

Practical No.	Name of the practical
1.	Write a Program that displays Welcome to Java, Learning Java Now and Programming is fun.
2.	Write a program that solves the following equation and displays the value x and y: 1) $3.4x + 50.2y = 44.5$ 2) $2.1x + .55y = 5.9$ (Assume Cramer's rule to solve equation $ax + by = e$ $x = (ed - bf) / (ad - bc)$ $cx + dy = f$ $y = (af - ec) / (ad - bc)$)
3.	Write a program that reads a number in meters, converts it to feet, and displays the result.
4.	Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI. Note:- 1 pound=.45359237 Kg and 1 inch=.0254 meters.
5.	Write a program that prompts the user to enter three integers and display the integers in decreasing order.
6.	Write a program that prompts the user to enter a letter and check whether a letter is a vowel or constant.
7.	Assume a vehicle plate number consists of three uppercase letters followed by four digits. Write a program to generate a plate number.
8.	Write a program that reads an integer and displays all its smallest factors in increasing order. For example if input number is 120, the output should be as follows:2,2,2,3,5.
9.	Write a method with following method header. <code>public static int gcd(int num1, int num2)</code> Write a program that prompts the user to enter two integers and compute the gcd of two integers.
10.	Write a test program that prompts the user to enter ten numbers, invoke a method to reverse the numbers, display the numbers.
11.	Write a program that generate 6*6 two-dimensional matrix, filled with 0's and 1's , display the matrix, check every raw and column have an odd number's of 1's.
12.	Write a program that creates a Random object with seed 1000 and displays the first 100 random integers between 1 and 49 using the NextInt (49) method.

13.	Write a program for calculator to accept an expression as a string in which the operands and operator are separated by zero or more spaces. For ex: 3+4 and 3 + 4 are acceptable expressions.
14.	Write a program that creates an Array List and adds a Loan object , a Date object , a string, and a Circle object to the list, and use a loop to display all elements in the list by invoking the object's to String() method.
15.	Write the bin2Dec (string binary String) method to convert a binary string into a decimal number. Implement the bin2Dec method to throw a Number Format Exception if the string is not a binary string.
16.	Write a program that prompts the user to enter a decimal number and displays the number in a fraction. Hint: Read the decimal number as a string, extract the integer part and fractional part from the string.
17.	Write a program that displays a tic-tac-toe board. A cell may be X, O, or empty. What to display at each cell is randomly decided. The X and O are images in the files X.gif and O.gif.
18.	Write a program that moves a circle up, down, left or right using arrow keys.
19.	Write a program that displays the color of a circle as red when the mouse button is pressed and as blue when the mouse button is released.
20.	Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed.
21.	Write a program to create a file name 123.txt, if it does not exist. Append a new data to it if it already exist. write 150 integers created randomly into the file using Text I/O. Integers are separated by space.
22.	Write a recursive method that returns the smallest integer in an array. Write a test program that prompts the user to enter an integer and display its product.
23.	Write a generic method that returns the minimum elements in a two dimensional array.
24.	Define MYPriorityQueue class that extends Priority Queue to implement the Cloneable interface and implement the clone() method to clone a priority queue.
25.	Write a program that reads words from a text file and displays all the nonduplicate words in descending order.The text file is passed as a command-line argument.

Practical – 1

- Write a Program that displays Welcome to Java, Learning Java Now and Programming is fun.

INPUT :

```
public class p1
{
    public static void main(String[] args)
    {
        System.out.println(" Hello World ");
        System.out.println(" Welcome to Java, \n Learning Java Programming is
fun. ");
    }
}
```

OUTPUT :

```
Hello World
Welcome to Java,
Learning Java Programming is fun.
```

Practical - 2

- Write a program that solves the following equation and displays the value x and y:

1) $3.4x + 50.2y = 44.5$

2) $2.1x + .55y = 5.9$ (Assume Cramer's rule to solve equation

$ax + by = e$ $x = \frac{ed - bf}{ad - bc}$

$cx + dy = f$ $y = \frac{af - ec}{ad - bc}$)

INPUT :

```
public class p2
{
    public static void main(String[] args) {
        System.out.println("Given System of Equations is:");
        System.out.println(" 1) 3.4x+50.2y=44.5 \n 2) 2.1x+.55y=5.9");
        double a=3.4,b=50.2,c=2.1,d=.55,e=44.5,f=5.9;
        double x=((e*d)-(b*f))/((a*d)-(b*c));
        double y=((a*f)-(e*c))/((a*d)-(b*c));
        System.out.println("Solution for the given system of equation:");
        System.out.println("X : "+x);
        System.out.println("Y : "+y);
    }
}
```

OUTPUT :

```
Given System of Equations is:
 1) 3.4x+50.2y=44.5
 2) 2.1x+.55y=5.9
Solution for the given system of equation:
X : 2.623901496861419
Y : 0.7087397392563978
```

Practical – 3

- Write a program that reads a number in meters, converts it to feet, and displays the result.

INPUT :

```
import java.util.Scanner;
```

```
public class p3 {  
    public static void main(String[] args) {  
        System.out.print("Enter the value in Meters: ");  
        Scanner sc = new Scanner(System.in);  
        double num = sc.nextDouble();  
        double feet=num*3.28084;  
        System.out.println(num + " Meters =" + feet + " Feets");  
  
        sc.close();  
    }  
}
```

OUTPUT :

```
Enter the value in Meters: 50  
50.0 Meters =164.042 Feets
```

Practical – 4

- **Body Mass Index (BMI)** is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI.

Note:- 1 pound=.45359237 Kg and 1 inch=.0254 meters.

INPUT :

```
import java.util.Scanner;
```

```
public class p4 {  
    public static void main(String[] args) {  
        double weight,height,bmi;  
        Scanner input = new Scanner(System.in);  
        System.out.print("Enter your weight in pound: ");  
        weight= input.nextDouble()*0.45359273; // convert pound to kg  
        System.out.print("Enter your height in inches: ");  
        height = input.nextDouble()*0.0254; // convert inches to meters  
        bmi = weight / (height*height);  
        System.out.print("Your BMI is : "+bmi);  
  
        input.close();  
    }  
}
```

OUTPUT :

```
Enter your weight in pound: 110  
Enter your height in inches: 5.5  
Your BMI is : 2556.618682328274
```

Practical - 5

- **Write a program that prompts the user to enter three integers and display the integers in decreasing order.**

INPUT :

```
import java.util.Scanner;

public class p5
{
    public static void main(String[] args) {
        int i,j,arr[]={0,0,0};
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the three numbers:");
        for(i=0;i<3;i++)
            arr[i]=input.nextInt();
        input.close();
        for(i=0;i<3-1;i++)
            for(j=0;j<3-i-1;j++)
                if(arr[j]<arr[j+1])
                {
                    int temp = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = temp;
                }
        System.out.println("Numbers in the Decreasing order: ");
        for(i=0;i<3;i++)
            System.out.println(arr[i]);

    }
}
```

OUTPUT :

Enter the three numbers:

2 5 8

Numbers in the Decreasing order:

8

5

2

Practical – 6

- Write a program that prompts the user to enter a letter and check whether a letter is a vowel or constant.

INPUT :

```
import java.util.Scanner;
```

```
public class p6 {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        System.out.print("Please enter a character:");  
        char ch = input.next().charAt(0);  
        switch(ch) {  
            case 'a':  
            case 'e':  
            case 'i':  
            case 'o':  
            case 'u':  
            case 'A':  
            case 'E':  
            case 'I':  
            case 'O':  
            case 'U':  
                System.out.println(ch + " - It is a Vowel.");  
                break;  
            default:  
                System.out.println(ch + " - It is Consonant.");  
        }  
        input.close();  
    }  
}
```

OUTPUT :

```
Please enter a character:A  
A - It is a Vowel.
```

Practical – 7

- Assume a vehicle plate number consists of three uppercase letters followed by four digits. Write a program to generate a plate number.

INPUT :

```
public class p7 {  
    public static void main(String[] args) {  
        String numberPlate = ""+(char)(Math.random()*26+65);  
        numberPlate += (char)(Math.random()*26+65);  
        numberPlate += (char)(Math.random()*26+65);  
        String digits = (""+ Math.random()).substring(3, 7);  
        System.out.println("Generated number plate value is: "  
                           + numberPlate + " " +digits);  
  
        //          numberPlate += (int)(Math.random()*10);  
        //          numberPlate += (int)(Math.random()*10);  
        //          numberPlate += (int)(Math.random()*10);  
        //          numberPlate += (int)(Math.random()*10);  
  
        //          double num = Math.random();  
        //          System.out.println(num);  
        //          String str = ""+num;  
        //          System.out.println(str);  
        //          System.out.println(subStr);  
    }  
}
```

OUTPUT :

```
Generated number plate value is: XMP 0332
```

Practical – 8

- Write a program that reads an integer and displays all its smallest factors in increasing order. For example if input number is 120, the output should be as follows:2,2,2,3,5

INPUT :

```
import java.util.Scanner;
```

```
public class p8 {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        System.out.print("Enter the number :");  
        int a = input.nextInt();  
        System.out.println("Factors of "+ a + " are :");  
        for(int i=2;a>1;)   
        {  
            if(a%i==0)  
            {  
                System.out.print(i+",");  
                a/=i;  
            }  
            else  
            {  
                i++;  
            }  
        }  
  
        //          for(int i=1;i<=a;i++)  
        //          if(a%i==0)  
        //          System.out.print(i+",");  
        input.close();  
    }  
}
```

OUTPUT :

```
Enter the number :100
Factors of 100 are :
2,2,5,5,
```

Practical – 9

- Write a method with following method header.

public static int gcd(int num1, int num2)

Write a program that prompts the user to enter two integers and compute the gcd of two integers.

INPUT :

```
import java.util.Scanner;
```

```
public class p9 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("\n Enter the number 1 : ");
        int num1 = input.nextInt();
        System.out.print("\n Enter the number 2 : ");
        int num2 = input.nextInt();
        int result = gcd(num1,num2);
        System.out.print("\n GCD of " + num1 + " and " + num2 + " is : " + result);
        input.close();
    }

    public static int gcd(int num1, int num2) {
        int result=1;
        for(int i=2;i<=num1 && i<=num2;)
        {
            if(num1%i==0 && num2%i==0)
            {
                result*=i;
                num1/=i;
                num2/=i;
            }
            else if(num1%i==0)
                num1/=i;
            else if(num2%i==0)
                num2/=i;
            else
                continue;
        }
        return result;
    }
}
```

```

        i++;
    }
    return result;
}
}

```

OUTPUT :

```

Enter the number 1 : 10

Enter the number 2 : 20

GCD of 10 and 20 is : 10

```

Practical -10

- **Write a test program that prompts the user to enter ten numbers, invoke a method to reverse the numbers, display the numbers.**

INPUT :

```

import java.util.Scanner;

public class p10 {
    public static void main(String[] args) {
        int[] list = new int[10];
        Scanner input = new Scanner(System.in);
        System.out.println("\n Please enter 10 Numbers:");
        for (int i = 0; i < list.length; i++)
            list[i]=input.nextInt();

        System.out.println("\nOriginal Array:");
        for (int i = 0; i < list.length; i++)
            System.out.print(list[i] + " ");
        reverse(list);
        System.out.println("\nReveresed Array:");
        for (int i = 0; i < list.length; i++)
            System.out.print(list[i] + " ");
    }
    public static void reverse(int[] arr) {
        for (int i = 0; i < arr.length/2; i++){
            int temp = arr[i];

```

```

arr[i] = arr[arr.length-1-i];
arr[arr.length-1-i] = temp;
// list[i]=list[i]+list[list.length-1-i];
// list[list.length-1-i]=list[i]-list[list.length-1-i];
// list[i]=list[i]-list[list.length-1-i];
    }
}
}

```

OUTPUT :

```

Please enter 10 Numbers:
2
4
6
8
10
12
14
16
18
20

Original Array:
2 4 6 8 10 12 14 16 18 20
Reveresed Array:
20 18 16 14 12 10 8 6 4 2

```

Practical – 11

- Write a program that generate 6*6 two-dimensional matrix, filled with 0's and 1's , display the matrix, check every row and column have an odd number's of 1's.

INPUT :

```
public class p11 {
    public static void main(String[] args) {

        int[][] arr = new int[6][6];
        System.out.println("New matrix:");
        printArray(arr);

        for(int i=0;i<arr.length-1;i++)
        {
            int cnt=0;
            for(int j=0;j<arr[i].length-1;j++) {
                arr[i][j]=(int)(Math.random()*2);
                cnt+=arr[i][j];
            }
            if(cnt%2==0) {
                arr[i][arr[i].length-1]=1;
            }
        }

        for(int j=0;j<6;j++) {
            int cnt=0;
            for(int i=0;i<6;i++) {
                cnt+=arr[i][j];
            }
            if(cnt%2==0) {
                arr[arr.length-1][j]=1;
            }
        }
        System.out.println("Matrix with odd no of 1's in row n column:");
        printArray(arr);
    }

    public static void printArray(int[][] arr) {
```

```

        for(int i=0;i<arr.length;i++) {
            for(int j=0;j<arr[i].length;j++) {
                System.out.print(arr[i][j]+" ");
            }
            System.out.println();
        }
    }
}

```

OUTPUT :

```

New matrix:
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
Matrix with odd no of 1's in row n column:
0 1 0 1 1 0
1 1 1 1 0 1
0 1 0 0 0 0
0 1 1 0 0 1
1 1 0 0 0 1
1 0 1 1 0 0

```


Practical – 12

- Write a program that creates a Random object with seed 1000 and displays the first 100 random integers between 1 and 49 using the NextInt (49) method.

INPUT :

```
import java.util.Arrays;
```

```
import java.util.Random;
```

```
public class p12 {
```

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

```
        Random random = new Random(1000);
```

```
        for( int i = 1 ; i <= 100 ; i++ ) {
```

```
            System.out.printf("%5d",random.nextInt(999)+1);
```

```
            if(i%10==0)
```

```
                System.out.println();
```

```
        }
```

```
        //
```

```
        System.out.println(Arrays.toString((random.ints(100,1,49).toArray())));
```

```
    }
```

```
}
```

OUTPUT :

```
125 713 364 575 247 588 350 189 867 73
550 674 902 558 943 244 231 503 134 416
460 627 507 56 501 114 14 603 298 213
979 231 985 558 38 612 131 872 174 451
40 400 883 569 51 214 903 901 744 264
306 159 784 521 831 172 386 419 510 208
607 702 432 864 46 121 667 60 511 798
704 219 444 45 691 348 339 76 938 676
710 217 5 343 658 606 999 297 730 475
859 144 539 785 355 528 429 287 436 805
```

Practical – 13

- Write a program for calculator to accept an expression as a string in which the operands and operator are separated by zero or more spaces. For ex: 3+4 and 3 + 4 are acceptable expressions.

INPUT :

```
import java.util.Scanner;
public class p13 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Please enter valid expression: ");
        String exp = input.nextLine();

        System.out.println("Given expression: "+exp);

        double a=0;
        double b=0;
        double ans=0;
        char op = '1';

        //=====
        //      char[] operators = {'+', '-', '*', '/'};
        //      for(int i=0;i<operators.length;i++) {
        //          if(exp.indexOf(operators[i])!=-1) {
        //              op = operators[i];
        //          }
        //      }
        //      a = Double.parseDouble(exp.substring(0,exp.indexOf(op)));
        //      b =
        Double.parseDouble(exp.substring(exp.indexOf(op)+1,exp.length()));
        //=====
        String[] oprnds = exp.split("[\\+\\-\\*\\/]");
        a = Double.parseDouble(oprnds[0]);
        b = Double.parseDouble(oprnds[1]);
        op = exp.charAt(oprnds[0].length());
        //=====
        switch(op) {
            case '+':ans=a+b;
```

```
        break;
        case '-':ans=a-b;
        break;
        case '*':ans=a*b;
        break;
        case '/':ans=a/b;
        break;
    }
    System.out.println("Answer: "+ans);
}
}
```

OUTPUT :

```
Please enter valid expression: 4 + 6
Given expression: 4 + 6
Answer: 10.0
```

Practical – 14

- **Write a program that creates an Array List and adds a Loan object , a Date object , a string, and a Circle object to the list, and use a loop to display all elements in the list by invoking the object's to String() method.**

INPUT :

```
import java.util.ArrayList;
```

```
import java.util.Date;
```

```
public class p14 {  
    public static void main(String[] args) {  
        ArrayList<Object> list = new ArrayList<>();  
        Loan loanAcc = new Loan(1, "Amit", 36, 8.6f, 100000);  
        list.add(loanAcc);  
        Date currDate = new Date();  
        list.add(currDate);  
        String str = "This is just a description.";  
        list.add(str);  
        MyCircleClass circle = new MyCircleClass(4);  
        list.add(circle);  
        for(int i=0;i<list.size();i++) {  
            System.out.println(list.get(i).toString());  
        }  
    }  
}
```

```
class Loan {  
    private int accNumber;  
    private String accHolderName;  
    private int tenure;  
    private float intRate;  
    private long amount;
```

```

    Loan(int accNumber,String accHolderName,int tenure,float intRate, long
amount)
    {
        this.accNumber=accNumber;
        this.accHolderName=accHolderName;
        this.tenure=tenure;
        this.intRate=intRate;
        this.amount=amount;
    }
    public long getAmount() {
        return amount;
    }
    public void setAmount(long amount) {
        this.amount = amount;
    }
    public int getAccNumber() {
        return accNumber;
    }
    public void setAccNumber(int accNumber) {
        this.accNumber = accNumber;
    }
    public String getAccHolderName() {
        return accHolderName;
    }
    public void setAccHolderName(String accHolderName) {
        this.accHolderName = accHolderName;
    }
    public int getTenure() {
        return tenure;
    }
    public void setTenure(int tenure) {
        this.tenure = tenure;
    }
}

```

```

public float getIntRate() {
    return intRate;
}
public void setIntRate(float intRate) {
    this.intRate = intRate;
}
public String toString() {
    String str = "Loan object details: \n  AccountNo: "+getAccNumber()
        +",\n  Account Holder Name: " + getAccHolderName()
        +",\n  Amount: " + getAmount()
        +",\n  Tenure: " + getTenure()
        +",\n  Interest Rate: " + getIntRate();
    return str;
}
}

```

```

class MyCircleClass{
    private double radius;

    MyCircleClass(double radius){
        this.radius = radius;
    }
    public double getRadius() {
        return radius;
    }
    public void setRadius(double radius) {
        this.radius = radius;
    }
    public double getArea() {
        return Math.PI * radius * radius;
    }
    public String toString() {
        String str = "Circle object details:\n  Radius: "

```

```
        + getRadius() + ",\n    Area: " + getArea());  
    return str;  
}  
}
```

OUTPUT :

```
Loan object details:  
  AccountNo: 1,  
  Account Holder Name: Amit,  
  Amount: 100000,  
  Tenure: 36,  
  Interest Rate: 8.6  
Wed May 05 14:28:59 IST 2021  
This is just a description.  
Circle object details:  
  Radius: 4.0,  
  Area: 50.26548245743669
```

HFI

Practical – 15

- **Write the bin2Dec (string binary String) method to convert a binary string into a decimal number. Implement the bin2Dec method to throw a Number Format Exception if the string is not a binary string.**

INPUT :

```
import java.util.Scanner;

public class p15 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Please enter a Binary Number: ");
        String binaryNumber = input.next();
        try {
            int decNumber = bin2Dec(binaryNumber);
            System.out.print("Decimal number: "+decNumber);
        } catch (NumberFormatException ex) {
            System.out.println("Error occurred: "+ex.getMessage());
        }
        input.close();
    }

    public static int bin2Dec(String binaryNumber) throws
    NumberFormatException {
        int result = 0;
        for(int i=0;i<binaryNumber.length();i++)
        {
            char currBinVal =
binaryNumber.charAt(binaryNumber.length()-i-1);
            if(currBinVal != '0' && currBinVal != '1')
                throw new NumberFormatException("Given number is
not a binary number: " + binaryNumber);
            else if(currBinVal == '1')
                result = result + (int)Math.pow(2, i);
        }
    }
}
```



```
    }  
    return result;  
}  
}
```

OUTPUT :

```
Please enter a Binary Number: 0100  
Decimal number: 4
```

Practical – 16

- **Write a program that prompts the user to enter a decimal number and displays the number in a fraction. Hint: Read the decimal number as a string, extract the integer part and fractional part from the string.**

INPUT :

```
import java.util.Scanner;
```

```
public class p16
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        Scanner input = new Scanner(System.in);
```

```
        System.out.print("Please enter a decimal number: ");
```

```
        String number = input.next();
```

```
        if(number.split("\\. ").length!=2) {
```

```
            System.out.println("Given number is not a valid decimal  
number.");
```

```
            System.exit(0);
```

```
        }
```

```
        Rational rNum = new Rational(number);
```

```
        System.out.println("Number in a fraction: "
```

```
        + rNum.getnumerator() + " / " + rNum.getDenominator());
```

```
        input.close();
```

```
    }
```

```
}
```

```
class Rational
```

```
{
```

```
    private int numerator;
```

```
    private int denominator;
```

```

Rational(String number){
    int decPos = number.indexOf('.');
    this.denominator = (int) Math.pow(10, number.length()-decPos-1);
    this.numerator=((int)(Double.parseDouble(number)*denominator));
    normalize();
}
Rational(int numerator, int denominator){
    this.setnumerator(numerator);
    this.setDenominator(denominator);
    normalize();
}
public int getnumerator() {
    return numerator;
}
public void setnumerator(int numerator) {
    this.numerator = numerator;
}
public int getDenominator() {
    return denominator;
}
public void setDenominator(int denominator) {
    this.denominator = denominator;
}

private int gcd(int num1,int num2) {
    int gcd = 1;
    for(int i=1;i<num1;i++) {
        if(num1%i == 0 && num2%i==0)
            gcd=i;
    }
    return gcd;
}
public void normalize() {

```

```
        int gcd = gcd(numerator,denominator);
        this.numerator = this.numerator/gcd;
        this.denominator = this.denominator/gcd;
    }
}
```

OUTPUT :

```
Please enter a decimal number: 2.1
Number in a fraction: 21 / 10
```

Practical – 17

- **Write a program that displays a tic-tac-toe board. A cell may be X, O, or empty. What to display at each cell is randomly decided. The X and O are images in the files X.gif and O.gif.**

INPUT :

```
import java.util.Scanner;
```

```
public class p17 {
```

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

```
        String[] playerName = new String[2];
```

```
        char[][] game = new char[3][3];
```

```
        char sign;
```

```
        int currPlayer = 0,row,col,times=0,wins=-1;
```

```
        Scanner input = new Scanner(System.in);
```

```
        System.out.println("=====");
```

```
        System.out.println("Welcome to the game TicTacToe");
```

```
        System.out.println("=====");
```

```
        System.out.print("Please enter Player 1 name: ");
```

```
        playerName[0] = input.next();
```

```
        System.out.println("Hi, "+playerName[0]+". Your sign is: O.");
```

```
        System.out.println("-----");
```

```
        System.out.print("Please enter Player 2 name: ");
```

```
        playerName[1] = input.next();
```

```
        System.out.println("Hi, "+playerName[1]+". Your sign is: X.");
```

```
        System.out.println("=====");
```

```
        System.out.println("Current canvas of game is empty.");
```

```
        System.out.println("Let us start game now.");
```

```
        System.out.println();
```

```
        System.out.println("-----");
```

```

while(true)
{
    if(currPlayer==1) sign='x'; else sign='o';
    System.out.println(playerName[currPlayer]+" (" +sign+" ) shall
enter a position:");
    System.out.print("Enter Row no: ");
    row = input.nextInt()-1;
    System.out.print("Enter Col no: ");
    col = input.nextInt()-1;
    if(row>=3 || row<0 || col>=3 || col<0 ||
game[row][col]!='\0'){
        System.out.println("Please enter any valid position from
(3x3) empty cells.");
        continue;
    }
    times++;
    wins=setGame(game,row,col,currPlayer,sign);
    System.out.println("Game Status:");
    printGame(game);
    System.out.println("-----");
    if(wins!=-1 || times==9)
        break;
    if(currPlayer==0) currPlayer=1; else currPlayer=0;
}
System.out.println("====End of Game====");
if(wins==-1)
    System.out.println("No one wins.");
else
    System.out.println(playerName[wins]+" wins.");
System.out.println("====");
}

```

```

public static int setGame(char[][] game,int row,int col,int currPlayer,char
sign) {
    game[row][col]=sign;

    if(game[row][0]==game[row][1] && game[row][1]==game[row][2])
        return currPlayer;
    if(game[0][col]==game[1][col] && game[1][col]==game[2][col])
        return currPlayer;
    if(row==col)
        if(game[0][0]==game[1][1] && game[1][1]==game[2][2])
            return currPlayer;
    if(row==2-col)
        if(game[0][2]==game[1][1] && game[1][1]==game[2][0])
            return currPlayer;

    return -1;
}

public static void printGame(char[][] game){
    for(int i=0;i<3;i++) {
        for(int j=0;j<3;j++){
            if(game[i][j]!='0')
                System.out.printf("%2c",game[i][j]);
            else
                System.out.printf("%2c",' ');
            if(j!=2)
                System.out.print(" |");
        }
        System.out.println();
        if(i!=2)
            System.out.println("----+----+----");
    }
}
}

```

OUTPUT :

```
=====
Welcome to the game TicTacToe
=====
Please enter Player 1 name: A
Hi, A. Your sign is: O.
-----
Please enter Player 2 name: B
Hi, B. Your sign is: X.
=====
Current canvas of game is empty.
Let us start game now.

-----
A (o) shall enter a position:
Enter Row no: 1
Enter Col no: 2
Game Status:
  | o |
---+---+---
  |  |
---+---+---
  |  |
-----
B (x) shall enter a position:
Enter Row no: 2
Enter Col no: 1
Game Status:
  | o |
---+---+---
x |  |
---+---+---
  |  |
-----
A (o) shall enter a position:
Enter Row no: 1
Enter Col no: 3
Game Status:
  | o | o
---+---+---
x | x |
---+---+---
  |  |
-----
=====End of Game=====
A wins.
=====
```

Practical – 18

- Write a program that moves a circle up, down, left or right using arrow keys.

INPUT :

```
import javafx.application.Application;
```

```
import javafx.scene.Scene;
```



```

import javafx.scene.shape.Circle;

import javafx.scene.layout.Pane;

import javafx.geometry.Insets;

import javafx.stage.Stage;


public class p18 extends Application {

    @Override

    public void start(Stage primaryStage) {

        Pane pane = new Pane();

        pane.setPadding(new Insets(20, 20, 20, 20));

        Circle circle = new Circle(20, 20, 20);

        pane.getChildren().add(circle);


        pane.setOnKeyPressed(e -> {

            switch (e.getCode()) {

                case UP : circle.setCenterY(circle.getCenterY() >
                    circle.getRadius() ? circle.getCenterY() - 10 :
                    circle.getCenterY()); break;

                case DOWN : circle.setCenterY(circle.getCenterY() <
                    pane.getHeight() - circle.getRadius() ?
                    circle.getCenterY() + 10 : circle.getCenterY());

                break;
            }
        });
    }
}

```

```

        case LEFT : circle.setCenterX(circle.getCenterX() >
            circle.getRadius() ? circle.getCenterX() - 10 :
            circle.getCenterX()); break;
        case RIGHT : circle.setCenterX(circle.getCenterX() <
            pane.getWidth() - circle.getRadius() ?
            circle.getCenterX() + 10: circle.getCenterX());
    }
});

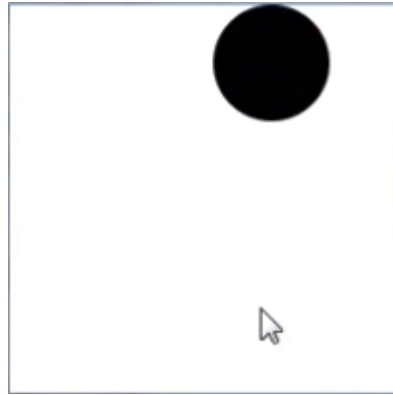
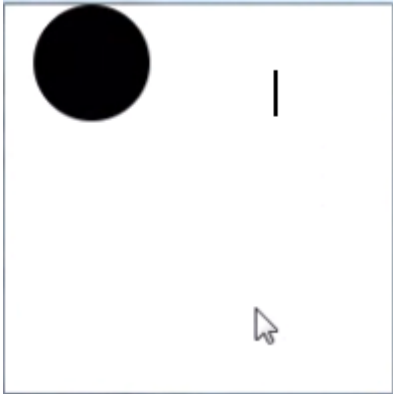
```

```

Scene scene = new Scene(pane, 200, 200);
primaryStage.setTitle("OOP Practical 18 ");
primaryStage.setScene(scene);
primaryStage.show();
pane.requestFocus();
}
}

```

OUTPUT :



Practical – 19

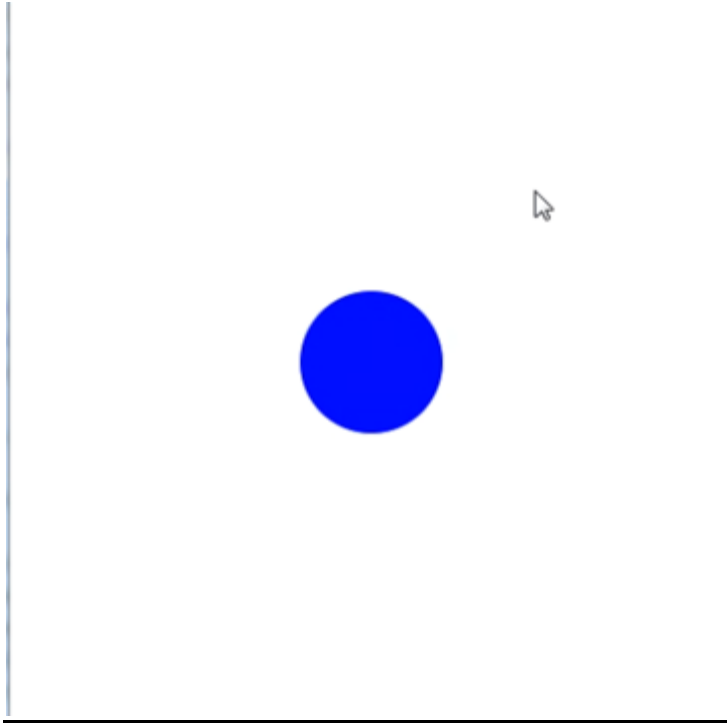
- **Write a program that displays the color of a circle as red when the mouse button is pressed and as blue when the mouse button is released.**

INPUT :

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.StackPane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;

public class p19 extends Application
{
    @Override
    public void start(Stage primaryStage)
    {
        Circle c = new Circle(200,200,50, Color.BLUE);
        StackPane pane = new StackPane(c);
        primaryStage.setScene(new Scene(pane, 600, 600));
        pane.setOnMousePressed(e -> c.setFill(Color.RED));
        pane.setOnMouseReleased(e -> c.setFill(Color.BLUE));
        primaryStage.setTitle("OOP-I Practical 19");
        primaryStage.show();
    }
    public static void main(String[] args) {
        Application.launch(args);
    }
}
```

OUTPUT :



Practical – 20

- **Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed.**

INPUT :

```
package sample;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.shape.Circle;
import javafx.scene.layout.Pane;
import javafx.geometry.Insets;
import javafx.stage.Stage;

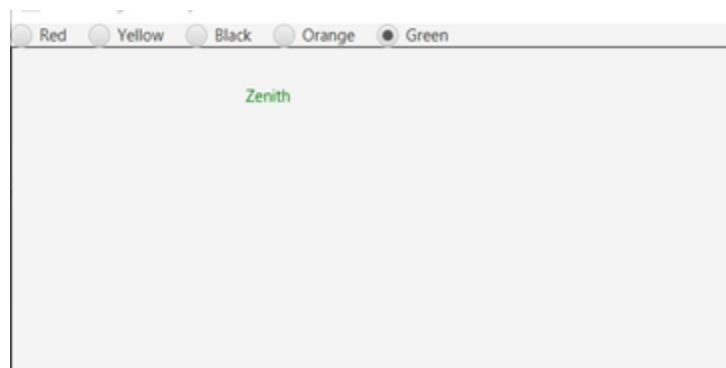
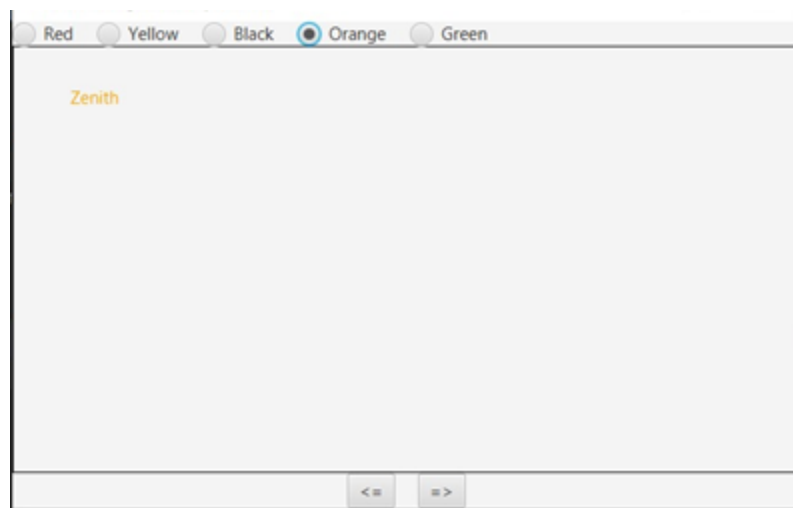
public class Practical18 extends Application {
    @Override
    public void start(Stage primaryStage) {
        Pane pane = new Pane();
        pane.setPadding(new Insets(20, 20, 20, 20));
        Circle circle = new Circle(20, 20, 20);
        pane.getChildren().add(circle);

        pane.setOnKeyPressed(e -> {
            switch (e.getCode()) {
                case UP :circle.setCenterY(circle.getCenterY() >
circle.getRadius() ? circle.getCenterY() - 10 :
circle.getCenterY()); break;
                case DOWN :circle.setCenterY(circle.getCenterY() <
pane.getHeight() - circle.getRadius() ?
circle.getCenterY() + 10 : circle.getCenterY());
                break;
                case LEFT :circle.setCenterX(circle.getCenterX() >
circle.getRadius() ? circle.getCenterX() - 10 :
circle.getCenterX()); break;
                case RIGHT :circle.setCenterX(circle.getCenterX() <
pane.getWidth() - circle.getRadius() ?
```

```
circle.getCenterX() + 10: circle.getCenterX());  
    }  
});
```

```
Scene scene = new Scene(pane, 200, 200);  
primaryStage.setTitle("OOP Practical 20");  
primaryStage.setScene(scene);  
primaryStage.show();  
pane.requestFocus();  
}  
}
```

OUTPUT :



Practical – 21

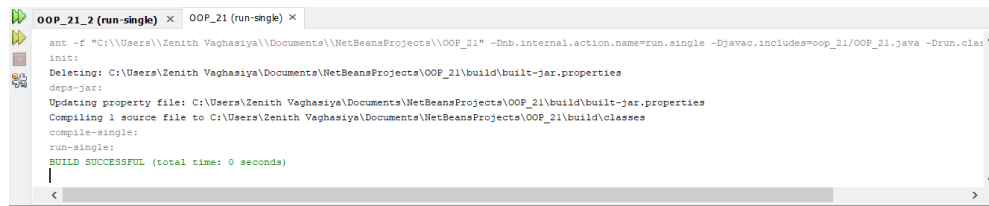
- **Write a program to create a file name 123.txt, if it does not exist. Append a new data to it if it already exist. write 150 integers created randomly into the file using Text I/O. Integers are separated by space.**

INPUT :

```
import java.io.*;
import java.util.Scanner;
public class practical21
{
    public static void main(String[] args)
    {

        try (
        PrintWriter pw = new PrintWriter(new FileOutputStream(new
        File("123.txt"), true));
        ) {
            for (int i = 0; i < 150; i++)
            {
                pw.print((int)(Math.random() * 150) + " ");
            }
        }
        catch (FileNotFoundException fnfe)
        {
            System.out.println("Cannot create the file.");
            fnfe.printStackTrace();
        }
    }
}
```

OUTPUT :



```
ant -f "C:\Users\Zenith Vaghasiya\Documents\NetBeansProjects\OOP_21" -Dnb.internal.action.name=run.single -Djavac.includes=oop_21/OOP_21.java -Drun.class=OOP_21/OOP_21.class
init:
Deleting: C:\Users\Zenith Vaghasiya\Documents\NetBeansProjects\OOP_21\build\build-jar.properties
done-jar:
Updating property file: C:\Users\Zenith Vaghasiya\Documents\NetBeansProjects\OOP_21\build\build-jar.properties
Compiling 1 source file to C:\Users\Zenith Vaghasiya\Documents\NetBeansProjects\OOP_21\build\classes
compile-single:
run-single:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Practical – 22

- **Write a recursive method that returns the smallest integer in an array. Write a test program that prompts the user to enter an integer and display its product.**

INPUT :

```
import java.util.Scanner;
```

```
public class p22
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        Scanner input = new Scanner(System.in);
```

```
        System.out.print("Enter five integers: ");
```

```
        int[] list = new int[5];
```

```
        int product = 1;
```

```
        for (int i = 0; i < list.length; i++)
```

```
        {
```

```
            list[i] = input.nextInt();
```

```
            product *= list[i];
```

```
        }
```

```
        System.out.println("The smallest element from all the elements is: "
                           + min1(list));
```

```
        System.out.println("The Product of all the elements is: " + product);
```

```
    }
```

```
    public static int min1(int[] list)
```

```
    {
```

```
        int min = list[list.length - 1];
```

```
        int index = list.length - 1;
```

```

    return min2(list, index, min);
}

private static int min2(int[] list, int index, int min)
{
    if (index < 0)
    {
        return min;
    }
    else if (list[index] < min)
    {
        return min2(list, index - 1, list[index]);
    }
    else
    {
        return min2(list, index - 1, min);
    }
}
}

```

OUTPUT :

```

Enter five integers:
1
2
3
4
5
The smallest element from all the elements is: 1
The Product of all the elements is: 120

```

Practical – 23

- Write a generic method that returns the minimum elements in a two dimensional array.

INPUT :

```

import java.util.*;

public class p23
{
    public static void main(String[] args)
    {
        Integer[][] list = new Integer[10][10];
        int value = 0;
        for (int i = 0; i < list.length; i++)
        {
            for (int j = 0; j < list[i].length; j++)
            {
                list[i][j] = ++value;
            }
        }
        System.out.println("Min = " + min(list));
    }

    public static <E extends Comparable<E>> E min(E[][] list)
    {
        E min = list[0][0];
        for (E[] elements : list)
        {
            for (E element : elements)
            {
                if (element.compareTo(min) < 0)
                {
                    min = element;
                }
            }
        }
        return min;
    }
}

```

```
}
```

OUTPUT :

```
Min = 1
```

Practical – 24

- **Define MyPriorityQueue class that extends PriorityQueue to implement the Cloneable interface and implement the clone() method to clone a priority queue.**

INPUT :

```
import java.util.PriorityQueue;
```

```
public class p24
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        MyPriorityQueue<String> queue = new MyPriorityQueue<>();
```

```
        queue.offer("1");
```

```
        queue.offer("2");
```

```
        queue.offer("3");
```

```
        MyPriorityQueue<String> queue1 = null;
```

```
        try
```

```
        {
```

```
            queue1 = (MyPriorityQueue<String>)(queue.clone());
```

```
        }
```

```
        catch (CloneNotSupportedException e)
```

```
        {
```

```
            e.printStackTrace();
```

```
        }
```

```
        System.out.print(queue1);
```

```
    }
```

```
    static class MyPriorityQueue<E> extends PriorityQueue<E> implements  
    Cloneable
```

```
    {
```

```
        @Override
```

```
public Object clone() throws CloneNotSupportedException
{
    MyPriorityQueue<E> clone = new MyPriorityQueue<>();
    this.forEach(clone::offer);
    return clone;
}
}
```

OUTPUT :

```
[1, 2, 3]
```

Practical – 25

- **Write a program that reads words from a text file and displays all the nonduplicate words in descending order. The text file is passed as a command-line argument.**

INPUT :

```
class p25 {
    public static void main(String[] args) {
        // if(args.length==1){
        // String fileName = args[0];
        String fileName = "C://list.txt";
        TreeSet<String> set = new TreeSet<>();
        File file = new File(fileName);
        try {
            Scanner s = new Scanner(file);
            while (s.hasNext()){
                set.add(s.next());
            }
            System.out.println(set);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        // }
        // else{
        // System.out.println("Please, Pass the File Name as Command Line
Argument");
        // }
    }
}
```