1. Define a abstract class figure. Define the area and volume method in the child classes. Use dynamic method dispatch.

Implenent the following design with suitable example classes.

1>

//Abstract class..............

abstract class figure{

abstract void area(int a);

abstract void volume(int b);

}

//Child class

class cube extends figure{

void area(int a){

System.out.println("Area="+a\*a\*6+" unit");

}

void volume(int a){

System.out.println("Volume="+a\*a\*a+" unit");

}

}

//Test class

import java.util.Scanner;

class Test extends cube{

public static void main(String[]args){

Scanner s=new Scanner(System.in);

System.out.println("Enter the side lenth of cube: ");

int i=s.nextInt();

figure f;

f=new cube();

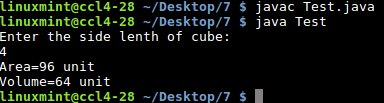
f.area(i);

f.volume(i);

}

}

Output:



2>

//Abstract class..............

abstract class figure implements Area,Volume{

abstract void area1(int a);

}

//interface....................

interface Area {

void area(int a);

}

//interface....................

interface Volume{

void volume(int b);

}

//Child class

class cube extends figure {

public void area(int a){

System.out.println("Area="+a\*a\*6+" unit");

}

public void volume(int a){

System.out.println("Volume="+a\*a\*a+" unit");

}

public void area1(int a){

System.out.println("Area of one surface="+a\*a+" unit");

}

}

//Test class

import java.util.Scanner;

class Test extends cube{

public static void main(String[]args){

Scanner s=new Scanner(System.in);

System.out.println("Enter the side lenth of cube: ");

int i=s.nextInt();

cube f=new cube();

f.area(i);

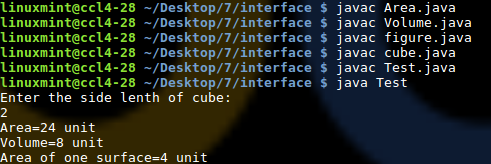
f.volume(i);

f.area1(i);

}

}

Output:



Explain: The methods Area & Volume are in two different interfaces

And area1 in abstract class(figure).this methods are defined in chid class (cube).