JOURNAL 3

1.

import java.util.\*; class room

{

Scanner s=new Scanner(System.in);

int rno;

String rtype,rarea,acmachine; void setdata()

{

System.out.println("enter room no="); rno=s.nextInt();

System.out.println("enter room area=");

rarea=s.next();

System.out.println("enter room type=");

rtype=s.next();

System.out.println("enter there is AC machine or not=");

acmachine=s.next();

}

void displaydata()

{

System.out.println("room no="+rno);

System.out.println("room area="+rarea);

System.out.println("room type="+rtype);

System.out.println("AC machine="+acmachine);

}

}

public class roomdemo

{

public static void main(String args[])

{

room r=new room();

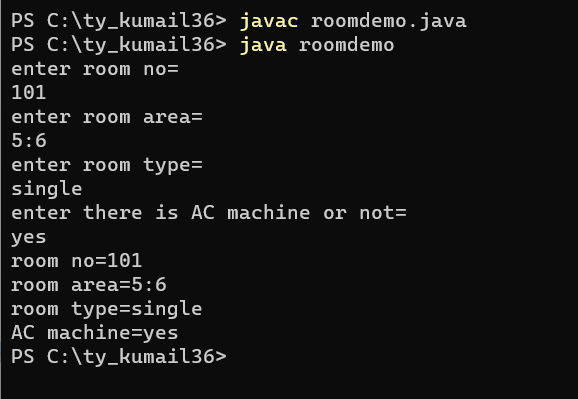
r.setdata();

r.displaydata();

}

}

Output-



class simpleobject

{

simpleobject()

{

System.out.println("welcome to java");

}

public static void main(String args[])

{

simpleobject s=new simpleobject();

}

}

Output-

-

class A

{

int no;

A getA()

{

return this;

}

void display(int no)

{

this.no=no;

System.out.println("no="+no);

}

}

class test

{

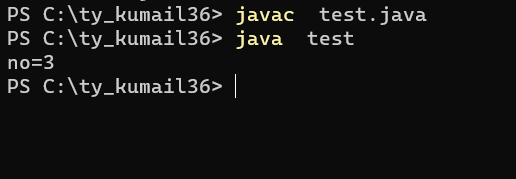
public static void main(String args[])

{

new A().getA().display(3);

}

}

Output-

import java.util.\*; class rectangle

{

int w,h;

rectangle(int w,int h)

{

this.w=w; this.h=h;

}

}

class box extends rectangle

{

int l;

int abox=0,pbox=0,arect=0,prect=0; box(int w,int h,int l)

{

super(w,h); this.l=l;

}

void calculate()

{

arect=w\*h; prect=2\*(w+h);

abox=(2\*l\*w)+(2\*l\*h)+(2\*w\*h); pbox=4\*(l+h+w)\*4\*(l+h+w);

}

void display()

{

System.out.println("area of rectangle="+arect);

System.out.println("perimeter of rectangle="+prect);

System.out.println("area of box="+abox);

System.out.println("perimeter of box="+pbox);

}

}

public class simple

{

public static void main(String args[])

{

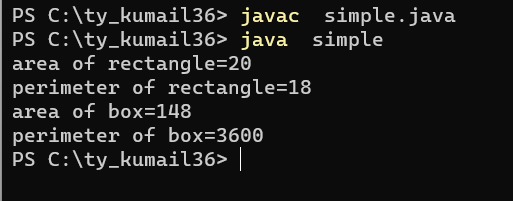
box b=new box(4,5,6);

b.calculate();

b.display();

}

}

Output- 

5.

import java.util.\*; class bankdetail

{

Scanner s=new Scanner (System.in);

long acno;

String acname,actype; double balance; bankdetail()

{

System.out.println("Please Enter followint details"); System.out.println("Enter account holder name="); acname=s.nextLine();

System.out.println("Enter account type="); actype=s.nextLine();

System.out.println("Enter account no="); acno=s.nextLong(); balance=10000;

}

void showdetail()

{

System.out.println("Accout Number="+acno);

System.out.println("Accout holder name="+acname);

System.out.println("Accout type="+actype);

System.out.println("Total balance in the account="+balance);

}

void deposit()

{

long amt;

System.out.println("enter amoutn to deposit");

amt=s.nextLong(); balance=balance+amt;

System.out.println("Total balance after deposit="+balance);

}

void withdrawn()

{

long amt;

System.out.println("enter amount to withdrawn="); amt=s.nextLong(); if(balance>=amt)

{

balance=balance-amt;

System.out.println("balance after withdrawal="+balance);

}

else

{

System.out.println("your total balance is less than your entered amount");

}

}

}

class demo

{

public static void main(String args[])

{

bankdetail b=new bankdetail();

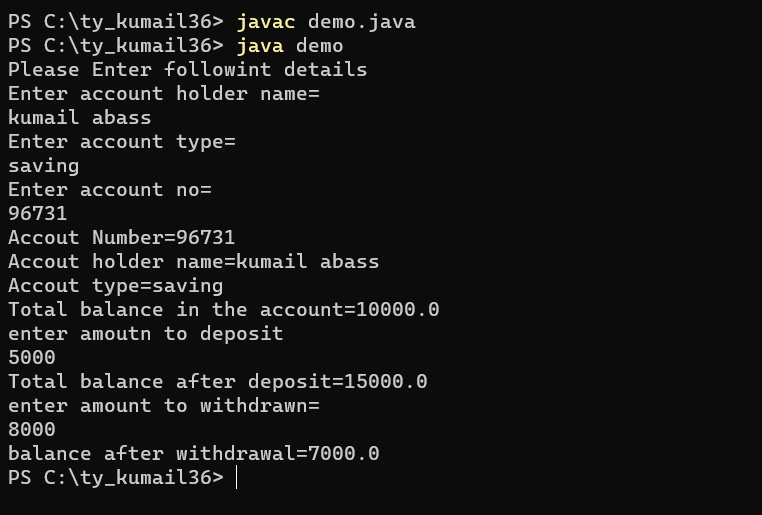
b.showdetail();

b.deposit();

b.withdrawn();

}

}

Output- 

6.

public class employee

{

int ecode;

String ename,address; employee(int ecode,String ename)

{

this.ecode=ecode; this.ename=ename;

}

employee(int ecode,String ename,String address)

{

this.ecode=ecode; this.ename=ename; this.address=address;

}

void display()

{

System.out.println("employee code="+ecode);

System.out.println("employee name="+ename);

System.out.println("employee address="+address);

}

public static void main(String args[])

{

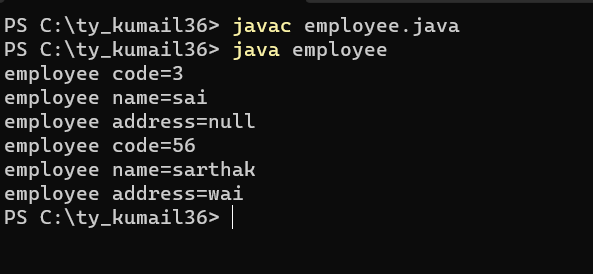
employee e=new employee(3,"sai");

e.display(); employee e2=new employee(56,"sarthak","wai");

e2.display();

}

}



Output-

JOURNAL 4

1.

public class trydemo

{

public static void main(String args[])

{ try

{

int array[]=new int[-5];

System.out.println(array[0]);

}

catch(NegativeArraySizeException ne)

{

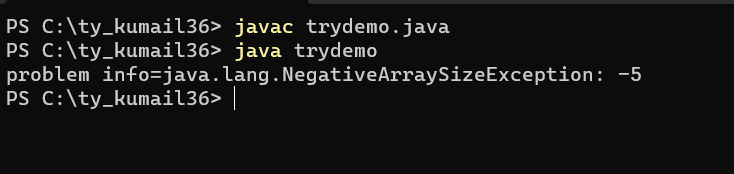
System.out.println("problem info="+ne);

}

}

}

OUTPUT-



2.

class multipletrycatch

{

public static void main(String args[])

{ try

{

int a[]=new int[5];

a[2]=10; a[4]=0;

a[3]=a[2]/a[4];

}

catch(ArithmeticException ae)

{

System.out.println("error:value of divisor cannot be 0");

}

catch(ArrayIndexOutOfBoundsException ar)

{

System.out.println("error:array index more than size");

}

catch(Exception e)

{

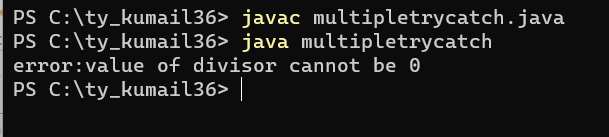
System.out.println("unknown exception has occur");

}

}

}

Output-



3. class trycatchfinally

{

public static void main(String args[])

{ try

{

int a[]=new int[5];

a[2]=10; a[4]=0;

a[3]=a[2]/a[4];

}

catch(Exception e)

{

System.out.println("division by zero is not possible");

}

finally

{

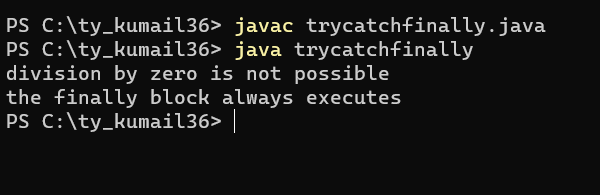
System.out.println("the finally block always executes");

}

}

}

Output-



4.

public class throwsex

{

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

{

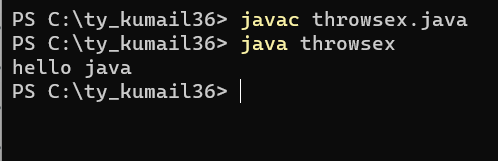
System.out.println("hello java");

Thread.sleep(1000);

}

}

Output-



5.

public class throwtest

{

public static void main(String args[])

{

int num=7;

for(num=1;num<=10;num++)

{ try

{

if(num==5)

throw new ArithmeticException("arithmetic exception");

else if(num<2)

throw new RuntimeException("runtime exception");

else if(num>9)

throw new NullPointerException("null pointer exception");

}

catch(Exception e)

{

System.out.println("caught an exception");

System.out.println(e.getMessage());

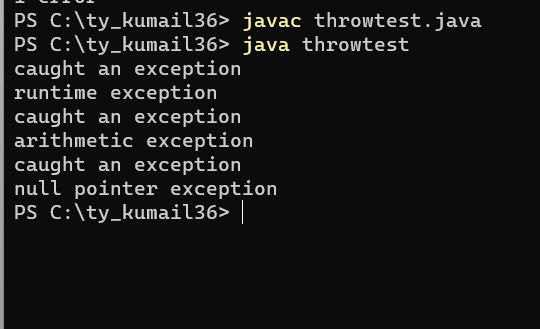
}

}

}

}

Output-



6.

import java.util.\*; class NotEligibleException extends Exception

{

NotEligibleException(String msg)

{

super(msg);

}

}

class voterlist

{

int age;

voterlist(int age)

{

this.age=age;

}

void checkeligibility()

{ try

{

if(age<18)

{

throw new NotEligibleException("error:not eligible for vote due to under age");

}

System.out.println("you are eligible for vote");

}

catch(NotEligibleException ne)

{

System.out.println(ne.getMessage());

}

}

public static void main(String args[])

{

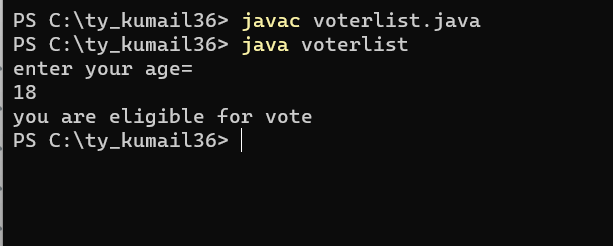
Scanner s=new Scanner(System.in); System.out.println("enter your age="); int age=s.nextInt(); voterlist v=new voterlist(age);

v.checkeligibility();

}

}

Output-



JOURNAL 5

1.

package mypack; public class demopackage

{

int r;

public void getinfo(int r)

{

this.r=r;

}

public void area()

{

System.out.println("area of circle="+(3.14\*r\*r));

}

public void circum()

{

System.out.println("circumference of circle="+(2\*3.14\*r));

}

}

Importing package- import mypack.\*;

public class circle

{

public static void main(String args[])

{

demopackage d=new demopackage();

d.getinfo(4);

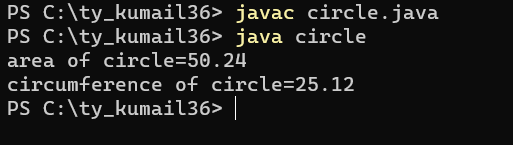
d.area();

d.circum();

}

}

Output-



2.

package p1; public class demop

{

public void display()

{

System.out.println("welcome to package");

}

}

Importing package- import p1.\*; public class ex1

{

public static void main(String args[])

{

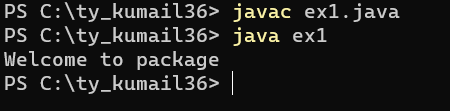
demop d=new demop();

d.display();

}

}

Output-



3.

package extvariable; public class edemo

{

public int no1,no2;

}

Importring package- import extvariable.\*; public class vdemo

{

public static void main(String args[])

{

edemo e=new edemo();

e.no1=10;

e.no2=20;

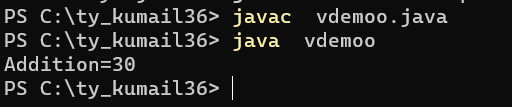
int sum=e.no1+e.no2;

System.out.println("addition="+sum);

}

}

Output-



4.

package userdefined; public class packagedemo

{

public int no1=54,no2=66;

}

Importing package- import userdefined.\*; public class calculation

{

public static void main(String args[])

{

packagedemo p=new packagedemo();

int sum=p.no1+p.no2;

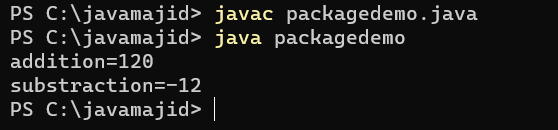
System.out.println("addition="+sum); int sub=p.no1-p.no2;

System.out.println("substraction="+sub);

}

}

Output-



5.

package calculator; public class cal

{

int a,b;

public void getinfo(int a,int b)

{

this.a=a; this.b=b;

}

public void sum()

{

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

}

public void sub()

{

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

}

public void mul()

{

System.out.println("multiplication="+(a\*b));

}

public void div()

{

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

}

}

Importing package- import calculator.\*; public class democalculator

{

public static void main(String args[])

{

cal c=new cal();

c.getinfo(7864,2434);

c.sum();

c.sub();

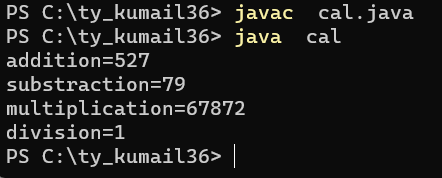
c.mul();

c.div();

}

}

Output-



JOURNAL 6

1.

class mythread implements Runnable

{

public void run()

{ try

{

System.out.println("values from 1 to 5 are="); for(int i=1;i<=5;i++)

{

System.out.println(i);

Thread.sleep(500);

}

}

catch(Exception e)

{

System.out.println("problrm info="+e);

}

}

public static void main(String args[])

{

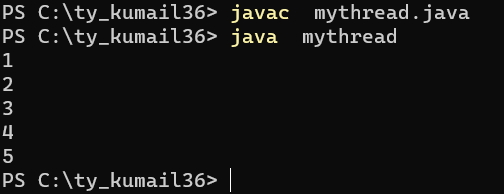
mythread m=new mythread(); Thread t=new Thread(m);

t.start();

}

}

Output-



2.

class mythread extends Thread

{

public mythread(String threadName)

{

super(threadName); start();

}

public void run()

{

for(int i=1;i<=5;i++)

{

System.out.println(getName()+"count"+i);

try

{

Thread.sleep(1000);

}

catch(InterruptedException e)

{

System.out.println(getName()+"interrupted");

}

}

}

}

public class threaddemo

{

public static void main(String args[])

{

new mythread("Thread 1");

try

{

for(int k=5;k>=0;k--)

{

System.out.println("running main thread"+k);

Thread.sleep(1000);

}

}

catch(InterruptedException e)

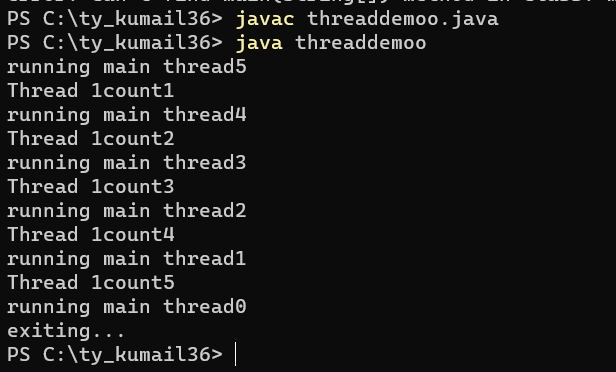
{}

System.out.println("exiting...");

}

}

Output-



3.

public class currentthread

{

public static void main(String args[])

{

Thread currentThread=Thread.currentThread();

System.out.println("current thread name="+currentThread.getName());

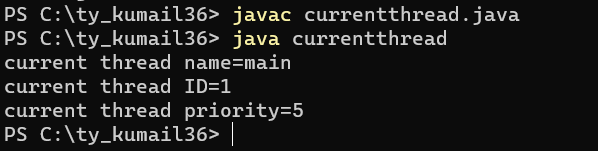
System.out.println("current thread ID="+currentThread.getId());

System.out.println("current thread priority="+currentThread.getPriority());

}

}

Output-



4.

class prioritydemo extends Thread

{

public prioritydemo(String threadName)

{

super(threadName);

}

public void run()

{

System.out.println("thread is running="+getName());

}

}

class threadpriority

{

public static void main(String args[])

{

prioritydemo p=new prioritydemo("Thread 1"); prioritydemo p1=new prioritydemo("Thread 2");

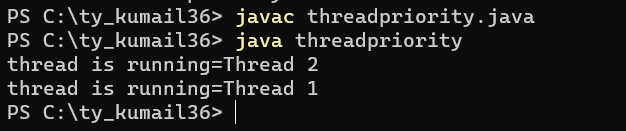
p.setPriority(10); p1.setPriority(1);

p.start(); p1.start();

}

}

Output0-



5. class institute

{

int i;

public void classroom(String facname)

{

synchronized(this)

{

for(i=1;i<=5;i++)

{

System.out.println(i+"class taking by\t"+facname);

}

try

{

Thread.sleep(1000);

}

catch(Exception e)

{

System.out.println(e);

}

}

}

}

class myThread extends Thread

{

institute ins; //ins is an reference variable

String facname;

public void run()

{

ins.classroom(facname);

}

myThread(institute ins,String facname)

{

this.ins=ins; this.facname=facname;

}

}

public class synchronizedblock extends Thread

{

public static void main(String args[])

{

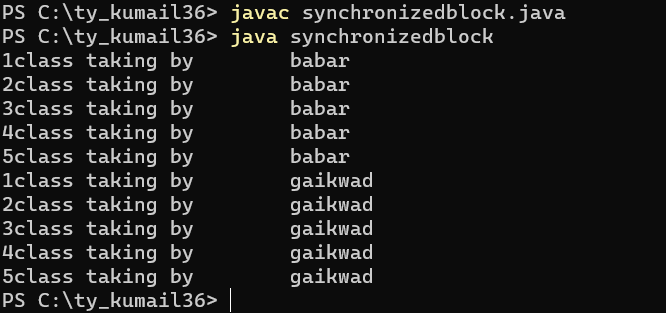
institute in=new institute(); myThread m1=new myThread(in,"babar"); myThread m2=new myThread(in,"gaikwad");

m1.start(); m2.start();

}

}

Output-



6.

class demo

{

int i;

synchronized public void display(String name)

{

for(i=1;i<=5;i++)

{

System.out.println(i+"Name\t"+name);

}

try

{

Thread.sleep(1000);

}

catch(Exception e)

{

System.out.println(e);

}

}

}

class myThread extends Thread

{

demo d; String name; public void run()

{

d.display(name);

}

myThread(demo d,String name)

{

this.d=d; this.name=name;

}

}

public class synchronizedmethod extends Thread

{

public static void main(String args[])

{

demo de=new demo();

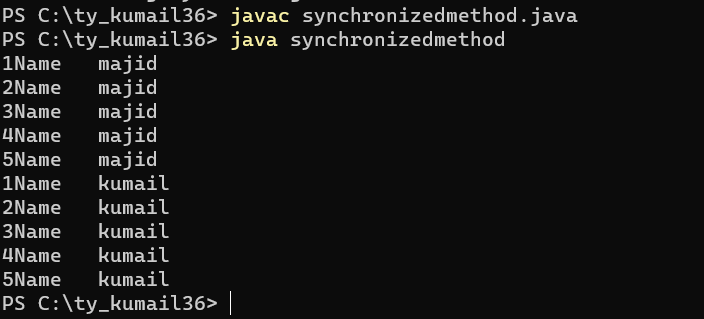
myThread m1=new myThread(de,"vaishnavi"); myThread m2=new myThread(de,"vaibhav");

m1.start(); m2.start();

}

}

Output-



# 

# JOURNAL 7

# 1.

abstract class shape

{

int a=5; int b=7;

abstract void printarea();

}

class circle extends shape

{

double area;

public void printarea()

{

area=3.14\*a\*a;

System.out.println("area of circle="+area);

}

}

class tringle extends shape

{

double area;

public void printarea()

{

area=0.5\*a\*b;

System.out.println("area of trangle="+area);

}

}

class rectangle extends shape

{

int area;

public void printarea()

{

area=a\*b;

System.out.println("area of rectangle="+area);

}

}

abstract class demo

{

public static void main (String argh[])

{

shape s;

s=new circle();

s.printarea(); s=new tringle();

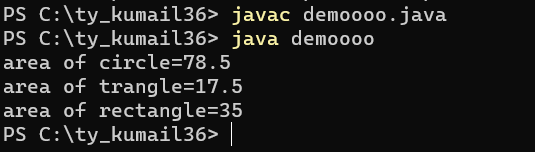
s.printarea(); s=new rectangle();

s.printarea();

}

}

Output-



2.

interface printable

{

void print();

}

interface showabale

{

void show();

}

class A implements printable,showabale

{

public void print()

{

System.out.println("print method");

}

public void show()

{

System.out.println("show method");

}

public static void main(String args[])

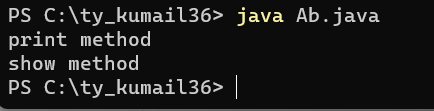
{

A obj=new A(); obj.print(); obj.show();

}

}

OUTPUT-



3.

class base

{

void show()

{

System.out.println("inside parent class method");

}

}

class derived extends base

{

void show()

{

System.out.println("inside child class method");

}

}

class runtimepoly

{

public static void main(String args[])

{

derived d=new derived();

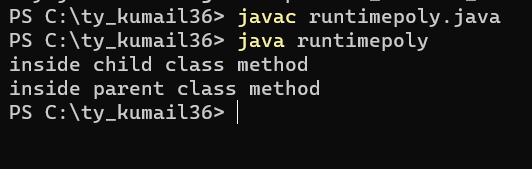
d.show();

base b=new base();

b.show();

}

}



4.

class demooverloading

{

int no1,no2,no3; void sum(int no1,int no2)

{

this.no1=no1; this.no2=no2;

System.out.println("sum="+(no1+no2));

}

void sum(int no1,int no2,int no3)

{

this.no1=no1; this.no2=no2; this.no3=no3;

System.out.println("sum="+(no1+no2+no3));

}

public static void main(String args[])

{

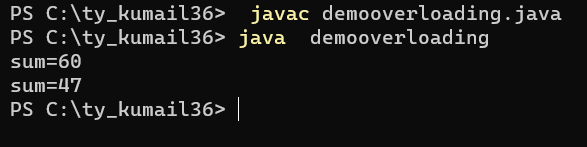
demooverloading d=new demooverloading();

d.sum(4,56);

d.sum(4,8,35);

}

}



5.

class parent

{

void display()

{

System.out.println("parent class method");

}

}

class child extends parent

{

void display()

{

System.out.println("child class method");

}

}

class overridingdemo

{

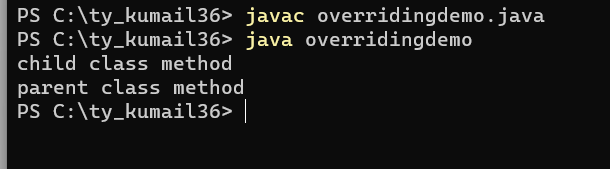
public static void main(String args[])

{

child c=new child();

c.display(); parent p=new parent();

p.display();



6. class animal

{

void eat()

{

System.out.println("animal eats both plants and fish");

}

}

class lion extends animal

{

void eat()

{

System.out.println("lion eats fish because they are carnivors");

}

}

public class dynamicbinding

{

public static void main(String args[])

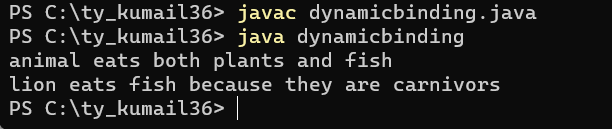
{

animal a=new animal();

a.eat();

animal a1=new lion();

a1.eat()

;

8. interface A

{

void print();

}

interface B

{

void show();

}

class multipleeinterfac implements A,B

{

public void print()

{

System.out.println("parent class1");

}

public void show()

{

System.out.println("parent class2");

}

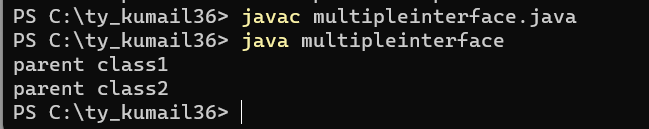
public static void main(String args[])

{

multipleinterface m=new multipleinterface();

m.print();

m.show();



9. class student

{

int rno; String name;

student(int rno,String name)

{

this.rno=rno; this.name=name;

}

}

class Exam extends student

{

int intmark,extmark;

Exam(int rno,String name,int intmark,int extmark)

{

super(rno,name); this.intmark=intmark; this.extmark=extmark;

}

void display()

{

System.out.println("roll number="+rno);

System.out.println("name="+name);

System.out.println("internal marks="+intmark);

System.out.println("external marks="+extmark);

}

}

class result extends Exam

{

int total;

result(int rno,String name,int intmark,int extmark)

{

super(rno,name,intmark,extmark);

}

void show()

{

total=intmark+extmark;

System.out.println("total="+total);

}

}

class demoinheritance

{

public static void main(String args[])

{

result r=new result(1,"sai",59,78);

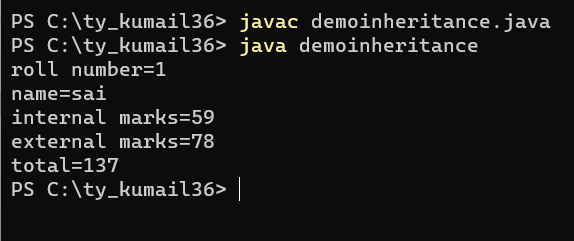
r.display();

r.show();

}

}

Output-



10. class p1

{

void display()

{

System.out.println("method of p1 class");

}

}

class p2 extends p1

{

void show()

{

System.out.println("method of p2 class");

}

}

class inheritanceEx

{

public static void main(String args[])

{

p2 p=new p2();

p.display();

p.show();

}

}

Output-

