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.

**public** **class** Students {

String name;

**int** roll\_no;

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Students s1= **new** Students();

s1.name="John";

s1.roll\_no=2;

}

}

2.Assign and print the roll number, phone number and address of two students having names "Sam" and "John" respectively by creating two objects of class 'Student'.

**public** **class** Students {

String name;

**int** roll\_no;

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Students s1= **new** Students();

Students s2= **new** Students();

s1.name="John";

s1.roll\_no=2;

s2.name="Sam";

s2.roll\_no=3;

System.***out***.println(s1.roll\_no+" "+s1.name);

System.***out***.print(s2.roll\_no+" "+s2.name);

}

}

O/p:

2 John

3 Sam

3. 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.

**public** **class** triangle {

triangle(){

**int** s1=3;

**int** s2=4;

**int** s3=5;

}

**void** display(){

System.***out***.println("perimeter is"+(s1+s2+s3));

System.***out***.println("area is :"+(0.5\*s2\*s3));

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

triangle t1=**new** triangle();

t1.display();

}

}

O/P:

perimeter is12

area is :10.0

4.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' with constructor having the three sides as its parameters.

**public** **class** triangle {

**int** s1;

**int** s2;

**int** s3;

triangle(**int** s, **int** b,**int** h){

s1=s;

s2=b;

s3=h;

}

**void** display(){

System.***out***.println("perimeter is"+(s1+s2+s3));

System.***out***.println("area is :"+(0.5\*s2\*s3));

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

triangle t1=**new** triangle(3,4,5);

t1.display();

}

}

O/P:

perimeter is12

area is :10.0

5. Write a program to print the area of two rectangles having sides (4,5) and (5,8) respectively by creating a class named 'Rectangle' with a method named 'Area' which returns the area and length and breadth passed as parameters to its constructor.

**public** **class** rectangle {

**int** len;

**int** bre;

rectangle(**int** l,**int** b){

len=l;

bre=b;

}

**int** Area() {

**int** a=len\*bre;

**return** a;

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

rectangle r1=**new** rectangle(4,5);

rectangle r2=**new** rectangle(5,8);

System.***out***.println("rectangle r1\nlength "+r1.len+"\nbreath "+r1.bre+"\nArea "+r1.Area());

System.***out***.println("\nrectangle r2\nlength "+r2.len+"\nbreath "+r2.bre+"\nArea "+r2.Area());

}

}

O/P:

rectangle r1

length 4

breath 5

Area 20

rectangle r2

length 5

breath 8

Area 40

6. 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 rectangle are entered through keyboard.

**import** java.util.Scanner;

**public** **class** rectangle {

**int** len;

**int** bre;

**private** **int** a;

rectangle(**int** l,**int** b){

len=l;

bre=b;

}

**void** Area() {

a=len\*bre;

}

**int** returnArea() {

**return** a;

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** b ,h;

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

b=s.nextInt();

h=s.nextInt();

rectangle r1=**new** rectangle(b,h);

b=s.nextInt();

h=s.nextInt();

rectangle r2=**new** rectangle(b,h);

r1.Area();

r2.Area();

System.***out***.println("rectangle r1\nlength "+r1.len+"\nbreath "+r1.bre+"\nArea "+r1.returnArea());

System.***out***.println("\nrectangle r2\nlength "+r2.len+"\nbreath "+r2.bre+"\nArea "+r2.returnArea());

}

}

O/P:

5

4

7

8

rectangle r1

length 5

breath 4

Area 20

rectangle r2

length 7

breath 8

Area 56

7. 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.Scanner;

**public** **class** complexNos {

**int** a,b,c,d,r,i;

complexNos(**int** a,**int** b, **int** c, **int** d){

**this**.a=a;

**this**.b=b;

**this**.c=c;

**this**.d=d;

}

**void** add() {

r=a+c;

i=b+d;

System.***out***.println("sum is :"+r+"+"+i+"i");

}

**void** sub() {

r=a-c;

i=b-d;

System.***out***.println("difference is :"+r+i+"i");

}

**void** mul() {

r=(a\*c)-(b\*d);

i=(a\*d)-(b\*c);

System.***out***.println("product is :"+r+i+"i");

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** a,b,c,d;

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

System.***out***.println("enter real part num 1: ");

a=s.nextInt();

System.***out***.println("enter imag part num 1: ");

b=s.nextInt();

System.***out***.println("enter real part num 2: ");

c=s.nextInt();

System.***out***.println("enter imag part num 2: ");

d=s.nextInt();

complexNos com=**new** complexNos(a,b,c,d);

com.add();

com.sub();

com.mul();

}

}

O/P:

enter real part num 1:

2

enter imag part num 1:

3

enter real part num 2:

4

enter imag part num 2:

5

sum is :6+8i

difference is :-2-2i

product is :-7-2i

8.Write a program that would print the information (name, year of joining, salary, address) of three employees by creating a class named 'Employee'. The output should be as follows: Name Year of joining Address

**public** **class** employee {

String name;

**int** YOJ;

String Address;

**void** setData(String n ,**int** yoj,String ad){

name=n;

YOJ=yoj;

Address=ad;

}

**void** display(){

System.***out***.println(name+"\t\t"+YOJ+"\t\t"+Address);

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

//array of object

employee arr[]= **new** employee[3];

arr[0]=**new** employee();

arr[1]=**new** employee();

arr[2]=**new** employee();

arr[1].setData("Robert",1994,"64C-WallsStreet");

arr[2].setData("Sam",2000,"68C-wallzSt");

arr[0].setData("Jhon",1999,"56B-mountSt");

System.***out***.println("NAME\t YEAR OF JOINING\tADDRESS");

**for**(**int** i=0;i<3;i++) {

arr[i].display();

}

}

}

O/P:

NAME YEAR OF JOINING ADDRESS

Jhon 1999 56B-mountSt

Robert 1994 64C-WallsStreet

Sam 2000 68C-wallzSt

9. The Matrix class has methods for each of the following: 1 - get the number of rows 2 - get the number of columns 3 - set the elements of the matrix at given position (i,j) 4 - adding two matrices. If the matrices are not addable, "Matrices cannot be added" will be displayed. 5 - multiplying the two matrices

**import** java.util.Scanner;

**public** **class** matrix {

**int** a[][]=**new** **int**[3][3];

**int** r[][]=**new** **int**[3][3];

matrix(**int** a[][]){

**this**.a=a;

}

**void** getRow() {

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

}

**void** getCol() {

System.***out***.println(a[0].length);

}

**void** setEle(**int** i,**int** j,**int** e) {

a[i][j]=e;

display(a,"setEle");

}

**void** add(**int** a[][],**int** b[][] ) {

**if**(a.length!=b.length || a[0].length!=b[0].length)

{

System.***out***.println("addition is not posible");

}

**else** {

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

**for**(**int** j=0;j<b[0].length;j++) {

r[i][j]=a[i][j]+b[i][j];

}

}

display (r,"add");

}

}

**void** mul(**int** a[][],**int** b[][]) {

**if**(a[0].length!=b.length)

{

System.***out***.println("multiplication is not posible");

}

**else** {

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

**for**(**int** j=0;j<a[0].length;j++) {

r[i][j]=0;

**for**(**int** k=0;k<3;k++) {

r[i][j]+=a[i][j]\*b[i][j];

}

}

}

display(r,"mul");

}

}

**void** display(**int** a[][],String str){

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

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

**for**(**int** j=0;j<a[0].length;j++) {

System.***out***.print(" "+a[i][j]);

}

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

}

System.***out***.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

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

**int** a[][]= {{1,2,3},{4,5,6},{7,8,9}};

**int** b[][]= {{2,4,6},{8,10,12},{14,16,18}};

**int** c[][]= {{3,6},{9,12},{6,7}};

matrix m1=**new** matrix(a);

matrix m2=**new** matrix(b);

matrix m3=**new** matrix(c);

m1.display(a,"A matrix");

m2.display(b,"B matrix");

m3.display(c,"C matrix");

m1.getRow();

m3.getCol();

m1.setEle(0, 0, 9);

m2.setEle(2, 2, 1);

m1.add(a, b);

m3.add(a,c);

m2.mul(a, b);

m3.mul(c, b);

}

}

O/P:

A matrix

1 2 3

4 5 6

7 8 9

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

B matrix

2 4 6

8 10 12

14 16 18

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

C matrix

3 6

9 12

6 7

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3

2

setEle

9 2 3

4 5 6

7 8 9

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

setEle

2 4 6

8 10 12

14 16 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

add

11 6 9

12 15 18

21 24 10

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

addition is not posible

mul

54 24 54

96 150 216

294 384 27

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

multiplication is not posible

10. Write a program to print the names of students by creating a Student class. If no name is passed while creating an object of Student class, then the name should be "Unknown", otherwise the name should be equal to the String value passed while creating object of Student class.

**public** **class** student {

String stud=”UNKNOWN”;

student(){

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

}

student(String str){

stud=str;

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

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

student s1=**new** student();

student s2= **new** student("Sri");

}

}

O/P:

null

Sri

11. Will the following code snippet compile successfully? If yes, what is the output of the following program?

Ans: No,it not compile successfully.

12.Identify the error in the following code snippet. If there is no error then what will be the output of the program?

Ans :10

13.what is the output

Ans:20

30

400

14.whats the output

Ans:200

15.Whats the error in the code

Ans:No error

24

16.Will this program execute what will be the output

Ans:

Static method

Instance method

Constructor

17.whats the output

Ans:

This is rectangle

Area:20

18. Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call a - method of parent class by object of parent class b - method of child class by object of child class c - method of parent class by object of child class

**class** parent{

**void** display() {

System.***out***.println("I am Parent");

}

}

**public** **class** child **extends** parent{

**void** show(){

System.***out***.println("I am child");

}

**public** **static** **void** main(String args[]) {

parent p=**new** parent();

child c=**new** child();

p.display();

c.show();

c.display();

}

}

O/P:

I am Parent

I am child

I am Parent

19. Create a class named 'Member' having the following members: Data members 1 - Name 2 - Age 3 - Phone number 4 - Address 5 - Salary It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.

//

**class** members{

String name;

**int** age;

**double** PH;

String Address;

**int** Salary;

**void** printSalary() {

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

}

}

**class** Manager **extends** members{

String Specizattion;

String Department;

}

**public** **class** employe **extends** members{

String Specizattion;

String Department;

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

employe e= **new** employe();

e.name="Bose";

e.age=22;

e.Address="Ashok nagar";

e.PH=1234567895;

e.Salary=24000;

Manager a=**new** Manager();

a.name="subash";

a.age=32;

a.Address="kk nagar";

a.PH=1234567895;

a.Salary=360000;

e.printSalary();

a.printSalary();

}

}

O/P:

24000

360000

20.

Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'. Print the area and perimeter of a rectangle and a square.

**class** rectangle1{

**int** a,p;

rectangle1(**int** l,**int** b){

a=l\*b;

p=(2\*l)+(2\*b);

}

**void** area() {

System.***out***.println("Area is"+a);

}

**void** perimeter() {

System.***out***.println("perimeter is"+p);

}

}

**public** **class** Square **extends** rectangle1{

Square(**int** l, **int** b) {

**super**(l, b);

// **TODO** Auto-generated constructor stub

}

**public** **static** **void** main(String args[]) {

Square s=**new** Square(3,3);

rectangle1 r=**new** rectangle1(3,9);

r.area();

r.perimeter();

s.area();

s.perimeter();

}

}

O/P:

Area is27

perimeter is24

Area is9

perimeter is12

21 whats the output

Ans: Inside GeeksforGeeks

22.can we overload main() method?

Ans: No,we cann’t overload main() method because it is static.

23.what is the output

Ans:200(same as 14)

**24.**

Ans:Sum of two numbers: 50

Sum of three numbers: 120

**25.what is the output**

Sum of two numbers: 50

**26.what is the output**

m1 method in class A

m1 method in class A

m1 method in class B

m1 method in class B

m1 method in class B

**27.What is the output**

I don’t know about sql.

**28.what is the output**

Ans: no output nor exception,

as there is nothing in bar().