+" is not prime number");

flag=1;

break;

}

}

if(flag==0) { System.out.println(n+" is prime number"); }

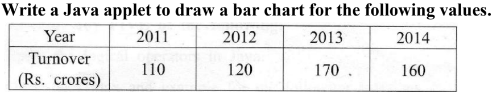
}//end of else

}

}

***SUMMER 22***

***NORMAL CODE***

1. 

import java.awt.\*;

import java.applet.\*;

/\* <applet code=BarChart width=400 height=400>

<param name=c1 value=110>

<param name=c2 value=120>

<param name=c3 value=170>

<param name=c4 value=160>

<param name=label1 value=2011>

<param name=label2 value=2012>

<param name=label3 value=2013>

<param name=label4 value=2014>

<param name=Columns value=4>

</applet>

\*/

public class BarChart extends Applet

{

int n=0;

String label[];

int value[];

public void init() {

setBackground(Color.yellow);

try {

int n = Integer.parseInt(getParameter("Columns"));

label = new String[n];

value = new int[n];

label[0] = getParameter("label1");

label[1] = getParameter("label2");

label[2] = getParameter("label3");

label[3] = getParameter("label4");

value[0] = Integer.parseInt(getParameter("c1"));

value[1] = Integer.parseInt(getParameter("c2"));

value[2] = Integer.parseInt(getParameter("c3"));

value[3] = Integer.parseInt(getParameter("c4"));

}

catch(NumberFormatException e){}

}

public void paint(Graphics g)

{

for(int i=0;i<4;i++) {

g.setColor(Color.black);

g.drawString(label[i],20,i\*50+30);

g.setColor(Color.red);

g.fillRect(50,i\*50+10,value[i],40);

}}}

1. Write a java program to sort an 1-d array in ascending order using bubble-sort.

{

int arr[] ={3,60,35,2,45,320,5};

System.out.println("Array Before Bubble Sort");

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

{

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

}

System.out.println();

int n = arr.length;

int temp = 0;

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

{

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

{

if(arr[j-1] >arr[j])

{

//swap elements

temp = arr[j-1];

arr[j-1] = arr[j];

arr[j] = temp;

}

}

}

System.out.println("Array After Bubble Sort");

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

{

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

}

}

}

1. Conditional operator

expression1 ? expression2:expression3;

The above syntax means that if the value given in Expression1 is true,

then Expression2 will be evaluated; otherwise, expression3 will be

evaluated.

Example

class test

{

public static void main(String[] args) {

String result;

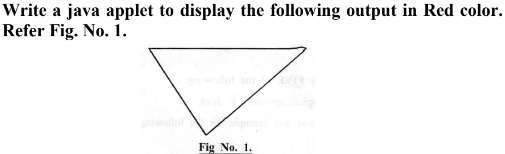
int a = 6, b = 12;

result = (a==b ? "equal":"Not equal");

System.out.println("Both are "+result);

}

}

1. 

import java.awt.\*;

import java.applet.\*;

public class myapplet extends Applet

{

public void paint(Graphics g)

{

int x[]={10,200,70};

int y[]={10,10,100};

g.setColor(Color.red);

g.drawPolygon(x,y,3);

}

}

/\*<applet code=myapplet height=400 width=400>

</applet>\*/

1. Single inheritance

Class A

{

void display()

{

System.out.println(“In Parent class A”);

}

}

class B extends A //derived class B from A

{

void show()

{

System.out.println(“In child class B”);

}

public static void main(String args[])

{

B b= new B();

b.display(); //super class method call

b.show(); // sub class method call

}

}

Multilevel

Class A

{

Void display()

{

System.out.println(“In Parent class A”);

}

}

class B extends A //derived class B from A

{

void show()

{

System.out.println(“In child class B”);

}

}

class C extends B //derived class C from B

{

public void print()

{

System.out.println(“In derived from derived class C”);

}

public static void main(String args[])

{

C c= new C();

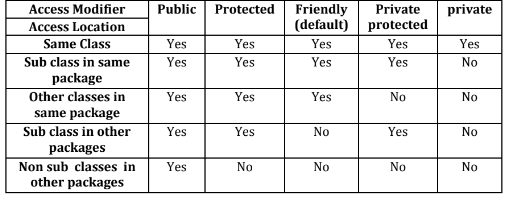
c.display(); //super class method call

c.show(); // sub class method call

c.print(); //sub-sub class method call

}

}

1. 
2. Write a Java Program to find out the even numbers from 1 to 100 using for loop.

class test

{

public static void main(String args[])

{

System.out.println("Even numbers from 1 to 100 :");

for(int i=1;i<=100; i++)

{

if(i%2==0)

System.out.print(i+" ");

}

}

}

***WINTER 23***

***NORMAL CODE***

1. Write a program to draw a chessboard in Java Applet.

import java.applet.\*;

import java.awt.\*;

/\*<applet code="Chess" width=600 height=600>

</applet>\*/

// Extends Applet Class

public class Chess extends Applet

{

static int N = 10;

// Use paint() method

public void paint(Graphics g)

{

int x, y;

for (int row = 0; row & lt; N; row++) {

for (int col = 0; col & lt; N; col++) {

// Set x coordinates of rectangle

// by 20 times

x = row \* 20;

// Set y coordinates of rectangle

// by 20 times

y = col \* 20;

// Check whether row and column are in even position

// If it is true set Black color

if ((row % 2 == 0) == (col % 2 == 0))

g.setColor(Color.BLACK);

else

g.setColor(Color.WHITE);

// Create a rectangle with

// length and breadth of 20

g.fillRect(x, y, 20, 20);

}

}

}

}

1. Write a program to 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 user.

import java.util.\*;

// User Defined Complex class

class Complex

{

// Declaring variables

int real, imaginary;

// Empty Constructor

Complex()

{

}

// Constructor to accept

// real and imaginary part

Complex(int tempReal, int tempImaginary)

{

real = tempReal;

imaginary = tempImaginary;

}

// Defining addComp() method

// for adding two complex number

Complex addComp(Complex C1, Complex C2)

{

// creating temporary variable

Complex temp = new Complex();

// adding real part of complex numbers

temp.real = C1.real + C2.real;

// adding Imaginary part of complex numbers

temp.imaginary = C1.imaginary + C2.imaginary;

// returning the sum

return temp;

}

// Defining subtractComp() method

// for subtracting two complex number

Complex subtractComp(Complex C1, Complex C2)

{

// creating temporary variable

Complex temp = new Complex();

// subtracting real part of complex numbers

temp.real = C1.real - C2.real;

// subtracting Imaginary part of complex numbers

temp.imaginary = C1.imaginary - C2.imaginary;

// returning the difference

return temp;

}

Complex productComp(Complex C1, Complex C2)

{

// creating temporary variable

Complex temp = new Complex();

// product of of complex numbers

//(a + ib) (c + id)= (ac - bd) + i(ad + bc).

temp.real = ((C1.real\*C2.real)-(C1.imaginary\*C2.imaginary));

temp.imaginary = ((C1.real\*C2.imaginary) + (C1.imaginary\*C2.real));

// returning the difference

return temp;

}

// Function for printing complex number

void printComplexNumber()

{

System.out.println("Complex number: " + real + " + " + imaginary + "i");

}

}

// Main Class

public class Main

{

// Main function

public static void main(String[] args)

{

// First Complex number

Complex C1 = new Complex(3, 2);

// printing first complex number

C1.printComplexNumber();

// Second Complex number

Complex C2 = new Complex(9, 5);

// printing second complex number

C2.printComplexNumber();

// for Storing the sum

Complex C3 = new Complex();

// calling addComp() method

C3 = C3.addComp(C1, C2);

// printing the sum

System.out.print("Sum of ");

C3.printComplexNumber();

// calling subtractComp() method

C3 = C3.subtractComp(C1, C2);

// printing the difference

System.out.print("Difference of ");

C3.printComplexNumber();

// calling productComp() method

C3 = C3.productComp(C1, C2);

// printing the product

System.out.print("product of ");

C3.printComplexNumber();

}

}

1. Write a program that throws an exception called "NoMatchException" when a string is not equal to "India".

import java.io.\*;

class NoMatchException extends Exception

{

private String str;

NoMatchException(String str1)

{

str=str1;

}

public String toString()

{

return "NoMatchException --> String is not India and string is "+str;

}

}

class Main

{

public static void main(String args[ ])

{

String str1= new String("India");

String str2= new String("Australlia");

try

{

if(str1.equals("India"))

System.out.println(" String is : "+str1);

else

throw new NoMatchException(str1);

if(str2.equals("India"))

System.out.println("\n String is : "+str2);

else

throw new NoMatchException(str2);

}

catch(NoMatchException e)

{

System.out.println("\nCaught ...."+e);

}

}

}

1. Develop and Interest Interface which contains Simple Interest and Compound Interest methods and static final field of rate 25%. Write a class to implement those methods.

import java.util.Scanner;

import static java.lang.Math.pow;

interface Interest

{

int roi=25;

public void simpleInterest(float principle,float time);

public void compoundInterest(float principle,float time);

}

public class InterestTest implements Interest

{

public void simpleInterest(float principle,float time)

{

float si = (principle\*roi\*time)/100;

System.out.println("Simple interested calculate by program is : " + si);

}

public void compoundInterest(float principle,float time)

{

double ci = principle \* (Math.pow((1.0 +(roi/100)), time)) - principle;

System.out.println("Compound interested calculate by program is : " + ci);

}

public static void main(String args[])

{

InterestTest i1 = new InterestTest();

i1.simpleInterest(1000,2);

i1.compoundInterest(1000,2);

}

}

1. FONT

import java.awt.\*;

import java.applet.\*;

public class Shapes extends Applet

{

Font f,f1;

String s,msg;

String fname;

String ffamily;

int size;

int style;

public void init()

{

f= new Font("times new roman",Font.ITALIC,20);

setFont(f);

msg="is interesting";

s="java programming";

fname=f.getFontName();

ffamily=f.getFamily();

size=f.getSize();

style=f.getStyle();

String f1=f.getName();

}

public void paint(Graphics g)

{

g.drawString("font name"+fname,60,44);

g.drawString("font family"+ffamily,60,77);

g.drawString("font size "+size,60,99);

g.drawString("fontstyle "+style,60,150);

g.drawString("fontname "+f1,60,190);

}

}

/\*<applet code=Shapes.class height=300 width=300></applet>\*/

1. HIERARCHICAL INHERITANCE

public void methodA()

{

System.out.println("method of Class A");

}

}

class B extends A

{

public void methodB()

{

System.out.println("method of Class B");

}

}

class C extends A

{

public void methodC()

{

System.out.println("method of Class C");

}

}

class D extends A

{

public void methodD()

{

System.out.println("method of Class D");

}

}

class JavaExample

{

public static void main(String args[])

{

B obj1 = new B();

C obj2 = new C();

D obj3 = new D();

//All classes can access the method of class A

obj1.methodA();

obj2.methodA();

obj3.methodA();

}

}

1. Write a program which displays functioning of ATM machine, (Hint: Withdraw, Deposit, Check Balance and Exit)

import java.util.Scanner;

public class ATM\_Transaction

{

public static void main(String args[] )

{

int balance = 5000, withdraw, deposit;

Scanner s = new Scanner(System.in);

while(true)

{

System.out.println("Automated Teller Machine");

System.out.println("Choose 1 for Withdraw");

System.out.println("Choose 2 for Deposit");

System.out.println("Choose 3 for Check Balance");

System.out.println("Choose 4 for EXIT");

System.out.print("Choose the operation you want to perform:");

int n = s.nextInt();

switch(n)

{

case 1:

System.out.print("Enter money to be withdrawn:");

withdraw = s.nextInt();

if(balance >= withdraw)

{

balance = balance - withdraw;

System.out.println("Please collect your money");

}

else

{

System.out.println("Insufficient Balance");

}

System.out.println("");

break;

case 2:

System.out.print("Enter money to be deposited:");

deposit = s.nextInt();

balance = balance + deposit;

System.out.println("Your Money has been successfully depsited");

System.out.println("");

break;

case 3:

System.out.println("Balance : "+balance);

System.out.println("");

break;

case 4:

System.exit(0);

}

}

}

1. Write a program lo display ASCII value of a number 9.

public class asciivalue

{

public static void main(String args[])

{

// Character whose ASCII is to be computed

char ch = '9';

// Creating a new variable of type int and assigning the character value.

int ascii = ch;

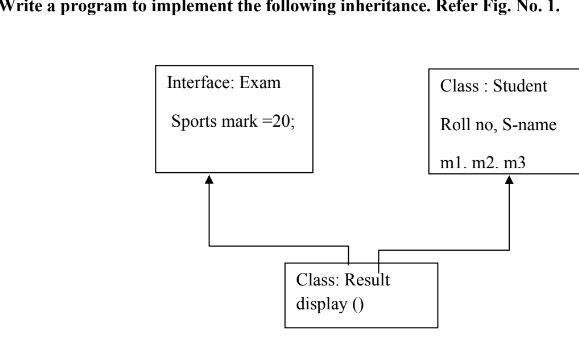
// Printing the ASCII value of above character

System.out.println("The ASCII value of " + ch+ " is: " + ascii);

}

}

***SUMMER 23***

1. 

interface Exam

{

int sports\_mark=20;

}

class Student

{

String S-name;

int Roll\_no, m1, m2, m3;

Student(String n, int a, int b, int c, int d)

{

S-name = n;

Roll\_no = a;

m1 = b;

m2 = c;

m3 = d;

}

void showdata()

{

System.out.println("Name of student :"+S-name);

System.out.println("Roll no. of the students :"+Roll\_no);

System.out.println(“Marks of subject 1:”+m1);

System.out.println(“Marks of subject 2:”+m2);

System.out.println(“Marks of subject 3:”+m3);

}

}

class Result extends Student implements Exam

{

Result(String n, int a, int b, int c, int d)

{

super(n, a, b, c, d );

}

void dispaly()

{

super.showdata();

int total=(m1+m2+m3);

float result=(total+Sports\_mark)/total\*100;

System.out.println(“result of student is:”+result);

}

}

class studentsDetails

{

public static void main(String args[])

{

Result r=new Result("Sachin",14, 78, 85, 97);

r.display();

}

}

1. Write a program to copy all elements of one array into another array.

public class CopyArray {

public static void main(String[] args)

{

int [] arr1 = new int [] {1, 2, 3, 4, 5};

int arr2[] = new int[arr1.length];

for (int i = 0; i < arr1.length; i++)

{

arr2[i] = arr1[i];

}

System.out.println("Elements of original array: ");

for (int i = 0; i < arr1.length; i++)

{

System.out.print(arr1[i] + " ");

}

System.out.println();

System.out.println("Elements of new array: ");

for (int i = 0; i < arr2.length; i++)

{

System.out.print(arr2[i] + " ");

}}}

1. Write an applet program for following graphics method.
2. Drawoval ( ) ii) Drawline ( )

import java.awt.\*;

import java.applet.\*;

public class CirSqr extends Applet

{

public void paint(Graphics g)

{

g.drawOval(70,30,100,100);

g.drawRect(90,50,60,60);

}

}

/\*<applet code="CirSqr.class" height=200 width=200>

</applet> \*/

1. Write a program to print all the Armstrong numbers from 0 to 999

class ArmstrongWhile

{

public static void main(String[] arg)

{

int i=1,a,arm,n,temp;

System.out.println("Armstrong numbers between 0 to 999 are");

while(i<500)

{

n=i;

arm=0;

while(n>0)

{

a=n%10;

arm=arm+(a\*a\*a);

n=n/10;

}

if(arm==i)

System.out.println(i);

i++;

}

}

}