

- 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.

Code: -

```
Student.java x
2 usages
1 public class Student {
2
3     //declaring variables
4     String name;
5     int roll_no;
6
7     //creating main method to create obj of class Student
8     public static void main(String[] args){
9         Student obj = new Student(); //creating object of class Student
10
11         //assigning values to class variables using object of class
12         obj.name = "John";
13         obj.roll_no = 2;
14
15         //printing values
16         System.out.println("Name is : " + obj.name);
17         System.out.println("Roll No is : " + obj.roll_no);
18     }
19 }
20
```

Output:-

```
Student x
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Name is : John
Roll No is : 2

Process finished with exit code 0
```

- 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'.

Code:-

```
Student.java x
4 usages
1 public class Student {
2
3     //declaring variables
4     int rollNo;
5     String phoneNo;
6     String address;
7
8     //creating method to get details
9     void getDetails(int rollNo, String phoneNo, String address){
10         this.address=address;
11         this.phoneNo=phoneNo;
12         this.rollNo=rollNo;
13     }
14 }
```

```

14
15 //creating method to display details
16 2 usages
17 void displayDetails(){
18     System.out.println("Roll No. is : " + rollNo);
19     System.out.println("Phone No. is : " + phoneNo);
20     System.out.println("Address is : " + address);
21 }
22
23 //creating main method to create object of class
24 public static void main(String[] args){
25
26     Student obj1 = new Student(); //creating object 1 of class "Student"
27
28     obj1.getDetails( rollNo: 101, phoneNo: "6756789876", address: "Delhi");
29     obj1.displayDetails();
30
31     System.out.println();
32
33     Student obj2 = new Student(); //creating object 2 of class "Student"
34
35     obj2.getDetails( rollNo: 102, phoneNo: "9987698427", address: "Kolkata");
36     obj2.displayDetails();
37
38 }
39 }
40

```

Output:-

```

Student x
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Roll No. is : 101
Phone No. is : 6756789876
Address is : Delhi

Roll No. is : 102
Phone No. is : 9987698427
Address is : Kolkata

Process finished with exit code 0

```

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.

Code:-

```

Triangle.java x
2 usages
1 public class Triangle {
2
3     //declaring variables
4     3 usages
5     float sideA;
6     3 usages
7     float sideB;
8     2 usages
9     float sideC;
10
11     2 usages
12     double area;
13
14     2 usages
15     double perimeter;
16
17     //creating default constructor to initialize the values of side of Triangle
18     1 usage
19     Triangle(){
20         sideA=3;
21         sideB=4;
22         sideC=5;
23     }
24
25

```

```

19 //creating method to calculate and display area of triangle of given sides
1 usage
20 void areaOfTriangle(){
21     area = (sideA*sideB)/2;
22
23     System.out.println("Area of Triangle having sides 3, 4 and 5 is : " + area);
24 }
25
26 //creating method to calculate and display perimeter of triangle of given sides
1 usage
27 void perimeterOfTriangle(){
28     perimeter = sideA + sideB + sideC;
29     System.out.println("Perimeter of Triangle having sides 3, 4 and 5 is : " + perimeter);
30 }
31
32 //creating main method to create object of class
33 public static void main(String[] args){
34     Triangle obj = new Triangle(); //creating object of class "Triangle"
35
36     obj.areaOfTriangle();
37     System.out.println();
38     obj.perimeterOfTriangle();
39 }
40 }
41

```

Output:-

```

Triangle x
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Area of Triangle having sides 3, 4 and 5 is : 6.0

Perimeter of Triangle having sides 3, 4 and 5 is : 12.0

Process finished with exit code 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.

Code:-

```

Triangle.java x
2 usages
1 public class Triangle {
2
3     //declaring variables
3 usages
4     float sideA;
3 usages
5     float sideB;
2 usages
6     float sideC;
7
8     2 usages
double area;
2 usages
9     double perimeter;
10
11     //declaring and initializing parameterized constructor with parameter sideA, sideB, sid
1 usage
12     Triangle(float sideA, float sideB, float sideC){
13         this.sideA = sideA;
14         this.sideB = sideB;
15         this.sideC = sideC;
16     }
17
18     //creating method to calculate Area of Triangle
1 usage
19     void areaOfTriangle(){
20         area = (sideA*sideB)/2;
21
22         System.out.println("Area of Triangle having sides 3, 4 and 5 is : " + area);
23     }
24

```

```

25 //creating method to find Perimeter of Triangle
26 1 usage
27 void perimeterOfTriangle(){
28     perimeter = sideA + sideB + sideC;
29     System.out.println("Perimeter of Triangle having sides 3, 4 and 5 is : " + perimeter);
30 }
31
32 //creating main method to create object of class Triangle
33 public static void main(String[] args){
34     Triangle obj = new Triangle( sideA: 3, sideB: 4, sideC: 5); //passing arguments (3,4,5) to parameterized constructor
35
36     obj.areaOfTriangle();
37     System.out.println();
38     obj.perimeterOfTriangle();
39 }
40

```

Output:-

```

Triangle x
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Area of Triangle having sides 3, 4 and 5 is : 6.0

Perimeter of Triangle having sides 3, 4 and 5 is : 12.0

Process finished with exit code 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.

Code:-

```

Rectangle.java x
4 usages
1 public class Rectangle {
2     //declaring variables
3     4 usages
4     float length;
5     4 usages
6     float breadth;
7     2 usages
8     double area;
9
10    //declaring parameterized constructor and initializing variables
11    2 usages
12    Rectangle(float length,float breadth){
13        this.length=length;
14        this.breadth=breadth;
15    }
16
17    //creating method to calculate and return area of rectangle
18    2 usages
19    double Area(){
20        area = length*breadth;
21
22        return area;
23    }
24
25    //creating main method to create object of class
26    public static void main(String[] args){
27
28        Rectangle obj1 = new Rectangle( length: 4, breadth: 5); //creating object 1 of class "Rectangle"
29        System.out.println("Area of Rectangle of side " + (int)obj1.length + " and " + (int)obj1.breadth + " is : " + obj1.Area() + " units Sqr");
30
31        Rectangle obj2 = new Rectangle( length: 5, breadth: 8); //creating object 2 of class "Rectangle"
32        System.out.println("Area of Rectangle of side " + (int)obj2.length + " and " + (int)obj2.breadth + " is : " + obj2.Area() + " units Sqr");
33    }
34 }

```

Output:-

```
Rectangle
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Area of Rectangle of side 4 and 5 is : 20.0 units Sqr
Area of Rectangle of side 5 and 8 is : 40.0 units Sqr

Process finished with exit code 0
```

- 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.

Code:-

```
Area.java
2 usages
1 public class Area {
2     //declaring variables
3     float length;
4     float breadth;
5
6     //declaring parameterized constructor and initializing value of sides of rectangle
7     Area(float length, float breadth){
8         this.length=length;
9         this.breadth=breadth;
10    }
11
12    //creating method to calculate the area of rectangle and return the value of area
13    double returnArea(){
14        return length*breadth;
15    }
16
17    //creating main method to create the object of class
18    public static void main(String[] args){
19        Area obj1 = new Area( length: 20, breadth: 15); //creating object of class "Area" with arguments
20
21        System.out.println("Area of Rectangle of side " + obj1.length + " and " + obj1.breadth + " is " + obj1.returnArea());
22    }
23 }
24
```

Output:-

```
Area
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Area of Rectangle of side 20.0 and 15.0 is 300.0

Process finished with exit code 0
```

- 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.

Code:-

```
Complex.java
2 usages
1 public class Complex {
2
3     //declaring variables
4     int real1, real2;
5     int complex1, complex2;
6     int real,img;
7
```

```

8      //creating method to add two complex numbers
9      1 usage
10     void sumComplex(int real1, int complex1, int real2, int complex2){
11         real = real1+real2;
12         img = complex1+complex2;
13
14         System.out.println("Sum of " + real1 + " + " + complex1 + "i and " + real2 + " + " + complex2 + "i is : " + real + " + "+img + "i");
15     }
16
17     //creating method to find difference of two complex numbers
18     1 usage
19     void differenceComplex(int real1, int complex1, int real2, int complex2){
20         real = real1-real2;
21         img = complex1-complex2;
22
23         System.out.println("Difference of " + real1 + " + " + complex1 + "i and " + real2 + " + " + complex2 + "i is : " + real + " + "+img + "i" );
24     }
25
26     //creating method to find product of two complex numbers
27     1 usage
28     void productComplex(int real1, int complex1, int real2, int complex2){
29         real = (real1*real2) - (complex1*complex2);
30         img = (real1*complex2)+(real2*complex1);
31
32         System.out.println("Product of " + real1 + " + " + complex1 + "i and " + real2 + " + " + complex2 + "i is : " + real + " + "+img + "i");
33     }
34
35     //creating main method to create object of class
36     public static void main(String[] args){
37
38         Complex obj = new Complex();
39
40         obj.sumComplex( real1: 3, complex1: 2, real2: 1, complex2: 4);
41         obj.differenceComplex( real1: 3, complex1: 2, real2: 1, complex2: 4);
42         obj.productComplex( real1: 3, complex1: 2, real2: 1, complex2: 4);
43     }
44 }

```

Output:-

```

Complex x
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Sum of 3 + 2i and 1 + 4i is : 4 + 6i
Difference of 3 + 2i and 1 + 4i is : 2 + -2i
Product of 3 + 2i and 1 + 4i is : -5 + 14i

Process finished with exit code 0

```

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

Code:-

```

Employee.java x
6 usages
1  public class Employee {
2
3      //declaring variables
4      2 usages
5      String employeeName;
6      2 usages
7      String yearOfJoining;
8      2 usages
9      String employeeSalary;
10     2 usages
11     String employeeAddress;
12
13     //creating method to get employee details
14     3 usages
15     void getEmployeeDetails(String employeeName,String yearOfJoining,String employeeSalary,String employeeAddress){
16         this.employeeName = employeeName;
17         this.yearOfJoining = yearOfJoining;
18         this.employeeSalary = employeeSalary;
19         this.employeeAddress = employeeAddress;
20     }
21 }

```

```

18 //creating method to display employee details
19 3 usages
20 void displayEmployeeDetails(){
21     System.out.println(employeeName+"\t\t"+yearOfJoining+"\t\t"+employeeSalary+"\t\t"+employeeAddress);
22 }
23 //creating main method to create object of class
24 public static void main(String[] args){
25
26     Employee emp1 = new Employee(); //creating object for employee 1
27     Employee emp2 = new Employee(); //creating object for employee 2
28     Employee emp3 = new Employee(); //creating object for employee 3
29
30     System.out.println("Name\t\tYear of joining\t\tSalary\t\t\tAddress\n");
31
32     emp1.getEmployeeDetails( employeeName: "Robert", yearOfJoining: "1994", employeeSalary: "60000", employeeAddress: "64C- WallsStreat");
33     emp1.displayEmployeeDetails();
34
35     emp2.getEmployeeDetails( employeeName: "Sam ", yearOfJoining: "2000", employeeSalary: "75000", employeeAddress: "68D- WallsStreat");
36     emp2.displayEmployeeDetails();
37
38     emp3.getEmployeeDetails( employeeName: "John", yearOfJoining: "1999", employeeSalary: "55000", employeeAddress: "26B- WallsStreat");
39     emp3.displayEmployeeDetails();
40 }
41 }
42

```

Output:-

```

Employee x
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe" "-javaage
Name      Year of joining      Salary      Address
Robert    1994                  60000      64C- WallsStreat
Sam        2000                  75000      68D- WallsStreat
John      1999                  55000      26B- WallsStreat

Process finished with exit code 0

```

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

Code:-

```

Matrix.java x
2 usages
1 public class Matrix {
2
3     //declaring two matrix
4     8 usages
5     int[][] myMatrix1 = new int[2][2];
6     8 usages
7     int[][] myMatrix2 = new int[2][2];
8
9     //creating a method to find no of rows in a matrix
10    2 usages
11    void getNoOfRows(int[][] matrix){
12        System.out.println("No of Rows in Matrix is : " + matrix.length);
13    }
14
15    //creating a method to find no of cols in a matrix
16    2 usages
17    void getNoOfCols(int[][] matrix){
18        System.out.println("No of Columns in Matrix is : " + matrix[0].length);
19    }
20 }

```

```

18 //creating a method to add two Matrix
19 1 usage
20 void addTwoMatrix(int[][] matrix1,int[][] matrix2){
21
22     if(matrix1.length != matrix2.length || matrix1[0].length != matrix2[0].length){
23         System.out.println("Matrices cannot be added");
24     }
25     else{
26         int[][] resultMatrix = new int[matrix1.length][matrix1[0].length];
27
28         for(int i=0;i<matrix1.length;i++){
29             for(int j=0;j<matrix1[0].length;j++){
30                 resultMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
31             }
32         }
33
34         System.out.println("Result Matrix after Adding : ");
35         for(int i=0;i<resultMatrix.length;i++){
36             for(int j=0;j<resultMatrix[0].length;j++){
37                 System.out.print(resultMatrix[i][j] + " ");
38             }
39             System.out.println();
40         }
41     }
42 }
43

```

```

44 //creating a method to multiply two Matrix
45 1 usage
46 void multiplyTwoMatrix(int[][]matrix1, int[][] matrix2){
47     if(matrix1[0].length != matrix2.length){
48         System.out.println("Matrices cannot be multiplied");
49     }
50     else{
51         int[][] resultMatrix = new int[matrix1.length][matrix2[0].length];
52
53         for(int i=0;i<matrix1.length;i++){
54             for(int j=0;j<matrix2[0].length;j++){
55                 for(int k=0;k<matrix2.length;k++){
56                     resultMatrix[i][j] += matrix1[i][k] * matrix2[k][j];
57                 }
58             }
59         }
60
61         System.out.println("Result Matrix after Multiplying : ");
62         for(int i=0;i<resultMatrix.length;i++){
63             for(int j=0;j<resultMatrix[0].length;j++){
64                 System.out.print(resultMatrix[i][j] + " ");
65             }
66             System.out.println();
67         }
68     }
69 }
70

```

```

71 //creating a method to set data in Matrix
72 8 usages
73 void setElementAtIndex(int[][] matrix,int i, int j,int data){
74     try{
75         matrix[i][j]=data;
76     }
77     catch (IndexOutOfBoundsException e){
78         System.out.println("Exception is : " + e);
79     }
80 }
81

```

```

82 //creating a main method to create object of class
83 public static void main(String[] args){
84
85     Matrix obj = new Matrix(); //creating object of class "Matrix"
86
87     //fill the data in myMatrix1
88     obj.setElementAtIndex(obj.myMatrix1, 0, 0, data: 3);
89     obj.setElementAtIndex(obj.myMatrix1, 0, 1, data: 4);
90     obj.setElementAtIndex(obj.myMatrix1, 1, 0, data: 2);
91     obj.setElementAtIndex(obj.myMatrix1, 1, 1, data: 1);
92
93     //fill the data in myMatrix2
94     obj.setElementAtIndex(obj.myMatrix2, 0, 0, data: 1);
95     obj.setElementAtIndex(obj.myMatrix2, 0, 1, data: 5);
96     obj.setElementAtIndex(obj.myMatrix2, 1, 0, data: 3);
97     obj.setElementAtIndex(obj.myMatrix2, 1, 1, data: 7);
98 }
99

```



```

98      //get the no of rows in myMatrix1
99      obj.getNoOfRows(obj.myMatrix1);
100     //get the no of cols in myMatrix1
101     obj.getNoOfCols(obj.myMatrix1);
102
103
104     //get the no of rows in myMatrix2
105     obj.getNoOfRows(obj.myMatrix2);
106     //get the no of cols in myMatrix2
107     obj.getNoOfCols(obj.myMatrix2);
108
109     //adding two matrix
110     obj.addTwoMatrix(obj.myMatrix1,obj.myMatrix2);
111
112     //multiply two matrix
113     obj.multiplyTwoMatrix(obj.myMatrix1,obj.myMatrix2);
114 }
115 }
116

```

Output:-

```

Matrix x
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
No of Rows in Matrix is : 2
No of Columns in Matrix is : 2
No of Rows in Matrix is : 2
No of Columns in Matrix is : 2
Result Matrix after Adding :
4 9
5 8
Result Matrix after Multiplying :
15 43
5 17

Process finished with exit code 0

```

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.

Code:-

```

Student.java x
4 usages
1 public class Student {
2
3     //declaring variables
4     String studentName;
5
6     //creating default constructor without parameters
7     Student(){
8         studentName = "Unknown";
9     }
10
11    //creating parameterized constructor
12    Student(String name){
13        studentName=name;
14    }
15
16    //creating method to display name of student
17    void displayName(){
18        System.out.println("Student Name is : " + studentName);
19    }
20
21    //creating a main method to create object of class
22    public static void main(String[] args){
23        Student stu1 = new Student();    //creating object of class "Student" without arguments
24        stu1.displayName();
25
26        Student stu2 = new Student( name: "Vishal Gupta");    //creating object of class "Student" with argument.
27        stu2.displayName();
28    }
29 }
30

```

Output:-

```
Student x
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Student Name is : Unknown
Student Name is : Vishal Gupta

Process finished with exit code 0
```

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

Output:- in comment

```
MyClass.java x
1 public class MyClass {
2     private int x = 10;
3     static int m1() {
4         int y = x; //Non-static field 'x' cannot be referenced from a static context, can not be compiled
5         return y;
6     }
7     public static void main(String[] args) {
8         System.out.println(m1());
9     }
10
11
12 }
```

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

Output:- in comment

```
MyClass.java x
1 public class MyClass {
2     private int x = 10;
3     static int m1()
4     {
5         MyClass obj = new MyClass(); //by creating object of class, non-static variable can be assigned
6         int y = obj.x;
7         return y;
8     }
9     public static void main(String[] args) {
10        System.out.println(m1()); //It will print 10
11    }
12 }
```

13) What is the output

Output:- in comment

```
MyClass.java x
1 public class MyClass {
2     static int a = 20;
3     static int b = 30;
4     static int c = 40;
5     MyClass()
6     {
7         a = 200;
8     }
9     static void m1() {
10        b = 300;
11    }
12    static {
13        c = 400;
14    }
15    public static void main(String[] args) {
16        System.out.println(a); //It will print 20
17        System.out.println(b); //It will print 30
18        System.out.println(c); //It will print 400
19    }
20 }
21 }
```

14) What's the output

Output:- in comment

```
MyClass.java x
1 usage
1 public class MyClass {
2     2 usages
3     static int a = 20;
4     1 usage
5     MyClass() {
6         a = 200;
7     }
8
9     public static void main(String[] args){
10         new MyClass();
11         System.out.println(a);    //It will Print 200
12     }
13 }
```

15) What's the error in the code

Output:- in comment

```
Myclass.java x
6 usages
1 public class Myclass {
2     3 usages
3     static int a = 20;
4     3 usages
5     Myclass() {
6         a++;
7     }
8
9     void m1() {
10        1 usage
11        a++;
12        System.out.println(a);
13    }
14
15    public static void main(String[] args)
16    {
17        Myclass obj = new Myclass();    //It will increment value of a by 1
18        Myclass obj2 = new Myclass();    //It will increment value of a by 1
19        Myclass obj3 = new Myclass();    //It will increment value of a by 1
20        obj3.m1();    //It will increment value of a by 1 and then print 24
21    }
22
23 }
24
25 }
```

16) Will this program execute what will be the output

Output:- in comment

```
Test.java x
2 usages
1 public class Test {
2     1 usage
3     Test(Test t) {
4         m1();    //2. This will call M1 method first
5         System.out.println("Constructor");    //6. This will "Constructor" Third
6     }
7
8     void m1() {
9         1 usage
10        m2();    //3. This will call M2 method First
11        System.out.println("Instance method");    //5. This will "Instance method" second
12    }
13
14    static void m2() {
15        1 usage
16        System.out.println("Static method");    //4. This will print "Static method" First
17    }
18
19    public static void main(String[] args)
20    {
21        new Test(null);    //1.This will call default constructor
22    }
23 }
```

17) What's the output

Soln:- in comment

```
Figure.java
1  class Figure {
2      final int length = 5;
3      final int breadth = 4;
4      final void area() {
5          int a = length * breadth;
6          System.out.println("Area:"+a);
7      }
8  }
9  class Rectangle extends Figure {
10     final void rect()
11     {
12         System.out.println("This is rectangle");
13     }
14 }
15 final class Final_Use extends Rectangle {
16     public static void main(String[] args)
17     {
18         Final_Use obj = new Final_Use();
19         obj.rect(); //this will access rect() method of class Rectangle and print "This is rectangle"
20         obj.area(); //this will access area() method of class Figure and print "Area:20"
21     }
22 }
23 }
```

- 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

Code:-

```
Parent.java
1  public class Parent {
2      void parentMethod(){
3          System.out.println("This is parent class");
4      }
5  }
6
7  //creating child class of Parent
8  class Child extends Parent{
9
10     void childMethod(){
11         System.out.println("This is child class");
12     }
13
14     //creating main method to create object of class "Parent" and "Child".
15     public static void main(String[] args){
16         Parent parent = new Parent();
17         Child child = new Child();
18
19         parent.parentMethod(); //calling method of parent class by object of parent class
20         child.childMethod(); //calling method of child class by object of child class
21         child.parentMethod(); //calling method of parent class by object of child class
22     }
23 }
```

Output:-

```
Child x
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
This is parent class
This is child class
This is parent class

Process finished with exit code 0
```

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.

Code:-

```
Member.java x
2 usages 2 inheritors
1 public class Member {
    4 usages
2     String memberName;
    4 usages
3     int memberAge;
    4 usages
4     String memberPhoneNo;
    4 usages
5     String memberAddress;
    5 usages
6     double memberSalary;
7
8     void printSalary(){
9         System.out.println("Member Salary is : " + memberSalary);
10    }
11 }
12
13 class Employee extends Member{
14     String employeeSpecialization;
15
16 }
17
18 class Manager extends Member{
19     String managerDepartment;
20
21 }
22
```

```

23 class Main{
24     //creating main method to create object of class "Employee" and "Manager"
25     public static void main(String[] args){
26         Employee emp = new Employee();
27         Manager man = new Manager();
28
29         //assigning values to employee object
30         emp.memberName = "Rahul Yadav";
31         emp.memberAge = 23;
32         emp.memberPhoneNo = "7658876926";
33         emp.memberAddress = "Sector-62 Noida";
34         emp.memberSalary = 42000;
35
36         System.out.println("Employee Details : ");
37         System.out.println("Employee Name : " + emp.memberName);
38         System.out.println("Employee Age : " + emp.memberAge);
39         System.out.println("Employee Salary : Rs" + emp.memberSalary);
40         System.out.println("Employee PhoneNo. : " + emp.memberPhoneNo);
41         System.out.println("Employee Address : " + emp.memberAddress);
42
43         System.out.println();
44
45         //assigning values to Manager object
46         man.memberName = "Vishal Gupta";
47         man.memberAge = 22;
48         man.memberPhoneNo = "6398222157";
49         man.memberAddress = "AOC Center Hyderabad";
50         man.memberSalary = 60000;
51
52         System.out.println("Manager Details : ");
53         System.out.println("Manager Name : " + man.memberName);
54         System.out.println("Manager Age : " + man.memberAge);
55         System.out.println("Manager Salary : Rs" + man.memberSalary);
56         System.out.println("Manager PhoneNo. : " + man.memberPhoneNo);
57         System.out.println("Manager Address : " + man.memberAddress);
58
59     }
60 }
61
62

```

Output:-

```

Main x
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Employee Details :
Employee Name : Rahul Yadav
Employee Age : 23
Employee Salary : Rs42000.0
Employee PhoneNo. : 7658876926
Employee Address : Sector-62 Noida

Manager Details :
Manager Name : Vishal Gupta
Manager Age : 22
Manager Salary : Rs60000.0
Manager PhoneNo. : 6398222157
Manager Address : AOC Center Hyderabad

Process finished with exit code 0

```

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.

Code:-

```
Rectangle.java 3 usages 1 inheritor
1 public class Rectangle {
2
3     //declaring variables
4     int length, breadth;
5
6     //declaring and initializing the Parameterized constructor
7     Rectangle(int length,int breadth){
8         this.length = length;
9         this.breadth = breadth;
10    }
11
12    //Creating method to calculate and print Area of Rectangle
13    void area(){
14        System.out.println("Area with sides " + length + " and " + breadth + " is : " + length*breadth + " units sq" );
15    }
16
17    //Creating method to calculate and print Perimeter of Rectangle
18    void perimeter(){
19        System.out.println("Perimeter with sides " + length + " and " + breadth + " is : " + 2*(length+breadth) + " units");
20    }
21
22 }
23
24
25 // creating child class of parent class "Rectangle"
26 class Square extends Rectangle{
27     float side;
28     Square(int side){
29         super(side,side);
30     }
31
32 //creating main method to create object of class
33 public static void main(String[] args){
34
35     //creating object for class "Rectangle"
36     Rectangle rectangle = new Rectangle( length: 12, breadth: 15);
37     rectangle.area();
38     rectangle.perimeter();
39
40     System.out.println();
41
42     //creating object for class "Square"
43     Square square = new Square( side: 15);
44     square.area();
45     square.perimeter();
46 }
47
48 }
```

Output:-

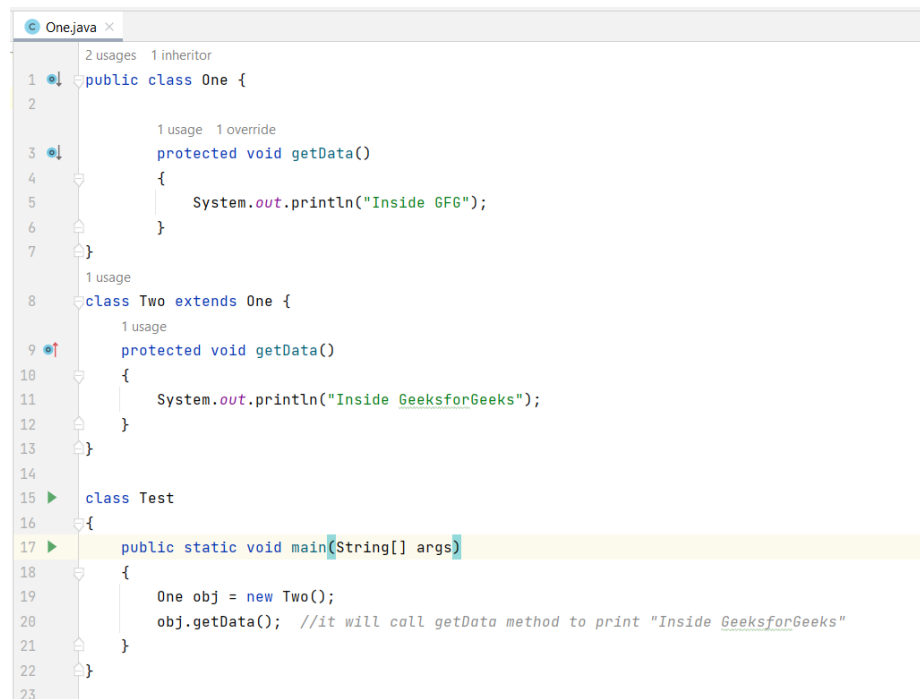
```
Square
"C:\Program Files\Amazon Corretto\jdk11.0.15_9\bin\java.exe"
Area with sides 12 and 15 is : 180 units sq
Perimeter with sides 12 and 15 is : 54 units

Area with sides 15 and 15 is : 225 units sq
Perimeter with sides 15 and 15 is : 60 units

Process finished with exit code 0
```

21) What's the output

Soln:- in comment



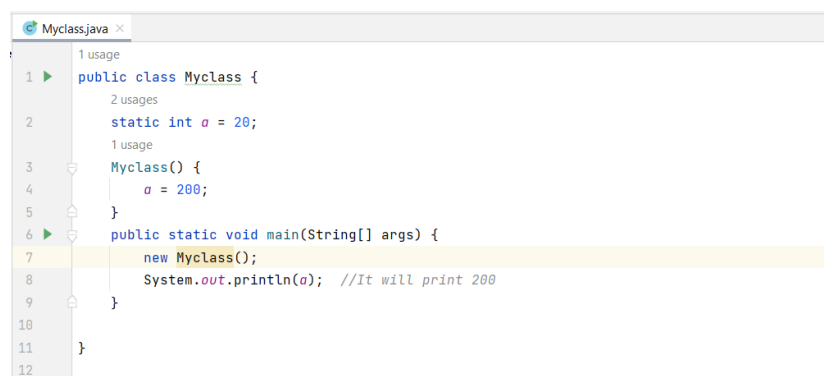
```
1 public class One {
2
3     protected void getData()
4     {
5         System.out.println("Inside GFG");
6     }
7 }
8 class Two extends One {
9     protected void getData()
10    {
11        System.out.println("Inside GeeksforGeeks");
12    }
13 }
14
15 class Test
16 {
17     public static void main(String[] args)
18     {
19         One obj = new Two();
20         obj.getData(); //it will call getData method to print "Inside GeeksforGeeks"
21     }
22 }
23
```

22) Can we overload main() method?

Soln:- Yes, we can overload main method but JVM only call the original main method not the overloaded.

23) What is the output ?

Soln:- in comment



```
1 public class MyClass {
2     static int a = 20;
3     MyClass() {
4         a = 200;
5     }
6     public static void main(String[] args) {
7         new MyClass();
8         System.out.println(a); //It will print 200
9     }
10
11 }
12
```


24) What will be the output:

Soln:- in comment

```
Ajava x
2 usages
1 public class A {
    1 usage
2     void sum(int x, int y){
3         System.out.println("Sum of two numbers: " + (x+y));
4     }
    1 usage
5     void sum(int x, int y, int z){
6         System.out.println("Sum of three numbers: " + (x+y+z));
7     }
8     public static void main(String[] args){
9         A a = new A();
10        a.sum(x: 20, y: 30);           //this will call Method sum with two parameters
11                                       //and print "Sum of two numbers: 50"
12
13        a.sum(x: 30, y: 40, z: 50);   //this will call Method sum with three parameters
14                                       //and print "Sum of two numbers: 120"
15    }
16
17 }
```

25) What is the output

Soln:- in comment

```
Ajava x
2 usages
1 public class A {
2     void sum(int x, int y) {
3         System.out.println("Sum of two numbers: " + (x + y));
4     }
5
6     void sum(int y, int x) {
7         System.out.println("Sum of three numbers: " + (x + y));
8     }
9
10    public static void main(String[] args) {
11        A a = new A();
12        a.sum(20, 30);
13    }
14
15 }
16
17 //OutPut :- since sum method is already defined in Class A, cannot define another method with same name and parameters
18 //This will so error
```

26) What is the output?

Soln:- in comment

```
Ajava x
6 usages 1 inheritor
1 public class A {
    5 usages 1 override
2     void m1(A a){
3         System.out.println("m1 method in class A");
4     }
5 }
    3 usages
6 class B extends A {
    5 usages
7     public void m1(A a){
8         System.out.println("m1 method in class B");
9     }
10 }
11 class Test{
12     public static void main(String[] args){
13         A a = new A();
14         a.m1(a);           //this will print "m1 method in class A"
15         a.m1(new B());     //this will print "m1 method in class A"
16
17         B b = new B();
18         b.m1(a: null);     //this will print "m1 method in class B"
19
20         a = b;
21         a.m1(a: null);     //this will print "m1 method in class B"
22         a.m1(new A());     //this will print "m1 method in class B"
23     }
24
25 }
```

27) What is the output?

Soln:- in comment

```
ExceptionInterviewQuestion_01.java
1 import java.io.FileNotFoundException;
2 import java.io.IOException;
3 import java.sql.SQLException;
4
5 public class ExceptionInterviewQuestion_01 {
6     public static void main(String[] args) {
7         try {
8             test(); //this will print "Inside test() method"
9         } catch (IOException e) {
10             e.printStackTrace();
11         } catch (FileNotFoundException e) {
12             e.printStackTrace();
13         } catch (SQLException e) {
14             e.printStackTrace();
15         }
16     }
17
18     1 usage
19     public static void test() throws IOException, SQLException, FileNotFoundException{
20         System.out.println("Inside test() method");
21     }
22
23 }
24
```

28) What is the Output.

Soln:- in comment

```
TestException3.java
1 public class TestException3{
2     public static void main(String[] args) {
3         try{
4             bar(); //this will not print anything because method contain nothing to print neither throw any exception
5         }catch (NullPointerException e){
6             e.printStackTrace();
7         }catch (Exception e){
8             e.printStackTrace();
9         }
10        foo(); //this will not print anything because method contain nothing to print neither throw any exception
11    }
12
13    1 usage
14    public static void bar(){
15    }
16
17    1 usage
18    public static void foo() throws NullPointerException{
19    }
20
21 }
22
23
```

Thank You 😊