



Devang Patel Institute of
Advance Technology and Research
(A Constituent Institute of CHARUSAT)

Certificate

This is to certify that

Mr./Mrs. Vridha Ha Devkumar Patelbhai

of DEPSTAR - CSE2 Class,

ID. No. 23DCS137 has satisfactorily completed

his/ her term work in CSE-201 JAVA PROGRAMMING for

the ending in Oct 2024 /2025

Date :

Sign. of Faculty

Head of Department

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH**

Department of Computer Science & Engineering

Subject Name: JAVA PROGRAMMING**Semester: 3****Subject Code: CSE201****Academic year: 2024-25****Part - 1**

No.	Aim of the Practical
2.	<p>Imagine you are developing a simple banking application where you need to display the current balance of a user account. For simplicity, let's say the current balance is \$20. Write a java program to store this balance in a variable and then display it to the user.</p> <p><u>PROGRAM CODE :</u></p> <pre>class banking { public static void main(String args[]){ int currentbalance=20; System.out.println("Your Balance is:" + currentbalance); } }</pre> <p><u>OUTPUT:</u></p> <hr/> <pre>C:\Windows\System32\cmd.exe Microsoft Windows [Version 10.0.19045.4598] (c) Microsoft Corporation. All rights reserved. C:\Users\Lenovo\Desktop\JAVA\SET 1>javac banking.java C:\Users\Lenovo\Desktop\JAVA\SET 1>java banking Your Balance is:20</pre>

CONCLUSION:

From this practical I have learned about writing a basic Java program, including declaring a class, defining the main method, initializing variables, and printing output.

3. Write a program to take the user for a distance (in meters) and the time taken (as three numbers: hours, minutes, seconds), and display the speed, in meters per second, kilometers per hour and miles per hour (hint:1 mile = 1609 meters).

PROGRAM CODE:

```
import java.util.Scanner;

public class speed
{
    public static void main(String args[])
    {
        int h, M, s;
        int distance;
        Scanner S1 = new Scanner(System.in);

        System.out.println("Enter the distance travelled by user in meter:");
        distance = S1.nextInt();

        System.out.println("Enter the time in hours:");
        h = S1.nextInt();

        System.out.println("Enter the time in minutes:");
        M = S1.nextInt();

        System.out.println("Enter the time in seconds:");
        s = S1.nextInt();

        int totalSeconds = (h * 3600) + (M * 60) + s;

        if (totalSeconds > 0)
        {
            double Mpersec = (double) distance / totalSeconds;
```

```
double KmPerHr = Mpersec * 3.6;  
  
double MilesPerHr = Mpersec * (3.6/1.609);  
  
System.out.printf("The speed of the vehicle is %.2f m/s%n", Mpersec);  
System.out.printf("The speed of the vehicle is %.2f km/hr%n", KmPerHr);  
System.out.printf("The speed of the vehicle is %.2f miles/hr%n", MilesPerHr);}  
else {  
System.out.println("Invalid time. Time should be greater than 0.");  
}  
}  
}  
}
```

OUTPUT:

```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 10.0.19045.4598]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\Lenovo\Desktop\JAVA\SET 1>javac speed.java  
  
C:\Users\Lenovo\Desktop\JAVA\SET 1>java speed  
Enter the distance travelled by user in meter:  
40000  
Enter the time in hours:  
1  
Enter the time in minutes:  
25  
Enter the time in seconds:  
12  
The speed of the vehicle is 7.82 m/s  
The speed of the vehicle is 28.17 km/hr  
The speed of the vehicle is 17.51 miles/hr
```

CONCLUSION:

This Java program calculates the speed of a vehicle based on user-provided distance and time inputs. It ensures accuracy by validating time inputs and provides speed outputs in meters per second, kilometers per hour, and miles per hour, making it practical for real-world applications requiring precise speed measurements.

4. Imagine you are developing a budget tracking application. You need to calculate the total expenses for the month. Users will input their daily expenses, and the program should compute the sum of these expenses. Write a Java program to calculate the sum of elements in an array representing daily expenses.

PROGRAM CODE:

```
import java.util.Scanner;
public class Budget {
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        int NumDays;
        double totalExpenses = 0.0;

        System.out.println("Enter the number of days in the month: ");
        NumDays = scanner.nextInt();

        for (int i = 1; i <= NumDays; i++) {
            System.out.println("Enter expenses for day : $" + i);
            double dailyExpense = scanner.nextDouble();
            totalExpenses += dailyExpense;
        }
        scanner.close();
        System.out.printf("Total expenses for the month: $%.2f\n", totalExpenses);
    }
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.19045.4598]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo\Desktop\JAVA\SET 1>javac Budget.java

C:\Users\Lenovo\Desktop\JAVA\SET 1>java Budget
Enter the number of days in the month:
4
Enter expenses for day : $1
95
Enter expenses for day : $2
45
Enter expenses for day : $3
32
Enter expenses for day : $4
65
Total expenses for the month: $237.00
```

Conclusion:

In this program, takes an array to input the daily expenses from the user and printing the sum of daily expenses.

5. An electric appliance shop assigns code 1 to motor,2 to fan,3 to tube and 4 for wires. All other items have code 5 or more. While selling the goods, a sales tax of 8% to motor,12% to fan,5% to tube light,7.5% to wires and 3% for all other items is charged. A list containing the product code and price in two different arrays. Write a java program using switch statement to prepare the bill.

PROGRAM CODE:

```
import java.util.Scanner;
public class shop {
    public static void main(String args[]) {

        Scanner S1 = new Scanner(System.in);

        System.out.println("Enter the product code:");
        int productCode = S1.nextInt();

        System.out.println("Enter the price:");
        double price = S1.nextDouble();
```

```
double tax = 0;
switch (productCode) {
    case 1:
        tax = price * 0.08;
        break;
    case 2:
        tax = price * 0.12;
        break;
    case 3:
        tax = price * 0.05;
        break;
    case 4:
        tax = price * 0.075;
        break;
    default:
        tax = price * 0.03;
}

double totalPrice = price + tax;
System.out.println("Price: " + price);
System.out.println("Tax: " + tax);
System.out.println("Total Price: " + totalPrice);
}
```

OUTPUT:

```
C:\Windows\System32\cmd.exe
C:\Users\Lenovo\Desktop\JAVA\SET 1>javac shop.java

C:\Users\Lenovo\Desktop\JAVA\SET 1>java shop
Enter the product code:
1
Enter the price:
500
Price: 500.0
Tax: 40.0
Total Price: 540.0

C:\Users\Lenovo\Desktop\JAVA\SET 1>java shop
Enter the product code:
2
Enter the price:
350
Price: 350.0
Tax: 42.0
Total Price: 392.0

C:\Users\Lenovo\Desktop\JAVA\SET 1>java shop
Enter the product code:
4
Enter the price:
920
Price: 920.0
Tax: 69.0
Total Price: 989.0
```

CONCLUSION:

Using a switch case in a Java program to handle different tax scenarios provides a structured and efficient way to calculate the final amount after adding taxes.

- 6** Create a Java program that prompts the user to enter the number of days (n) for which they want to generate their exercise routine. The program should then calculate and display the first n terms of the Fibonacci series, representing the exercise duration for each day.

PROGRAM CODE:

```
import java.util.Scanner;
class fibonacci
{
    public static void main(String args[])
    {
        long sum = 0;
        long c;
```

```
System.out.println("Enter the number of terms:");
Scanner S = new Scanner(System.in);
int n = S.nextInt();
long a = 0;
long b = 1;
System.out.print("Fibonacci Series: ");
for (int i = 1; i <= n; i++)
{
    System.out.print(a + " ");
    sum += a;
    c = a + b;
    a = b;
    b = c;
}
System.out.println("\nSum of Fibonacci Series: " + sum);
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.19045.4598]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo\Desktop\JAVA\SET 1>javac fibonacci.java

C:\Users\Lenovo\Desktop\JAVA\SET 1>java fibonacci
:Enter the number of terms:
9
:Fibonacci Series: 0 1 1 2 3 5 8 13 21
:Sum of Fibonacci Series: 54
```

CONCLUSION:

In this Java program, the concepts of the Fibonacci series and the sum of the series are implemented to demonstrate sequence generation and summation techniques. By computing the Fibonacci series, the program illustrates how each number is the sum of the two preceding ones, starting from 0 and 1.

Part - 2

No.	Aim of the Practical
7.	<p>Given a string and a non-negative int n, we'll say that the front of the string is the first 3 chars, or whatever is there if the string is less than length 3. Return n copies of the front;</p> <pre>front_times('Chocolate', 2) → 'ChoCho' front_times('Chocolate', 3) → 'ChoChoCho' front_times('Abc', 3) → 'AbcAbcAbc'</pre> <p><u>PROGRAM CODE:</u></p> <pre>public class PRACT7 { static int Choco(String s, int a) { String S1 = s.substring(0, 3); for (int i = 0; i < a; i++) { System.out.print(S1); } System.out.println(); return 0; } public static void main(String args[]) { Choco("Chocolate", 3); Choco("Chocolate", 2); Choco("Abc", 3); } }</pre> <p><u>OUTPUT:</u></p> <pre>C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 2>javac PRACT7.java C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 2>java PRACT7 ChoChoCho ChoCho AbcAbcAbc C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 2></pre>

CONCLUSION:

In this Java Program we have used the concept of substring and repeating that string for n times.

8. Given an array of ints, return the number of 9's in the array. `array_count9([1, 2, 9]) → 1`
`array_count9([1, 9, 9]) → 2` `array_count9([1, 9, 9, 3, 9]) → 3`

PROGRAM CODE :

```
public class PRACT8{

    public static void main(String args[]) {

        int[] arr1 = { 1, 2, 9 };

        int[] arr2 = { 1, 9, 9 };

        int[] arr3 = { 1, 9, 9, 3, 9 };

        System.out.println(arrayCount9(arr1));

        System.out.println(arrayCount9(arr2));

        System.out.println(arrayCount9(arr3));

    }

    public static int arrayCount9(int[] a) {

        int count = 0;

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

            if (a[num] == 9) {

                count++;

            }

        }

    }

}
```

```
    }  
  
    return count;  
  
}  
  
}
```

OUTPUT:

```
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 2>javac PRACT8.java  
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 2>java PRACT8  
1  
2  
3
```

CONCLUSION:

In this Java Program we have applied the logic for counting the no. of two's that we have entered in the string.

- 9 Given a string, return a string where for every char in the original, there are two chars.
`double_char('The') → 'TThhee'` `double_char('AAbb') → 'AAAAAbbbb'` `double_char('Hi-There') → 'HHii--TThheerree'`

PROGRAM CODE:

```
public class PRACT9 {

    public static void main(String[] args) {
        System.out.println(doubleChar("The"));
        System.out.println(doubleChar("AAbb"));
        System.out.println(doubleChar("Hi-There"));
    }

    public static String doubleChar(String str) {
        String doubledStr = ""; // Initialize an empty string to store the result

        for (int i = 0; i < str.length(); i++) {
            char c = str.charAt(i); // Get the current character

            doubledStr += c + "" + c;
        }

        return doubledStr;
    }
}
```

OUTPUT:

```
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 2>javac PRACT9.java
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 2>java PRACT9
TThhee
AAAAAbbbb
HHii--TThheerree
```

CONCLUSION:

In this java program we have applied logic to double every character of the string and print it.

- 10** Perform following functionalities of the string:
- Find Length of the String
 - Lowercase of the String
 - Uppercase of the String
 - Reverse String

PROGRAM CODE:

```
public class PRACT10
{
    public static void main(String args[])
    {

        String S1 = "Charusat";
        System.out.println(S1.toLowerCase());

        String S2 = "charusat";
        System.out.println(S2.toUpperCase());

        String S3 = "HelloWorld!";
        System.out.println(S3.length());

        String S4 = "Bhalodia Veer";
        System.out.println(new StringBuilder(S4).reverse().toString());
    }
}
```

OUTPUT:

```
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 2>javac PRACT10.java
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 2>java PRACT10
charusat
CHARUSAT
11
reeV aidolahB
```

CONCLUSION:

In this java program we have applied logic to find the length of the string , convert the string to lowercase , convert to uppercase , and reverse the entered string.

- 11 Perform following Functionalities of the string: “CHARUSAT UNIVERSITY”
- Find length
 - Replace ‘H’ by ‘FIRST LETTER OF YOUR NAME’
 - Convert all character in lowercase

PROGRAM CODE:

```
public class PRACT11
{
    public static void main(String[] args) {

        String Str="CHARUSAT UNIVERSITY";
        System.out.println("To Lower case:" + Str.toLowerCase());

        System.out.println("Length of String:" + Str.length());

        System.out.println("After H is replaced:" + Str.replace('H', 'V'));
    }
}
```

OUTPUT:

```
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 2>javac PRACT11.java
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 2>java PRACT11
To Lower case:charusat university
Length of String:19
After H is replaced:CVARUSAT UNIVERSITY
```

CONCLUSION:

In this java program we have applied the logic to replace the character of the entered string to some other character .

Part - 3

No.	Aim of the Practical
12.	<p>Imagine you are developing a currency conversion tool for a travel agency. This tool should be able to convert an amount in Pounds to Rupees. For simplicity, we assume the conversion rate is fixed: 1 Pound = 100 Rupees. The tool should be able to take input both from command-line arguments and interactively from the user.</p> <p><u>PROGRAM CODE :</u></p> <pre>import java.util.*; class PRACT12 { public static void main(String[] args) { int a= Integer.parseInt(args[0]); int c=a*100; System.out.println("Currency in a rupees=" + c); Scanner S1 = new Scanner(System.in); int R=S1.nextInt(); int P=R*100; System.out.println("Currency in Rupees=" + P); } }</pre> <p><u>OUTPUT:</u></p> <pre>Microsoft Windows [Version 10.0.19045.4717] (c) Microsoft Corporation. All rights reserved. C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 3>javac PRACT12.java C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 3>java PRACT12 15 Currency in a rupees=1500 23 Currency in Rupees=2300</pre>

CONCLUSION:

It converts a given amount to a different currency unit by multiplying it with a fixed conversion rate (100). The program utilizes command-line arguments for initial input and the Scanner class for runtime input from the user.

13. Create a class called Employee that includes three pieces of information as instance variables—a first name (type String), a last name (type String) and a monthly salary (double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10% raise and display each Employee's yearly salary

PROGRAM CODE :

```
import java.util.Scanner;

class Employee {

    private String fn;
    private String ln;
    private double salary;

    Scanner s = new Scanner(System.in);

    Employee() {
    }

    Employee(String fn, String ln, double salary) {
        this.fn = fn;
        this.ln = ln;
    }

    void setSalary(double salary) {
        if (salary <= 0.0)
            this.salary = 0.0;
        else
            this.salary = salary;
    }

    double getSalary() {
        return this.salary;
    }

    void setFn(String fn) {
        this.fn = fn;
    }

    String getFn() {
        return this.fn;
    }

    void setLn(String ln) {
        this.ln = ln;
    }

    String getLn() {
        return this.ln;
    }
}
```

```
this.salary = salary;  
}  
  
public void setfn() {  
  
    System.out.print("Enter employee first name :");  
  
    fn = s.next();  
}  
  
public void setln() {  
  
    System.out.print("Enter employee last name :");  
  
    ln = s.next();  
}  
  
public void setsalary() {  
  
    System.out.print("Enter employee salary :");  
  
    salary = s.nextDouble();  
  
    if(salary<0){  
  
        salary=0;  
    }  
  
    else{  
  
        salary=(salary*12)+(salary*12)*0.1;  
    } }  
  
public String getfn() {
```

```
return fn;  
}  
  
public String getln() {  
  
    return ln;  
}  
  
public double getsalary() {  
  
    return salary;  
}  
}  
  
public class PRACT13 {  
  
    public static void main(String[] args) {  
  
        Employee e1=new Employee();  
  
        Employee e2=new Employee();  
  
        e1.setfn();  
  
        e1.setln();  
  
        e1.setsalary();  
  
        e2.setfn();  
  
        e2.setln();  
  
        e2.setsalary();  
  
        System.out.print(e1.getfn()+" ");  
  
        System.out.println(e1.getln());
```

```
System.out.println(e1.getsalary());  
  
System.out.print(e2.getfn()+" ");  
  
System.out.println(e2.getln());  
  
System.out.println(e2.getsalary());  
  
}  
  
}
```

OUTPUT:

```
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 3>javac PRACT13.java  
  
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 3>java PRACT13  
Enter employee first name :Veer  
Enter employee last name :Bhalodia  
Enter employee salary :5000  
Enter employee first name :Yash  
Enter employee last name :Khanpara  
Enter employee salary :6000  
Veer Bhalodia  
66000.0  
Yash Khanpara  
79200.0
```

CONCLUSION:

This `Employee` class in Java demonstrates encapsulation by using private fields for first name, last name, and salary, with a constructor for initialization and public methods for controlled access and modification, ensuring data hiding and integrity.

- 14** Create a class called Date that includes three pieces of information as instance variables—a month (type int), a day (type int) and a year (type int). Your class should have a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date's capabilities.

PROGRAM CODE :

```
import java.util.Scanner;
class Date

{
int year;
int month;
int day;
Scanner S1 = new Scanner(System.in);

void getDate()
{
System.out.println("Enter the Date:");

day=S1.nextInt();
}

void getMonth()
{
System.out.println("Enter the Month:");
month=S1.nextInt();
}

void getYear()
{
System.out.println("Enter the Year:");
year=S1.nextInt();
}

void displayDate()
{
System.out.println( "Your output is:" + day+"/"+month+"/"+year);
}

Date(int d,int m ,int y)
{
```

```
day=d;
month=m;
year=y;
}
Date()
{
}
}
class PRACT14
{
public static void main(String[] args)
{

Date d2 = new Date(29,10,2004);
Date d1 = new Date(0,0,0);
d1.getDate();
d1.getMonth();
d1.getYear();
d1.displayDate();
d2.displayDate();
}
}
```

OUTPUT:

```
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 3>javac PRACT14.java
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 3> java PRACT14
Enter the Date:
14
Enter the Month:
6
Enter the Year:
2005
Your output is:14/6/2005
Your output is:29/10/2004
```

CONCLUSION:

This Java program demonstrates OOP fundamentals by defining a Date class with a constructor for initialization and a method to display the date. The main method creates an instance of Date and calls displayDate.

- 15 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.

PROGRAM CODE :

```
import java.util.Scanner;
class Area {
    double length;
    double breadth;

    Area(double l, double b) {
        length = l;
        breadth = b;
    }
    double returnArea() {
        return length * breadth;
    }
}
class PRACT15 {
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter length:");
        double length = scanner.nextDouble();
        System.out.println("Enter breadth:");
        double breadth = scanner.nextDouble();
        Area A = new Area(length, breadth);
        System.out.println(A.returnArea());
    }
}
```

OUTPUT:

```
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 3>javac PRACT15.java
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 3> java PRACT15
Enter length:
9
Enter breadth:
8
72.0
```

CONCLUSION:

It includes user input handling with the Scanner class to dynamically receive length and breadth values. An Area class instance is created with these values, and its method returnArea() calculates and returns the area of a rectangle.

- 16** 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.

PROGRAM CODE :

```
import java.util.Scanner;

class complex {

    int r, i;
    int sumi, sumr;
    Scanner s = new Scanner(System.in);

    void setr() {
        System.out.print("enter real number :");
        r = s.nextInt();
    }

    void seti() {
        System.out.print("enter imaginary number :");
        i = s.nextInt();
    }
```

```
void sum(complex c) {  
sumi = i + c.i;  
sumr = r + c.r;  
System.out.println(sumr + "+" + sumi + "i");  
}  
  
void difrence(complex c) {  
int difi = i - c.i;  
int difr = r - c.r;  
if (difi >= 0) {  
System.out.println(difr + "+" + difi + "i");  
} else {  
System.out.println(difr + "" + difi + "i");  
}  
}  
void product(complex c){  
  
int pror=(r*c.r)-(i*c.i);  
int proi=(r*c.i)+(i*c.r);  
if(proi>=0)  
System.out.println(pror+"+"+proi+"i");  
else  
System.out.println(pror+" "+proi+"i");  
}  
  
}  
public class PRACT16 {  
  
public static void main(String[] args) {  
complex c1 = new complex();  
complex c2 = new complex();  
c1.setr();  
c1.seti();  
c2.setr();  
c2.seti();  
c1.sum(c2);  
c1.difrence(c2);  
c1.product(c2);  
}  
}
```

OUTPUT:

```
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 3>javac PRACT16.java
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 3>java PRACT16
enter real number :4
enter imaginary number :5
enter real number :9
enter imaginary number :2
13+7i
-5+3i
26+53i
```

CONCLUSION:

This Java program defines a `complex` class to handle complex numbers, including methods for setting real and imaginary parts and summing two complex numbers. The main method creates two `complex` objects, sets their values, and sums them, demonstrating basic OOP principles.

Part - 4

No.	Aim of the Practical
17.	<p>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 1 - method of parent class by object of parent.</p> <p><u>PROGRAM CODE :</u></p> <pre> class Parent{ void printparent(){ System.out.println("This is Parent class"); } } class Child extends Parent{ void printchild(){ System.out.println("This is Child class"); } } public class PRACT17{ public static void main(String[] args) { Parent parent = new Parent(); parent.printparent(); Child child = new Child(); child.printchild(); } } </pre> <p><u>OUTPUT:</u></p> <pre> C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 4>javac PRACT17.java C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 4>java PRACT17 This is Parent class This is Child class </pre> <p><u>CONCLUSION:</u> In this java code we have applied a concept of single inheritance.</p>

18. 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.

PROGRAM CODE:

```
import java.util.Scanner;

class Employee {

    private String fn;
    private String ln;
    private double salary;
    Scanner s = new Scanner(System.in);

    Employee() {
    }

    public void setfn() {
        System.out.print("Enter employee first name :");
        fn = s.next();
    }

    public void setln() {
        System.out.print("Enter employee last name :");
        ln = s.next();
    }

    public void setsalary() {
        System.out.print("Enter employee salary :");
    }
}
```

```
salary = s.nextDouble();
if(salary<0){
    salary=0;
}
else{
    salary=(salary*12)+(salary*12)*0.1;
}

}

public String getfn() {
return fn;
}

public String getln() {
return ln;
}

public double getsalary() {
return salary;
}

}

public class PRACT18 {

public static void main(String[] args) {
Employee e1=new Employee();
Employee e2=new Employee();
e1.setfn();
e1.setln();
e1.setsalary();
e2.setfn();
e2.setln();
e2.setsalary();

System.out.print(e1.getfn()+" ");
System.out.println(e1.getln());
System.out.println(e1.getsalary());
```

```
System.out.print(e2.getfn()+" ");
System.out.println(e2.getln());
System.out.println(e2.getsalary());

}
}
```

OUTPUT:

```
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 4>javac PRACT18.java
C:\Users\Lenovo\Desktop\JAVA PRACTICALS\SET 4>java PRACT18
Enter employee first name :Veer
Enter employee last name :Bhalodia
Enter employee salary :50000
Enter employee first name :Yash
Enter employee last name :Khanpara
Enter employee salary :60000
Veer Bhalodia
660000.0
Yash Khanpara
792000.0
```

CONCLUSION:

In this java code we have implemented a concept of inheritance to take input from the user About the details of employee and manager and displaying all information.