Name	**Hatim Yusuf Sawai**
UID no.	**2021300108**
Experiment No.	3

AIM:	Programs on Polymorphism: Implement a Program to demonstrate method overloading and constructor overloading.
Program 1	
PROBLEM STATEMENT:	Create a Date class with data int year, int month, int date, int hrs, int min, int sec. Create a default, no-argument constructor which sets the default date to January 1, 2000, 00:00:00 Create 3 overloaded setter methods void setDate(int year, int month, int date) void setDate(int year, int month, int date, int hrs, int min) void setDate(int year, int month, int date, int hrs, int min, int sec) Also write a displayDate() method which will display the date depending on the type of date object created

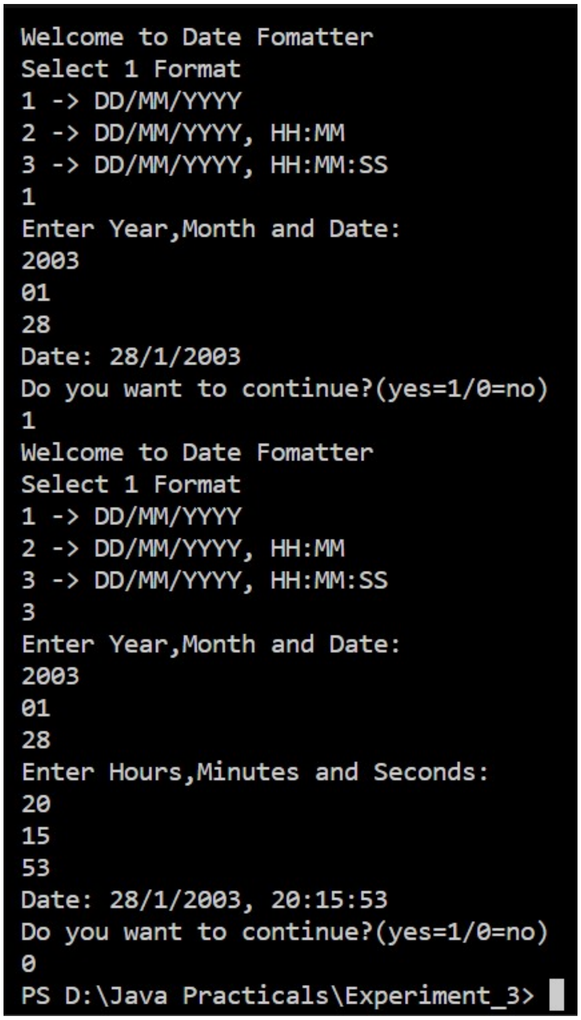
PROGRAM:

```
import java.util.*;
class Date {
    int year,month,date;
    int hrs,min,sec;
    Date() {
        year = 2000;
        month = 01;
        date = 01;
        hrs = 00;
        min = 00;
        sec = 00;
    }
    //setters
    void setDate(int y,int m,int d) {year = y;month = m;date = d;};
    void setDate(int y,int m,int d,int h,int mi) {year = y;month = m;date =
d;hrs = h;min = mi;};
    void setDate(int y,int m,int d,int h,int mi,int s) {year = y;month =
m;date = d;hrs = h;min = mi;sec = s;};
    //Utility methods
    void displayDate(int n) {
        if(n==1) {
            System.out.printf("Date: %d/%d/%d\n",date,month,year);
        }
        else if(n==2) {
            System.out.printf("Date: %d/%d/%d,
%d:%d\n",date,month,year,hrs,min);
        }
        else if(n==3) {
            System.out.printf("Date: %d/%d/%d,
%d:%d:%d\n",date,month,year,hrs,min,sec);
        }
    }
    public static void main( String[] args){
        Scanner sc= new Scanner(System.in);
        Date d1= new Date();
        Date d2= new Date();
        Date d3= new Date();
        int choice,flag;
        int year,month,date,hrs,min,sec;
        while(true) {
            System.out.println("Welcome to Date Fomatter");
            System.out.println("Select 1 Format\n1 -> DD/MM/YYYY\n2 ->
DD/MM/YYYY, HH:MM\n3 -> DD/MM/YYYY, HH:MM:SS");
            choice = sc.nextInt();
            System.out.println("Enter Year,Month and Date: ");
            year = sc.nextInt();
            month = sc.nextInt();
            date = sc.nextInt();
            if(month>12) {
                System.out.println("Invalid Month(1-12)");
                break;
```

```
}
else if(date>31) {
    System.out.println("Invalid Date(1-31)");
    break;
}
switch(choice) {
    case 1:
        d1.setDate(year, month, date);
        d1.displayDate(choice);
        break;
    case 2:
        System.out.println("Enter Hours and Minutes: ");
        hrs = sc.nextInt();
        min = sc.nextInt();
        if(hrs>23) {
            System.out.println("Invalid Hours(0-23)");
            break;
        }
        else if(min>59) {
            System.out.println("Invalid Minutes(0-59)");
            break;
        }
        d2.setDate(year, month, date, hrs, min);
        d2.displayDate(choice);
        break;
    case 3:
        System.out.println("Enter Hours,Minutes and Seconds: ");
        hrs = sc.nextInt();
        min = sc.nextInt();
        sec = sc.nextInt();
        if(hrs>23) {
            System.out.println("Invalid Hours(0-23)");
            break;
        }
        else if(min>59) {
            System.out.println("Invalid Minutes(0-59)");
```

```
                        break;
                    }
                    else if(sec>59) {
                        System.out.println("Invalid Seconds(0-59)");
                        break;
                    }
                    d3.setDate(year, month, date, hrs, min, sec);
                    d3.displayDate(choice);
                    break;
                default:
                    System.out.println("Invalid Choice!");
                    break;
            }
            System.out.println("Do you want to continue?(yes=1/0=no)");
            flag = sc.nextInt();
            if(flag==0) {
                break;
            }
        }
    }
}
```

RESULT:



```
Welcome to Date Fomatter
Select 1 Format
1 -> DD/MM/YYYY
2 -> DD/MM/YYYY, HH:MM
3 -> DD/MM/YYYY, HH:MM:SS
1
Enter Year,Month and Date:
2003
01
28
Date: 28/1/2003
Do you want to continue?(yes=1/0=no)
1
Welcome to Date Fomatter
Select 1 Format
1 -> DD/MM/YYYY
2 -> DD/MM/YYYY, HH:MM
3 -> DD/MM/YYYY, HH:MM:SS
3
Enter Year,Month and Date:
2003
01
28
Enter Hours,Minutes and Seconds:
20
15
53
Date: 28/1/2003, 20:15:53
Do you want to continue?(yes=1/0=no)
0
PS D:\Java Practicals\Experiment_3>
```

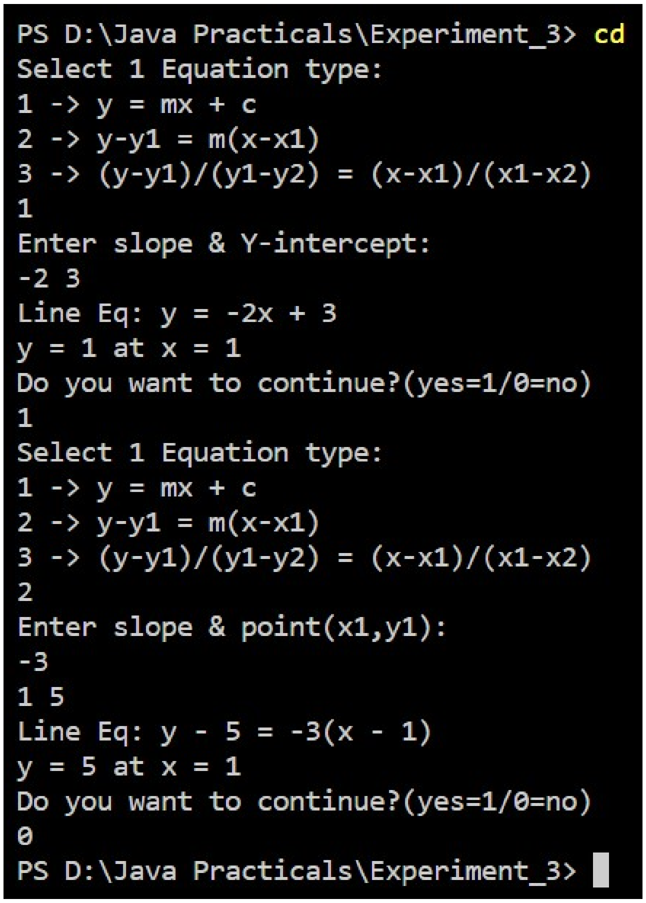
Program 2

PROBLEM STATEMENT:	Given a class Line with slope, y-intercept, x1, y1, x2, y2 as attributes, write 3 constructors for equations for the line given Slope-y-intercept, Slope Point and two Point forms Slope-y-intercept: $y = mx + c$ Slope point form: $y - y1 = m(x - x1)$ Two Point form: $(y - y1) / (y1 - y2) = (x - x1) / (x1 - x2)$ Each constructor should display the appropriate Line equation and appropriate value of y for given x.

PROGRAM:

```
import java.util.*;
class Line {
    double m,c;
    double x=1,x1,y1,x2,y2;
    //constructor 1
    Line(double m,double c) {
        this.m = m;
        this.c = c;
        System.out.printf("Line Eq: y = %.0f x + %.0f",m,c);
        System.out.printf("\ny = %.0f at x = %.0f",m*x + c,x);
    }
    //constructor 2
    Line(double m,double x1,double y1) {
        this.m = m;
        this.x1 = x1;
        this.y1 = y1;
        System.out.printf("Line Eq: y - %.0f = %.0f (x - %.0f )",y1,m,x1);
        System.out.printf("\ny = %.0f at x = %.0f",(m*(x-x1))+y1);
    }
    Line(double x1,double y1,double x2,double y2) {
        this.x1 = x1;
        this.y1 = y1;
        this.x2 = x2;
        this.y2 = y2;
        System.out.printf("Line Eq: (y - %.0f )/%.0f = (x -
%.0f)/%.0f",y1,y1-y2,x2,x1-x2);
        System.out.printf("\ny = %.0f at x = %.0f",((x-x1)/(x1-x2))*(y1-
y2)+y1,x);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice,flag;
        double m,c;
        double x1,x2,y1,y2;
        while(true) {
            System.out.println("Select 1 Equation type:\n1 -> y = mx + c\n2
-> y-y1 = m(x-x1)\n3 -> (y-y1)/(y1-y2) = (x-x1)/(x1-x2)");
            choice = sc.nextInt();
            switch(choice) {
                case 1:
                    System.out.println("Enter slope & Y-intercept: ");
                    m = sc.nextDouble();
                    c = sc.nextDouble();
                    Line l1 = new Line(m, c);
                    break;
                case 2:
                    System.out.println("Enter slope & point(x1,y1): ");
                    m = sc.nextDouble();
                    x1 = sc.nextDouble();
                    y1 = sc.nextDouble();
                    Line l2 = new Line(m, x1, y1);
                    break;
                case 3:
                    System.out.println("Enter point(x1,y1) & point(x2,y2): ");
                    x1 = sc.nextDouble();
                    y1 = sc.nextDouble();
                    x2 = sc.nextDouble();
                    y2 = sc.nextDouble();
                    Line l3 = new Line(x1, y1, x2, y2);
                    break;
                default:
                    System.out.println("Invalid choice");
                    break;
            }
            System.out.println("\nDo you want to continue?(yes=1/0=no)");
            flag = sc.nextInt();
            if(flag==0) {
                break;
            }
        }
    }
}
```

RESULT:



```
PS D:\Java Practicals\Experiment_3> cd
Select 1 Equation type:
1 -> y = mx + c
2 -> y-y1 = m(x-x1)
3 -> (y-y1)/(y1-y2) = (x-x1)/(x1-x2)
1
Enter slope & Y-intercept:
-2 3
Line Eq: y = -2x + 3
y = 1 at x = 1
Do you want to continue?(yes=1/0=no)
1
Select 1 Equation type:
1 -> y = mx + c
2 -> y-y1 = m(x-x1)
3 -> (y-y1)/(y1-y2) = (x-x1)/(x1-x2)
2
Enter slope & point(x1,y1):
-3
1 5
Line Eq: y - 5 = -3(x - 1)
y = 5 at x = 1
Do you want to continue?(yes=1/0=no)
0
PS D:\Java Practicals\Experiment_3>
```

Program 3

PROBLEM STATEMENT:

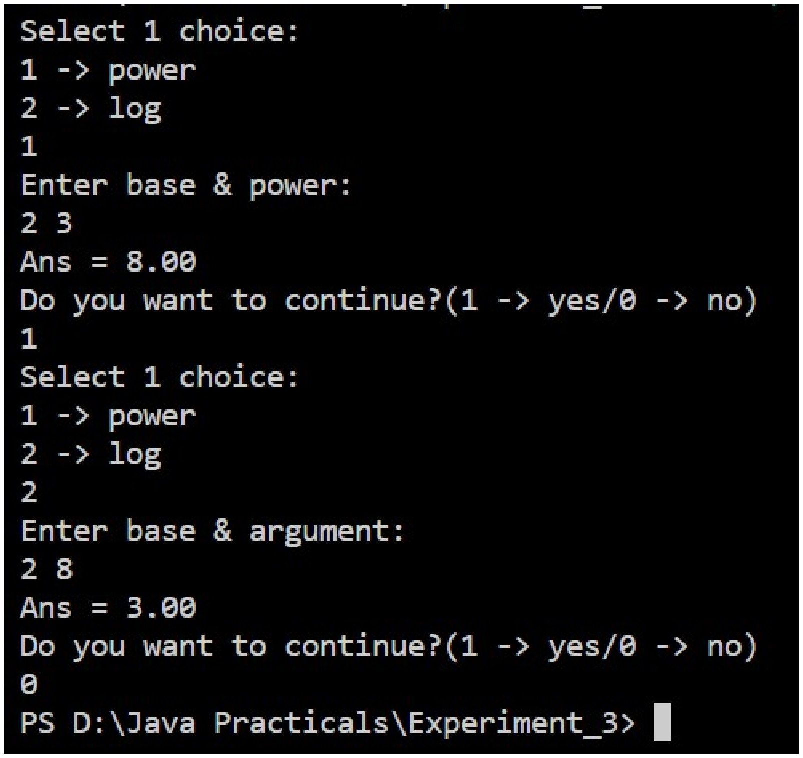
Create a Test class with data double base, int power, int logBase, int argument.
Create a no-argument constructor which sets the default value of all variables to 2.
There are 2 overloaded functions:
1. double calculate (double base, int power)
This function returns the value when *base* is raised to *power*
For example: calculate (3.0, 2) returns the value of 3.0 raised to 2 i.e., 9.0
2. double calculate (int logBase, int argument)
This function returns the value of the log of *argument* to the base *logBase*.
For example: calculate (3, 9) returns log of 9 to the base 3 i.e., 2.0

Create a main method in a separate class to call the above functions with the following inputs:
1. calculate (2, 4)
2. calculate (2.0, 4.0)
Create a display() method which displays the output based on the type of Test object created.

PROGRAM:

```
import java.util.*;
import java.lang.Math;
class Test {
    double base,res;
    int power,logBase,argument;
    Test() {
        base = 2;
        power = 2;
        logBase = 2;
        argument = 2;
    }
    double calculate(double base, int power) {
        res = Math.pow(base,power);
        return res;
    }
    double calculate(int logBase,int argument) {
        res = Math.log(argument)/Math.log(logBase);
        return res;
    }
    void display() {
        System.out.printf("Ans = %.2f",res);
    }
}
class Expo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice,flag;
        while(true) {
            System.out.println("Select 1 choice:\n1 -> power\n2 -> log");
            choice = sc.nextInt();
            switch(choice) {
                case 1:
                    System.out.println("Enter base & power: ");
                    Test t1 = new Test();
                    t1.base = sc.nextDouble();
                    t1.power = sc.nextInt();
                    t1.calculate(t1.base,t1.power);
                    t1.display();
                    break;
                case 2:
                    System.out.println("Enter base & argument: ");
                    Test t2 = new Test();
                    t2.logBase = sc.nextInt();
                    t2.argument = sc.nextInt();
                    t2.calculate(t2.logBase,t2.argument);
                    t2.display();
                    break;
                default:
                    System.out.println("Invalid choice");
                    break;
            }
            System.out.println("\nDo you want to continue?(1 -> yes/0 ->
no)");
            flag = sc.nextInt();
            if(flag == 0)
                break;
        }
    }
}
```

RESULT:



```
Select 1 choice:
1 -> power
2 -> log
1
Enter base & power:
2 3
Ans = 8.00
Do you want to continue?(1 -> yes/0 -> no)
1
Select 1 choice:
1 -> power
2 -> log
2
Enter base & argument:
2 8
Ans = 3.00
Do you want to continue?(1 -> yes/0 -> no)
0
PS D:\Java Practicals\Experiment_3>
```

Program 4

PROBLEM STATEMENT:

Write a menu-driven program to recruit an employee (depending on his performance in various rounds) in some software company using constructor overloading.

Selection Criteria for each post is given below:

i) Programmer (Minimum total of 80 marks):-

Rounds:-

(1) Course Work

(2) Aptitude Test

(3) Technical Test

(4) Interview

ii) Team Leader (Minimum total of 85 marks):-

Rounds:-

(1) Technical Test

(2) Interview

iii) Project Manager (Minimum score 90 marks)

Rounds:-

(1) Interview

Create a class Posting and write 3 constructors to initialize the object and set the parameters
and display the employee post according to selection criteria.
Data members:

- int courseWork;
- int AptTest;
- int TechTest;
- int interview;

Methods:

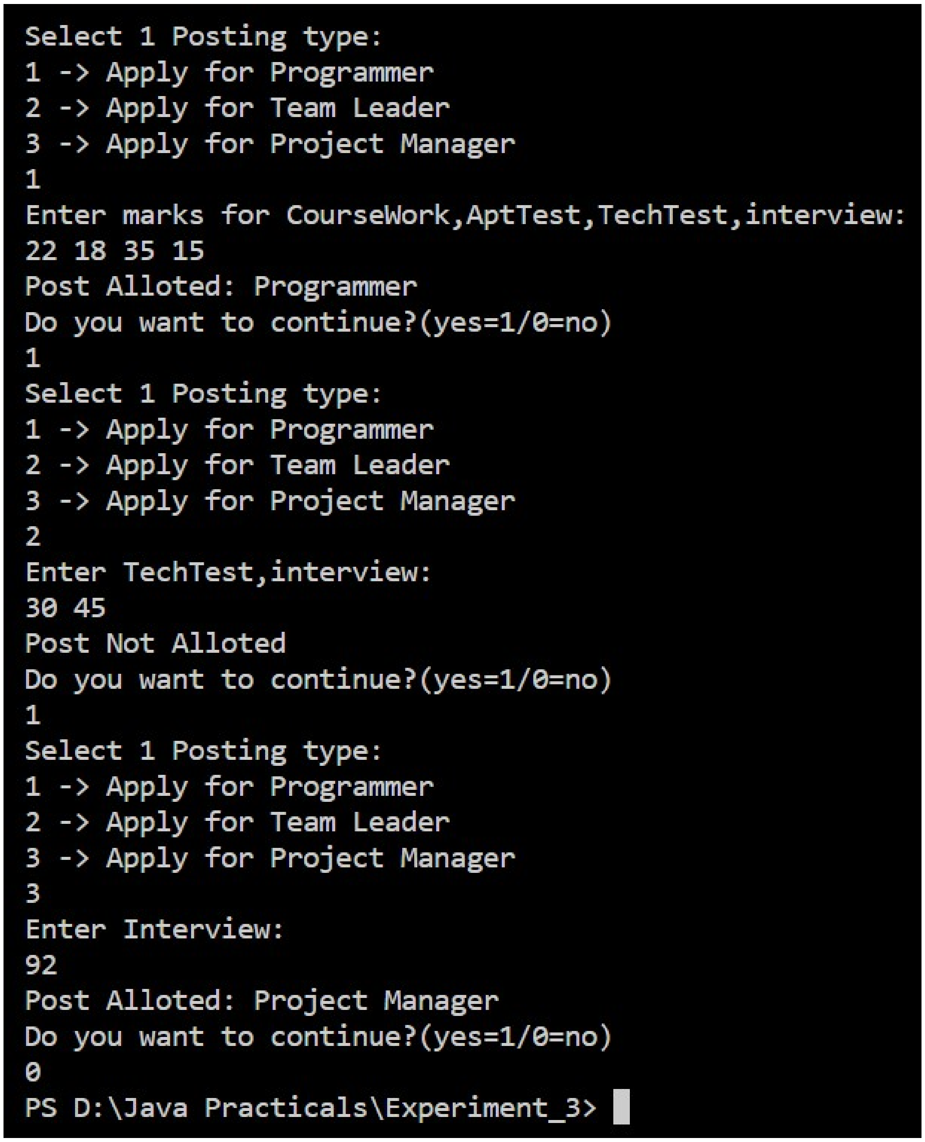
- Posting (int courseWork, int AptTest, int TechTest,int interview)
- Posting (int TechTest,int interview)
- Posting (int interview)

Make use of ‘this’ keyword.

PROGRAM:

```
import java.util.*;
class Posting {
    int courseWork;
    int AptTest;
    int TechTest;
    int interview;
    Posting(int courseWork,int AptTest,int TechTest,int interview) {
        this.courseWork = courseWork;
        this.AptTest = AptTest;
        this.TechTest = TechTest;
        this.interview = interview;
    }
    Posting(int TechTest,int interview) {
        this.TechTest = TechTest;
        this.interview = interview;
    }
    Posting(int Interview) {
        this.interview = Interview;
    }
    void allotpost(int choice) {
        if(interview>=90 && choice==3) {
            System.out.println("Post Alloted: Project Manager");
        }
        else if(TechTest+interview>=85 && choice==2) {
            System.out.println("Post Alloted: Team Leader");
        }
        else if(courseWork+AptTest+TechTest+interview>=80 &&
choice==1) {
            System.out.println("Post Alloted: Programmer");
        }
        else {
            System.out.println("Post Not Alloted");
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice,flag;
        int courseWork,AptTest,TechTest,interview;
        while(true) {
            System.out.println("Select 1 Posting type:\n1 -> Apply for
Programmer\n2 -> Apply for Team Leader\n3 -> Apply for Project
Manager");
            choice = sc.nextInt();
            switch(choice) {
                case 1:
                    System.out.println("Enter marks for
CourseWork,AptTest,TechTest,interview: ");
                    courseWork = sc.nextInt();
                    AptTest = sc.nextInt();
                    TechTest = sc.nextInt();
                    interview = sc.nextInt();
                    Posting p1 = new
Posting(courseWork,AptTest,TechTest,interview);
                    p1.allotpost(choice);
                    break;
                case 2:
                    System.out.println("Enter TechTest,interview: ");
                    TechTest = sc.nextInt();
                    interview = sc.nextInt();
                    Posting p2 = new Posting(TechTest,interview);
                    p2.allotpost(choice);
                    break;
                case 3:
                    System.out.println("Enter Interview: ");
                    interview = sc.nextInt();
                    Posting p3 = new Posting(interview);
                    p3.allotpost(choice);
                    break;
                default:
                    System.out.println("Invalid choice");
                    break;
            }
            System.out.println("Do you want to continue?(yes=1/0=no)");
            flag = sc.nextInt();
            if(flag==0) {
                break;
            }
        }
    }
}
```

RESULT:



```
Select 1 Posting type:
1 -> Apply for Programmer
2 -> Apply for Team Leader
3 -> Apply for Project Manager
1
Enter marks for CourseWork,AptTest,TechTest,interview:
22 18 35 15
Post Alloted: Programmer
Do you want to continue?(yes=1/0=no)
1
Select 1 Posting type:
1 -> Apply for Programmer
2 -> Apply for Team Leader
3 -> Apply for Project Manager
2
Enter TechTest,interview:
30 45
Post Not Alloted
Do you want to continue?(yes=1/0=no)
1
Select 1 Posting type:
1 -> Apply for Programmer
2 -> Apply for Team Leader
3 -> Apply for Project Manager
3
Enter Interview:
92
Post Alloted: Project Manager
Do you want to continue?(yes=1/0=no)
0
PS D:\Java Practicals\Experiment_3>
```

CONCLUSION:	In this experiment we learned how to use polymorphism in java and overload methods and constructor and build menu-driven programs.