

Name	Virinchi Sadashiv Shettigar
UID no.	2021300118
Experiment No.	6

AIM:	Polymorphism: To demonstrate method overriding.																																
Program 1																																	
PROBLEM STATEMENT:	<p>Consider a scenario where Bank is a class that provides functionality to get the rate of interest. However, the rate of interest varies according to banks. For example, SBI, ICICI and AXIS banks are given below.</p> <table> <tr> <td></td><td>SBI Interest Rate (Rates in % per annum)</td></tr> <tr> <td>Period</td><td><Rs. 2 Cr</td></tr> <tr> <td>7–14 Days</td><td>3.00</td></tr> <tr> <td>15 –30 Days</td><td>3.00</td></tr> <tr> <td>31-45 Days</td><td>3.00</td></tr> <tr> <td>46 -90 Days</td><td>4.05</td></tr> <tr> <td>91–120 Days</td><td>4.10</td></tr> <tr> <td>121-180 Days</td><td>4.10</td></tr> </table> <table> <tr> <td></td><td>ICICI Interest Rate (Rates in % per annum)</td></tr> <tr> <td>Period</td><td><Rs. 2 Cr</td></tr> <tr> <td>7–14 Days</td><td>3.10</td></tr> <tr> <td>15 –30 Days</td><td>3.20</td></tr> <tr> <td>31-45 Days</td><td>3.50</td></tr> <tr> <td>46 -90 Days</td><td>4.50</td></tr> <tr> <td>91–120 Days</td><td>4.70</td></tr> <tr> <td>121-180 Days</td><td>4.90</td></tr> </table>		SBI Interest Rate (Rates in % per annum)	Period	<Rs. 2 Cr	7–14 Days	3.00	15 –30 Days	3.00	31-45 Days	3.00	46 -90 Days	4.05	91–120 Days	4.10	121-180 Days	4.10		ICICI Interest Rate (Rates in % per annum)	Period	<Rs. 2 Cr	7–14 Days	3.10	15 –30 Days	3.20	31-45 Days	3.50	46 -90 Days	4.50	91–120 Days	4.70	121-180 Days	4.90
	SBI Interest Rate (Rates in % per annum)																																
Period	<Rs. 2 Cr																																
7–14 Days	3.00																																
15 –30 Days	3.00																																
31-45 Days	3.00																																
46 -90 Days	4.05																																
91–120 Days	4.10																																
121-180 Days	4.10																																
	ICICI Interest Rate (Rates in % per annum)																																
Period	<Rs. 2 Cr																																
7–14 Days	3.10																																
15 –30 Days	3.20																																
31-45 Days	3.50																																
46 -90 Days	4.50																																
91–120 Days	4.70																																
121-180 Days	4.90																																

Period	AXIS Interest Rate (Rates in % per annum)
	<Rs. 2 Cr

7–14 Days	3.15
-----------	------

15 –30 Days	3.15
-------------	------

31-45 Days	3.45
------------	------

46 -90 Days	4.05
-------------	------

91–120 Days	4.70
-------------	------

121-180 Days	5.00
--------------	------

Aayush has deposited Rs. 10000 in SBI Bank, Rs. 12500 in ICICI Bank, and Rs. 20000 in AXIS bank respectively for a particular month.

You need to print the money he will get by applying the rate of interest as per the bank and days.

Create a class 'Bank' with a method 'get_rate_of_interest' which returns 2%.

Make three subclasses named SBI_Bank, 'ICICI_Bank' and 'AXIS_bank' with a method with the same name 'get_rate_of_interest' which returns the rate of interest.

Also, give the final amount Ayush will get from that particular bank by applying the rate of interest and period. Use Calendar Class to count the number of days and amount he will get after maturity with the date of Maturity, if he deposits today.

Note:

1. Use compound interest
2. Get time period from the user
3. Solve using method overriding

PROGRAM:

```
import java.util.*;

class Bank {
    int get_rate_of_interest() {
        int interest = 2;
        return 2;
    }
}

class SBI_Bank extends Bank {
    double get_rate_of_interest(int t) {
        if (t >= 7 && t <= 14) {
            return 3;
        }
        if (t >= 15 && t <= 30) {
            return 3;
        }
        if (t >= 31 && t <= 45) {
            return 3;
        }
        if (t >= 46 && t <= 90) {
            return 4.05;
        }
        if (t >= 91 && t <= 120) {
            return 4.10;
        }
        if (t >= 121 && t <= 180) {
            return 4.10;
        }
        return 2;
    }
}

class ICICI_Bank extends Bank {
    double get_rate_of_interest(int t) {
        if (t >= 7 && t <= 14) {
            return 3.1;
        }
        if (t >= 15 && t <= 30) {
            return 3.2;
        }
        if (t >= 31 && t <= 45) {
```

```

        return 3.5;
    }
    if (t >= 46 && t <= 90) {
        return 4.5;
    }
    if (t >= 91 && t <= 120) {
        return 4.7;
    }
    if (t >= 121 && t <= 180) {
        return 4.9;
    }
    return 2;
}
}

class AXIS_Bank extends Bank {
    double get_rate_of_interest(int t) {
        if (t >= 7 && t <= 14) {
            return 3.15;
        }
        if (t >= 15 && t <= 30) {
            return 3.15;
        }
        if (t >= 31 && t <= 45) {
            return 3.45;
        }
        if (t >= 46 && t <= 90) {
            return 4.05;
        }
        if (t >= 91 && t <= 120) {
            return 4.70;
        }
        if (t >= 121 && t <= 180) {
            return 5;
        }
        return 2;
    }
}

class Calender {
    int date = 21;
    int month = 6;

```

```

void finaldate(int t) {
    int d = t % 30;
    int m = t / 30;
    date = d + date;
    month = m + month;
    if (date >= 30) {
        date = date - 30;
        month = month + 1;
    }
    System.out.println(date + " / " + month + " / 22");
}

}

public class deposit {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int t1, t2, t3;
        SBI_Bank sbi = new SBI_Bank();
        ICICI_Bank icici = new ICICI_Bank();
        AXIS_Bank axis = new AXIS_Bank();
        Calender l1 = new Calender();
        Calender l2 = new Calender();
        Calender l3 = new Calender();
        System.out.print("Enter time period for SBI ");
        t1 = sc.nextInt();
        System.out.println( "Final value $" + (10000 * Math.pow((1 +
(sbi.get_rate_of_interest(t1) / 100)), (float) t1 / 365)));
        l1.finaldate(t1);

        System.out.print("Enter time period for ICICI ");
        t2 = sc.nextInt();
        System.out.println( "Final value $" + (12500 * Math.pow((1 +
(icici.get_rate_of_interest(t2) / 100)), (float) t2 / 365)));
        l2.finaldate(t2);

        System.out.print("Enter time period for AXIS ");
        t3 = sc.nextInt();
        System.out.println( "Final value $" + (20000 * Math.pow((1 +
(axis.get_rate_of_interest(t3) / 100)), (float) t3 / 365)));
        l3.finaldate(t3);
    }
}

```

RESULT:

```
PS V:\Java Practicals- SPIT\Experiment-  
Enter time period for SBI 48  
Final value $10052.34654690067  
9 / 8 / 22  
Enter time period for ICICI 60  
Final value $12590.773661585647  
21 / 8 / 22  
Enter time period for AXIS 120  
Final value $20304.290050439082  
21 / 10 / 22  
PS V:\Java Practicals- SPIT\Experiment-
```

Program 2**PROBLEM
STATEMENT:**

Ankit works at ABC Company. He noticed that different roles(positions) have different salaries and bonuses.

The 1st Role is an 'Intern' which has 3/4th of the base salary of an Employee.

Then there is 'Clerk' which has 1/2 of base salary.

And then there are 'Manager' who have twice the base salary of that of an employee.

Help him write a program in Java as follows.

Create a class 'Employee' which has a method named 'getSalary' which returns a base salary of Rs. 10,000. It also has methods named 'getBonus' which returns the bonus amount for that role(initially set to Rs. 0).

Make 3 subclasses for different roles which inherit the 'Employee' class and each has functions named 'getSalary' and 'getBonus'.(You can assume values for 'getBonus' method)

Display the output for all cases. Also print the total salary received by each Employee after getting the bonus.

Note : Solve using method overriding

PROGRAM:

```
import java.util.*;  
  
class employee {  
    int salary = 10000;  
  
    int getsalary() {  
        return salary;  
    }  
}
```

```

class clerk extends employee {
    int getsalary() {
        return 10000 / 2;
    }

    int getbonus() {
        return 1000;
    }
}

class manager extends employee {
    int getsalary() {
        return 10000 * 2;
    }

    int getbonus() {
        return 8000;
    }
}

class intern extends employee {
    int getsalary() {
        return 10000 * 3 / 4;
    }

    int getbonus() {
        return 3000;
    }
}

public class salary {
    public static void main(String[] args) {
        employee obj = new employee();
        clerk ob = new clerk();
        intern o = new intern();
        manager i = new manager();
        System.out.println("The salary and bonus of intern is " + o.getsalary() + "
and " + o.getbonus());
        System.out.println("The salary and bonus of manager is " + i.getsalary() + "
and " + i.getbonus());
        System.out.println("The salary and bonus of clerk is " + ob.getsalary() + "
and " + ob.getbonus());
    }
}

```

	}
RESULT: <pre> The salary and bonus of intern is 7500 and 3000 The salary and bonus of manager is 20000 and 8000 The salary and bonus of clerk is 5000 and 1000 </pre>	
Program 3	
PROBLEM STATEMENT:	<p>Create a class named 'Shape' which has a method 'getArea', 'getPerimeter' and 'getSide' and all of them return 0. Make three subclasses for three different shapes - 'Circle', 'Triangle' and 'Pentagon'. These subclasses inherit the 'Shape' class and they also have 'getArea', 'getPerimeter' and 'getSide' methods.</p> <p>Write a program for the above scenario and display the solution.</p> <p>Note: Solve using method overriding</p>
PROGRAM:	<pre> import java.util.*; import java.lang.Math; class Shape { Scanner scan = new Scanner(System.in); double area, side, perimeter, radius, s1, s2, s3, a; void getArea() { area = 0; } void getPerimeter() { perimeter = 0; } void getSide() { side = 0; } } class Circle extends Shape { void getArea() { </pre>


```

        area = Math.PI * Math.pow(radius, 2);

    }

    void getPerimeter() {
        perimeter = Math.PI * 2 * radius;

    }

    void getSide() {
        System.out.print("Enter the radius of the circle: ");
        radius = scan.nextDouble();
    }
}

class Triangle extends Shape {
    void getArea() {
        double s = (s1 + s2 + s3) / 2;
        area = s * (s - s1) * (s - s2) * (s - s3);
        area = Math.sqrt(area);

    }

    void getPerimeter() {
        double s = (s1 + s2 + s3) / 2;
        perimeter = (s * 2);

    }

    void getSide() {
        System.out.print("Enter the 3-sides of the triangle: ");
        s1 = scan.nextDouble();
        s2 = scan.nextDouble();
        s3 = scan.nextDouble();
    }
}

class Pentagon extends Shape {
    void getArea() {
        area = ((Math.sqrt(5 * (5 + 2 * Math.sqrt(5)))) / 4) * Math.pow(a, 2);

    }
}

```

```

void getPerimeter() {
    perimeter = (5 * a);
}

void getSide() {
    System.out.print("Enter the side of the pentagon: ");
    a = scan.nextDouble();
}
}

public class area {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        Circle ob_c = new Circle();
        Triangle ob_t = new Triangle();
        Pentagon ob_p = new Pentagon();
        int option, flag;
        while (true) {
            System.out.println("\n 1) Circle \n 2) Triangle \n 3) Pentagon ");
            System.out.print("Enter the Shape you want: ");
            option = scan.nextInt();
            switch (option) {
                case 1:
                    ob_c.getSide();
                    ob_c.getArea();
                    ob_c.getPerimeter();
                    System.out.printf("Area= %.2f\nPerimeter=%.2f\n", ob_c.area,
ob_c.perimeter);

                    break;

                case 2:
                    ob_t.getSide();
                    ob_t.getArea();
                    ob_t.getPerimeter();
                    System.out.printf("Area= %.2f\nPerimeter=%.2f\n", ob_t.area,
ob_t.perimeter);
                    break;

                case 3:
                    ob_p.getSide();
                    ob_p.getArea();
                    ob_p.getPerimeter();

```

```
        System.out.printf("Area= %.2f\nPerimeter=%.2f\n", ob_p.area,
ob_p.perimeter);
        break;

        default:
            System.out.println("Invalid choice ");
            break;

    }
    System.out.println("");
    System.out.println("Do you want to continue?(yes=1/0=no)");
    flag = scan.nextInt();
    if (flag == 0) {
        break;
    }
}
}
```

RESULT:

```

1) Circle
2) Triangle
3) Pentagon
Enter the Shape you want: 1
Enter the radius of the circle: 4
Area= 50.27
Perimeter=25.13

Do you want to continue?(yes=1/0=no)
1

1) Circle
2) Triangle
3) Pentagon
Enter the Shape you want: 2
Enter the 3-sides of the triangle: 2 3 4
Area= 2.90
Perimeter=9.00

Do you want to continue?(yes=1/0=no)
1

1) Circle
2) Triangle
3) Pentagon
Enter the Shape you want: 3
Enter the side of the pentagon: 5
Area= 43.01
Perimeter=25.00

Do you want to continue?(yes=1/0=no)
0

```

Program 4

PROBLEM STATEMENT:

Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float). Each of these three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display its data.

Add a class sales that holds an array of three floats so that it can record the dollar sales of a particular publication for the last three months. Include a getdata() function to get three sales amounts from the user, and a putdata() function to display the sales figures. Alter the book class so they are derived from both publication and sales. An object of class book should input and output sales data along with its other data. Write a main() function to create a book object and a tape object and exercise their input/output capabilities.

PROGRAM:

```
import java.util.*;
```

```

class sales {
    Scanner sc = new Scanner(System.in);
    float[] sales = new float[3];

    void getdata() {
        System.out.println("Enter the dollar sales of the last three months.");
        for (int i = 0; i < 3; i++)
            sales[i] = sc.nextFloat();
    }

    void putdata() {
        System.out.println(sales[0] + "\t" + sales[1] + "\t" + sales[2] + "\t");
    }
}

class publication extends sales {
    int price;
    String name;

    void getdata() {
        price = sc.nextInt();
        name = sc.nextLine();
    }

    void putdata() {
        System.out.println(name + " " + price);
    }
}

class book extends publication {
    int page;

    void getdata() {
        System.out.println("Enter name of book");
        name = sc.nextLine();
        System.out.println("Enter price of " + name);
        price = sc.nextInt();
        System.out.println("Enter no. of pages the book contains");
        page = sc.nextInt();
        System.out.println("Enter the dollar sales of the last three months.");
        for (int i = 0; i < 3; i++)
            sales[i] = sc.nextFloat();
    }
}

```

```

    }

    void putdata() {
        System.out.println(
            name + "\t" + price + "\t" + page + "\t" + sales[0] + "\t" + sales[1] + "\t" +
sales[2] + "\t");
    }
}

class tape extends publication {
    int time;

    void getdata() {
        System.out.println("Enter name of tape");
        name = sc.nextLine();
        System.out.println("Enter price of " + name);
        price = sc.nextInt();
        System.out.println("Enter playing time of " + name + " in minutes");
        time = sc.nextInt();
        System.out.println("Enter the dollar sales of the last three months.");
        for (int i = 0; i < 3; i++)
            sales[i] = sc.nextFloat();
    }

    void putdata() {
        System.out.println(
            name + "\t" + price + "\t" + time + "\t" + sales[0] + "\t" + sales[1] + "\t" +
sales[2] + "\t");
    }
}

public class company {
    public static void main(String[] args) {
        book[] b = new book[3];
        tape[] t = new tape[3];
        for (int i = 0; i < 3; i++) {
            b[i] = new book();
            t[i] = new tape();
            System.out.println("ENTER DATA FOR BOOK");
            b[i].getdata();
            System.out.println("ENTER DATA FOR TAPE");
            t[i].getdata();
        }
    }
}

```

```

        System.out.println("Name\tPrice\tPage\tSales(3 MONTHS)\t");
        for (int i = 0; i < 3; i++) {
            b[i].putdata();
        }
        System.out.println("\n");
        System.out.println("Name\tPrice\tPlaying time\tSales(3 MONTHS)\t");
        for (int i = 0; i < 3; i++) {
            t[i].putdata();
        }
    }
}

```

RESULT:

```

ENTER DATA FOR BOOK
Enter name of book
A
Enter price of A
12
Enter no. of pages the book contains
23
Enter the dollar sales of the last three months.
123 123 333
ENTER DATA FOR TAPE
Enter name of tape
A
Enter price of A
14
Enter playing time of A in minutes
19
Enter the dollar sales of the last three months.
233 234 345
ENTER DATA FOR BOOK
Enter name of book
Enter name of tape
B
Enter price of B
23
Enter playing time of B in minutes
30
Enter the dollar sales of the last three months.
233 332 453

```

```

ENTER DATA FOR BOOK
Enter name of book
C
Enter price of C
10
Enter no. of pages the book contains
12
Enter the dollar sales of the last three months.
123 234 232
ENTER DATA FOR TAPE
Enter name of tape
C
Enter price of C
12
Enter playing time of C in minutes
17
Enter the dollar sales of the last three months.
123 324 453
Name    Price    Page    Sales(3 MONTHS)
A       12       23      123.0   123.0   333.0
B       24       29      134.0   345.0   245.0
C       10       12      123.0   234.0   232.0

Name    Price    Playing time    Sales(3 MONTHS)
A       14       19      233.0   234.0   345.0
B       23       30      233.0   332.0   453.0
C       12       17      123.0   324.0   453.0

```

Program 5

PROBLEM STATEMENT:

Give the definition of four classes, Person, Doctor, Patient and Billing, whose objects are records for a clinic.

Class Doctor will be derived from the class Person. A doctor have name and Date (inherited from the class Person), it's speciality; fees and income;

Patient will be derived from the class Person. A Patient record has the patient's name and Date (inherited from the class Person) and a Doctor object.

A Billing object will contain a Patient object, a Doctor object, Date of bill using date object and an amount due of type double. Be sure your classes have a reasonable complement of constructors, override equals(check equality of object) and toString methods and member functions.

First write a driver program to test all your member functions, and then write a test program that creates at least two patients, at least two doctors, and at least two Billing records, then prints out the total income from the Billing records.

	<p>At the time of Billing the Patient name and doctor name equality should be checked.</p> <p>Total bill generated will be no of days the patient admitted (admitted date + current date) * 2000+ doctore fees * no of days from the date of billing .</p> <p>Update the Doctors income also simultaneously</p>
PROGRAM:	<pre> import java.util.*; class Person { Calendar current = Calendar.getInstance(); Calendar bill = Calendar.getInstance(); Calendar admit = Calendar.getInstance(); Scanner sc = new Scanner(System.in); String name; int year, month, date; void getName() { name = sc.nextLine(); } String setName() { return name; } void getDate() { date = sc.nextInt(); month = sc.nextInt(); year = sc.nextInt(); } int setDate() { bill.set(year, month - 1, date); return 0; } } class Doctor extends Person { int salary, fee; void getName() { System.out.println("Enter the doctor's name:"); </pre>

```

        name = sc.nextLine();
    }

    String setName() {
        return name;
    }

    void getSal() {
        System.out.println("Enter the doctor's salary:");
        salary = sc.nextInt();
    }

    int setSal() {
        return salary;
    }

    void getFee() {
        System.out.println("Enter the doctor's fee:");
        fee = sc.nextInt();
    }

    int setFee() {
        return fee;
    }

    void getDate() {
        System.out.println("Enter the billing date (date, month, year):");
        date = sc.nextInt();
        month = sc.nextInt();
        year = sc.nextInt();
    }

    int setDate() {
        bill.set(year, month - 1, date);
        long d3 = bill.getTimeInMillis() / (1000 * 3600 * 24);
        return (int) d3;
    }
}

class Patient extends Person {
    void getName() {
        System.out.println("Enter the patient's name:");
        name = sc.nextLine();
    }
}

```

```

    }

    String setName() {
        return name;
    }

    void getDate() {
        System.out.println("Enter the date when the patient was admitted:");
        date = sc.nextInt();
        month = sc.nextInt();
        year = sc.nextInt();
    }

    int setDate() {
        admit.set(year, month - 1, date);
        long d2 = admit.getTimeInMillis() / (1000 * 3600 * 24);
        return (int) d2;
    }
}

class Billing extends Doctor {
    void getDate() {
        System.out.println("Enter the current date (date, month, year): ");
        date = sc.nextInt();
        month = sc.nextInt();
        year = sc.nextInt();
    }

    int setDate() {
        current.set(year, month - 1, date);
        long d1 = current.getTimeInMillis() / (1000 * 3600 * 24);
        return (int) d1;
    }

    int Total_bill(int d1, int d2, int d3, int fee) {
        int n1 = Math.abs((int) (d1 - d2));
        int n2 = Math.abs((int) (d1 - d3));
        int tot = ((n1 * 2000) + (fee * n2));
        return tot;
    }
}

public class clinic {

```

```

public static void main(String[] args) {
    Doctor[] d = new Doctor[2];
    Patient[] p = new Patient[2];
    Billing[] b = new Billing[2];
    for (int i = 0; i < 2; i++) {
        d[i] = new Doctor();
        p[i] = new Patient();
        b[i] = new Billing();
        b[i].getDate();
        d[i].getName();
        d[i].getSal();
        d[i].getFee();
        d[i].getDate();
        p[i].getName();
        p[i].getDate();
    }
    for (int i = 0; i < 2; i++) {
        System.out.println("Doctor : " + d[i].setName());
        System.out.println("Patient: " + p[i].setName());
        System.out.println(
            "Fees : Rs. " + b[i].Total_bill(b[i].setDate(), p[i].setDate(),
d[i].setDate(), d[i].setFee()));
        System.out.println("Doctor's new salary: Rs. " + (d[i].setSal() +
            b[i].Total_bill(b[i].setDate(), p[i].setDate(), d[i].setDate(),
d[i].setFee())));
    }
}
}

```

RESULT:

```

Enter the current date (date, month, year):
27 6 2022
Enter the doctor's name:
Dr. Rick
Enter the doctor's salary:
30000
Enter the doctor's fee:
1000
Enter the billing date (date, month, year):
30 6 2022
Enter the patient's name:
Morty
Enter the date when the patient was admitted:
28 6 2022
Enter the current date (date, month, year):
27 6 2022
Enter the doctor's name:
Dr. Ross Geller
Enter the doctor's salary:
50000
Enter the doctor's fee:
1000
Enter the billing date (date, month, year):
1 7 2022
Enter the patient's name:
Rachel Green
Enter the date when the patient was admitted:
29 6 2022

```

```

Doctor : Dr. Rick
Patient: Morty
Fees : Rs. 5000
Doctor's new salary: Rs. 35000
Doctor : Dr. Ross Geller
Patient: Rachel Green
Fees : Rs. 8000
Doctor's new salary: Rs. 58000

```

CONCLUSION:	We learned about method overriding and have implemented in the above programs.
--------------------	--