Java Questions

.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.

class Student {  
 String name;  
 int roll\_no;  
 void getData(){  
 System.*out*.println(name+" "+roll\_no);  
 }  
 public static void main(String[] args) {  
 Student obj = new Student();  
 obj.name="John";  
 obj.roll\_no=2;  
 obj.getData();  
 }  
}

.   
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'.

class Student {  
 String name;  
 int roll\_no;  
 int pno;  
 String add;  
 void setData(String name, int roll\_no, int pno, String add){  
 this.name=name;  
 this.roll\_no=roll\_no;  
 this.pno=pno;  
 this.add=add;  
 }  
 void getData(){  
 System.*out*.println("Name- "+name+" R.No- "+roll\_no+" P.No- "+pno+" Add- "+add);  
 }  
 public static void main(String[] args) {  
 Student obj = new Student();  
 obj.setData("John",2,224400,"USA");  
 obj.getData();  
 Student obj1 = new Student();  
 obj1.setData("Sam",1,54464864,"Malaysia");  
 obj1.getData();  
 }  
}

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.

class Triangle {  
 int a= 3;  
 int b=4;  
 int c=5;  
 Triangle(){  
 int p= a+b+c;  
 double s = p/2;  
 double area= Math.*sqrt*(s\*(s-a)\*(s-b)\*(s-c));  
 System.*out*.println("Area-"+area);  
 System.*out*.println("Perimeter-"+p);  
 }  
  
 public static void main(String[] args) {  
 Triangle obj = new Triangle();  
 }  
}

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.

class Triangle {  
 Triangle(int a,int b,int c){  
 int p= a+b+c;  
 int s = p/2;  
 double area= Math.*sqrt*(s\*(s-a)\*(s-b)\*(s-c));  
 System.*out*.println("Area-"+area);  
 System.*out*.println("Perimeter-"+p);  
 }  
  
 public static void main(String[] args) {  
 Triangle obj = new Triangle(4,5,7);  
 }  
}

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.

class Rectangle {  
 int a;  
 int b;  
 Rectangle(int a,int b){  
 this.a=a;  
 this.b=b;  
 }  
 void area(){  
 int area= a\*b;  
 System.*out*.println("Area-"+area);  
 }  
  
 public static void main(String[] args) {  
 Rectangle obj = new Rectangle(4,5);  
 obj.area();  
 Rectangle obj1 = new Rectangle(5,8);  
 obj1.area();  
 }  
}

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;  
  
class Area {  
 int a;  
 int b;  
 Area(int a,int b){  
 this.a=a;  
 this.b=b;  
 }  
 void returnArea(){  
 int area= a\*b;  
 System.*out*.println("Area-"+area);  
 }  
  
 public static void main(String[] args) {  
 Scanner s =new Scanner(System.*in*);  
 System.*out*.println("Enter the Length and Breadth-");  
 int x = s.nextInt();  
 int y = s.nextInt();  
 Area obj = new Area(x,y);  
 obj.returnArea();  
 }  
}

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;  
  
class Complex{  
 int r1;  
 int i1;  
 int r2;  
 int i2;  
 Complex(int r1,int i1,int r2,int i2){  
 this.r1 = r1;  
 this.i1 = i1;  
 this.r2 = r2;  
 this.i2 = i2;  
 }  
 void sum(){  
 int s1= r1+r2;  
 int s2= i1+i2;  
 System.*out*.println("Sum is "+s1+" +i"+s2);  
 }  
 void difference(){  
 int d1= r1-r2;  
 int d2= i1-i2;  
 System.*out*.println("Difference is "+d1+" +i"+d2);  
 }  
 void product(){  
 int p1= r1\*r2;  
 int p2= i1\*i2;  
 System.*out*.println("Product is "+p1+" +i"+p2);  
 }  
  
 public static void main(String[] args) {  
 Scanner s1 = new Scanner(System.*in*);  
 System.*out*.println("Enter the real and imaginary parts of 1st no-");  
 int x1= s1.nextInt();  
 int y1= s1.nextInt();  
 Scanner s2 = new Scanner(System.*in*);  
 System.*out*.println("Enter the real and imaginary parts of 2nd no-");  
 int x2= s2.nextInt();  
 int y2= s2.nextInt();  
 Complex obj = new Complex(x1,y1,x2,y2);  
 obj.sum();  
 obj.difference();  
 obj.product();  
 }  
}

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  
Robert            1994                64C- WallsStreat  
Sam                2000                68D- WallsStreat  
John                1999                26B- WallsStreat

class Employee{  
 String name;  
 String add;  
 int yoj;  
 Employee(String name, int yoj, String add){  
 this.name= name;  
 this.add= add;  
 this.yoj=yoj;  
 System.*out*.println(name+"\t"+yoj+"\t"+add);  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println("Name\t Year of Joining\t Address");  
 Employee e1= new Employee("Robert",1994,"64C-WallStreet");  
 Employee e2= new Employee("Sam",2000,"68D-WallStreet");  
 Employee e3= new Employee("John",1999,"26B-WallStreet");  
 }  
}

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;  
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) {  
 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);  
  
 }  
  
}

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.

class Student{  
 String name;  
 Student(){  
 name="Unknown";  
 System.*out*.println(name);  
 }  
 Student(String name){  
 this.name=name;  
 System.*out*.println(name);  
 }  
 public static void main(String[] args) {  
 Student s1= new Student();  
 Student s2= new Student("Mohit");  
 }  
}

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

public class Myclass

{

private int x = 10;

static int m1() {

int y = x;

return y;

}

public static void main(String[] args) {

m1();

}

}

**NO**

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

public class Myclass

{

private int x = 10;

static int m1()

{

Myclass obj = new Myclass();

int y = obj.x;

return y;

}

public static void main(String[] args) {

System.out.println(m1());

}

}

**OUTPUT-10**

13.what is the output

public class Myclass

{

static int a = 20;

static int b = 30;

static int c = 40;

Myclass()

{

a = 200;

}

static void m1() {

b = 300;

}

static {

c = 400;

}

public static void main(String[] args) {

System.out.println(a);

System.out.println(b);

System.out.println(c);

}

}

**OUTPUT-**

**20**

**30**

**400**

14.whats the output

public class Myclass {

static int a = 20;

Myclass() {

a = 200;

}

public static void main(String[] args) {

new Myclass();

System.out.println(a);

}

}

**OUTPUT- 200**

15.Whats the error in the code

public class Myclass {

static int a = 20;

Myclass() {

a++;

}

void m1() {

a++;

System.out.println(a);

}

public static void main(String[] args)

{

Myclass obj = new Myclass();

Myclass obj2 = new Myclass();

Myclass obj3 = new Myclass();

obj3.m1();

}

}

**OUTPUT-24**

16.Will this program execute what will be the output

public class Test

{

Test(Test t) {

m1();

System.out.println("Constructor");

}

void m1() {

m2();

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

}

static void m2() {

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

}

public static void main(String[] args)

{

new Test(null);

}

}

**OUTPUT-**

**Static method**

**Instance method**

**Constructor**

17.whats the output

import java.util.Scanner;

class Figure

{

final int length = 5;

final int bredth = 4;

final void area()

{

int a = length \* bredth;

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

}

}

class Rectangle extends Figure

{

final void rect()

{

System.out.println("This is rectangle");

}

}

final public class Final\_Use extends Rectangle

{

public static void main(String[] args)

{

Final\_Use obj = new Final\_Use();

obj.rect();

obj.area();

}

}

**OUTPUT-**

**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 Test{  
 void work(){  
 System.*out*.println("This is Parent Class");  
 }  
}  
class Exam extends Test{  
 void todo(){  
 System.*out*.println("This is Child Class");  
 }  
  
 public static void main(String[] args) {  
 Test obj = new Test();  
 obj.work();  
 Exam obj1 = new Exam();  
 obj1.todo();  
 obj1.work();  
 }  
}

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 Member{  
 String name;  
 int age;  
 int phn;  
 String add;  
 int sal;  
  
 Member(String name, int age, int phn, String add, int sal){  
 this.name=name;  
 this.age = age;  
 this.phn = phn;  
 this.add = add;  
 this.sal = sal;  
 }  
  
 void printSalary(int sal){  
 System.*out*.println("Salary is "+sal);  
 }  
}  
class Employee extends Member {  
 String specialization;  
 String department;  
  
 Employee(String name, int age, int phn, String add, int sal, String specialization, String department) {  
 super(name, age, phn, add, sal);  
 this.specialization = specialization;  
 this.department = department;  
 }  
  
 void display() {  
 System.*out*.println("Name- " + name + " Age- " + age + " Phone- " + phn + " Add- " + add + " Salary- " + sal + " Specialization- " + specialization + " Department- " + department);  
 }  
}  
class Manager extends Employee{  
 Manager(String name, int age, int phn, String add, int sal, String specialization, String department) {  
 super(name, age, phn, add, sal, specialization, department);  
 }  
  
 public static void main(String[] args) {  
 Employee e1 = new Employee("Mohit",21,224582,"India",600000,"Java","Engineer");  
 e1.display();  
 Manager m1 = new Manager("Mohit",21,224582,"India",600000,"Java","Engineer");  
 m1.display();  
 }  
}

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 Rectangle{  
 int length;  
 int breadth;  
  
 Rectangle(int length,int breadth){  
 this.length = length;  
 this.breadth = breadth;  
 }  
  
 void area(){  
 int a = length\*breadth;  
 System.*out*.println("Area-"+a);  
 }  
  
 void perimeter(){  
 int p = 2\*(length+breadth);  
 System.*out*.println("Perimeter-"+p);  
 }  
}  
class Square extends Rectangle {  
 Square(int s) {  
 super(s, s);  
 }  
  
 public static void main(String[] args) {  
 Rectangle r1 = new Rectangle(6, 4);  
 r1.area();  
 r1.perimeter();  
  
 Square s1 = new Square(5);  
 s1.area();  
 s1.perimeter();  
 }  
  
}

21 whats the output

class One

{

protected void getData()

{

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

}

}

class Two extends One

{

protected void getData()

{

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

}

}

public class Test

{

public static void main(String[] args)

{

One obj = new Two();

obj.getData();

}

}

**OUTPUT- Inside GeeksforGeeks**

22.can we overload main() method?

**Yes, we can overload the main() method. But remember that the JVM always calls the original main() method. It does not call the overloaded main() method.**

23.what is the output

public class Myclass {

static int a = 20;

Myclass() {

a = 200;

}

public static void main(String[] args) {

new Myclass();

System.out.println(a);

}

}

**OUTPUT- 200**

24.class A {

void sum(int x, int y){

System.out.println("Sum of two numbers: " +(x+y));

}

void sum(int x, int y, int z){

System.out.println("Sum of three numbers: " +(x+y+z));

}

public static void main(String[] args){

A a = new A();

a.sum(20, 30);

a.sum(30, 40, 50);

}

}

**OUTPUT-**

**Sum of two numbers: 50**

**Sum of three numbers: 120**

25.what is the output

class A {

void sum(int x, int y){

System.out.println("Sum of two numbers: " +(x+y));

}

void sum(int y, int x){

System.out.println("Sum of three numbers: " +(x+y));

}

public static void main(String[] args){

A a = new A();

a.sum(20, 30);

}

}

**OUTPUT-**

**ERROR- method sum(int,int) is already defined in class A**

26.what is the output

class A {

void m1(A a){

System.out.println("m1 method in class A");

}

}

class B extends A {

public void m1(A a){

System.out.println("m1 method in class B");

}

}

public class Test{

public static void main(String[] args){

A a = new A();

a.m1(a);

a.m1(new B());

B b = new B();

b.m1(null);

a = b;

a.m1(null);

a.m1(new A());

}

}

**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

import java.io.FileNotFoundException;

import java.io.IOException;

import javax.sql.SQLException;

public class ExceptionInterviewQuestion\_01 {

public static void main(String[] args) {

try {

test();

} catch (IOException e) {

e.printStackTrace();

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

}

}

public static void test() throws IOException, SQLException, FileNotFoundException{

System.out.println("Inside test() method");

}

}

**OUTPUT- Inside test() method**

28.what is the output

public class TestException3 {

public static void main(String[] args) {

try{

bar();

}catch(NullPointerException e){

e.printStackTrace();

}catch(Exception e){

e.printStackTrace();

}

foo();

}

public static void bar(){

}

public static void foo() throws NullPointerException{

}

}

**OUTPUT- No output as nothing to print.**