Computer Engineering
Semester-4
Object Oriented Programming
[3140705]

Practical 1 to 25

Name : Radadiya Param K.

Enrollment no. : 190130107118

Branch : Computer Engineering

Division : B

Batch : B2

Practical 1

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

Code:

```
package com.company;

import java.time.LocalDateTime;
public class Main {

    public static void main(String[] args) {
        LocalDateTime t1 = LocalDateTime.now();
        System.out.println(t1);
        System.out.println("190130107118 Param Radadiya
4CEB2");
        System.out.println("Practical - 1");
        System.out.println("Welcome to Java, Learning Java Now
and Programming is fun.");
    }
}
```

Output :

```
2020-12-29T13:22:16.923913400
190130107118 Param Radadiya 4CEB2
Practical - 1
Welcome to Java, Learning Java Now and 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=ed-bf/ad-bc
cx+dy=f y=af-ec/ad-bc

Code:

```
package com.company;

import java.time.LocalDateTime;
public class practical2 {

    public static void main(String[] args) {
        LocalDateTime t1 = LocalDateTime.now();
        System.out.println(t1);
        System.out.println("190130107118 Param Radadiya
4CEB2");
        System.out.println("Practical - 2\n");
        double a=3.4,b=50.2,e=44.5,c=2.1,d=0.55,f=5.9,x=0,y=0;
        x=((e*d)-(b*f))/((a*d)-(b*c));
        y=((a*f)-(e*c))/((a*d)-(b*c));
        System.out.println("x="+x +"\ny="+ y);
    }
}
```

Output :

```
2020-12-29T13:49:18.802971600
190130107118 Param Radadiya 4CEB2
Practical - 2

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.

Code:

```
package com.company;

import java.time.LocalDateTime;
public class practical3
{
   public static void main(String[] args)
   {
      LocalDateTime t1 = LocalDateTime.now();
      System.out.println(t1);
      System.out.println("190130107118 Param Radadiya
4CEB2");
      System.out.println("Practical - 3\n");
      float f,m= 2.020F;
      f = m * 3.281F;
      System.out.println("In Meters: "+m +"\nIn Feet: "+f);
   }
}
```

Output :

```
2020-12-29T13:54:45.739479300
190130107118 Param Radadiya 4CEB2
Practical - 3

In Meters: 2.02
In Feet: 6.6276197
```

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.

Code : Next page

Code:

```
package com.company;

import java.time.LocalDateTime;
import java.util.Scanner;

public class practical4
{
   public static void main(String[] args)
   {
      LocalDateTime t1 = LocalDateTime.now();
      System.out.println(t1);
      System.out.println("190130107118 Param Radadiya 4CEB2");
      System.out.println("Practical - 4\n");

      Scanner sc = new Scanner(System.in);

      System.out.print("Enter Weight (in pounds) : ");
      Float W = sc.nextFloat();

      System.out.print("Enter Height (in inches) : ");
      Float H = sc.nextFloat();

      W = (float) (0.4536 * W);
      H = (float) (H * 0.0254);

      float BMI = W/(H * H);

      System.out.println("Body Mass Index is " + BMI);
   }
}
```

Output :

```
2021-01-11T16:20:58.636242200
190130107118 Param Radadiya 4CEB2
Practical - 4

Enter Weight (in pounds) : 140
Enter Height (in inches) : 45
Body Mass Index is 48.608097
```

Practical 5

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

Code : Next page

Code:

```java
package com.company;

import java.math.BigInteger;
import java.time.LocalDateTime;
import java.util.Scanner;

public class practical5
{
    public static void main(String[] args)
    {
        LocalDateTime t1 = LocalDateTime.now();
        System.out.println(t1);
        System.out.println("190130107118 Param Radadiya 4CEB2");
        System.out.println("Practical - 5\n");

        Scanner input = new Scanner(System.in);

        System.out.print("Enter three integers:");
        int num1 = input.nextInt();
        int num2 = input.nextInt();
        int num3 = input.nextInt();
        int temp = 0;

        if (num1 > num2) {
            temp = num1;
            num1 = num2;
            num2 = temp;
        }
        if (num2 > num3) {
            temp = num2;
            num2 = num3;
            num3 = temp;
        }
        if (num1 > num2) {
            temp = num1;
            num1 = num2;
            num2 = temp;
       }
        System.out.println("Descending order : " + num1+" "+num2+" "+num3);

    }
}
```

Output :

```
2021-01-11T16:45:42.787320100
190130107118 Param Radadiya 4CEB2
Practical - 5

Enter three integers:20 5 6
Descending order : 5 6 20
```

Practical 6

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

Code : Next page

Code:

```
package com.company;

import java.math.BigInteger;
import java.time.LocalDateTime;
import java.util.Scanner;

public class practical6 {
    public static void main(String[] args) {
        LocalDateTime t1 = LocalDateTime.now();
        System.out.println(t1);
        System.out.println("190130107118 Param Radadiya 4CEB2");
        System.out.println("Practical - 6\n");

        Scanner input = new Scanner(System.in);

        System.out.print("Enter one character : ");
        char c = input.next().charAt(0);

        boolean lowercase_vowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

        boolean uppercase_vowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');

        if (lowercase_vowel || uppercase_vowel)
            System.out.println(c + " is a vowel.");
        else
            System.out.println(c + " is a consonant.");


    }

}
```

Output :

```
2021-01-11T17:13:00.767579900
190130107118 Param Radadiya 4CEB2
Practical - 6

Enter one character : z
z is a consonant.
```

Practical 7

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

Code:

```
package com.company;

import java.time.LocalDateTime;
import java.util.Scanner;

public class practical7 {
   public static void main(String[] args) {
      LocalDateTime t1 = LocalDateTime.now();
      System.out.println(t1);
      System.out.println("190130107118 Param Radadiya 4CEB2");
      System.out.println("Practical - 7\n");

      StringBuilder s = new StringBuilder();
      for (int i = 0; i < 3; i++) {
         char ch = (char) (Math.random() * 26 + 'A');
         s.append(ch);
      }
      for (int i = 0; i < 4; i++) {
         char digit = (char) (Math.random() * 10 + '0');
         s.append(digit);
      }
      System.out.println("Random vehicle plate number : " + s);

   }
}
```

Output :

```
2021-01-11T17:13:48.132408200
190130107118 Param Radadiya 4CEB2
Practical - 7

Random vehicle plate number : GZ00785
```

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.

Code:

```
package com.company;

import java.time.LocalDateTime;
import java.util.Scanner;

public class practical8 {
   public static void main(String[] args) {
      LocalDateTime t1 = LocalDateTime.now();
      System.out.println(t1);
      System.out.println("190130107118 Param Radadiya 4CEB2");
      System.out.println("Practical - 8\n");

      Scanner input = new Scanner(System.in);

      System.out.print("Enter Number : ");
      Integer num = input.nextInt();

      System.out.print("Factor of "+num + " is ");
      int i=2;
      while (num!=1){
         while(num%i==0){
            num=num/i;
            System.out.print(i + " ");
         }
         i++;
      }

   }
}
```

Output :

```
2021-01-11T17:25:15.029222900
190130107118 Param Radadiya 4CEB2
Practical - 8

Enter Number : 120
Factor of 120 is 2 2 2 3 5
```

Practical 9

Write a method with the 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.

GCD: (Greatest Common Divisor)

Int O=1
Step-1: Take
N1 = _____
N2 = _____
Step-2: find min number
Step-3: Iterate i=2 to min
if(N1%i==0 && N2%i==0)
O = O * i
Step-4: Print O

Code : Next page

Code:

```
package com.company;
import java.time.LocalDateTime;
import java.util.Scanner;

public class practical9 {

    public static int gcd(int num1, int num2)
    {
        while (num1 != num2)
        {
            if(num1 > num2)
            {
                num1 = num1 - num2;
            }
            else
            {
                num2 = num2 - num1;
            }
        }
        return num1;
    }
    public static void main(String[] args)
    {
        LocalDateTime time = LocalDateTime.now();
        System.out.println(time);

        System.out.println("190130107118 Param Radadiya 4CEB2");
        System.out.println("Practical - 9\n");

        Scanner input = new Scanner(System.in);
        System.out.print("Enter First Number  : ");
        int number1 = input.nextInt();
        System.out.print("Enter Second Number : ");
        int number2 = input.nextInt();
        System.out.print("GCD of "+number1+" and "+number2+" =
"+gcd(number1, number2));
    }
  }
```

Output :

```
2021-01-22T20:05:25.704321200
190130107118 Param Radadiya 4CEB2
Practical - 9

Enter First Number  : 20
Enter Second Number : 30
GCD of 20 and 30 = 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.

Array
Input : 5,3,8,2,9,1,4,6,10,7
Output: 7,10,6,4,1,9,2,8,3,5

Code : Next page

Code:

```
package com.company;
import java.time.LocalDateTime;
import java.util.Scanner;


public class practical10
{
   public static void reverse(int numbers[])
   {
      int j=0,temp;
      while(j<=numbers.length/2)
      {
         temp=numbers[j];
         numbers[j]=numbers[numbers.length-1-j];
         numbers[numbers.length-1-j]=temp;
         j++;
      }
   }
   public static void main(String[] args)
   {
      int i=0;
      int num_array[]=new int[10];
      Scanner input = new Scanner(System.in);

      LocalDateTime time = LocalDateTime.now();
      System.out.println(time);

      System.out.println("190130107118 Param Radadiya 4CEB2");
      System.out.println("Practical - 10");

      for(i=0;i<10;i++)
      {
         System.out.print("Enter number - "+ (i+1) + " : ");
         num_array[i] = input.nextInt();
      }
      reverse(num_array);
      System.out.print("After reversing : ");
      for(i=0;i<10;i++)
      {
         System.out.print(" "+num_array[i]);
      }
   }
}
```

Output :

```
2021-01-22T20:22:52.679277700
190130107118 Param Radadiya 4CEB2
Practical - 10
Enter number - 1 : 10
Enter number - 2 : 2
Enter number - 3 : 36
Enter number - 4 : 5
Enter number - 5 : 25
Enter number - 6 : 25
Enter number - 7 : 25
Enter number - 8 : 69
Enter number - 9 : 5
Enter number - 10 : 5
After reversing :   5  5  69  25  25  25  5  36  2  10
Process finished with exit code 0
```

Practical 11 (main)

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 has an odd number of 1's.

Array[6][6] -- fill this with 0,1.

Output:
Status True/False
True : when ever raw and column have odd number of 1's
False: when above is not

Code : Next page

Code:

```
package com.company;
import java.time.LocalDateTime;
import java.util.Scanner;


public class trial {

    static int N = 6;
    static int[][] m = input(6,6);;
    static Boolean status = false;

    public static int[][] input(int row, int columns) {

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

        int[][] m = new int[row][columns];

        System.out.println("Enter value for 6*6 matrix (only 0-1) : ");
        for (int i = 0; i < m.length; i++) {
            for (int j = 0; j < m[i].length; j++) {
                m[i][j] = input.nextInt();
            }
        }
        System.out.println();

        return m;
    }
public static boolean validators(int[][] m) {

    for (int i = 0; i < N; i++) {
        status = checkOdd(m , i, N, true);
        if (status==false)
            break;
    }
```

Code:

```
    for (int j = 0; j < N; j++) {
        status = checkOdd(m, j, N, false);
        if (status==false)
            break;
    }
    return (status);
}

private static Boolean checkOdd(int[][] a, int s, int e, Boolean f) {
    boolean b = true;
    int t=0;

    if (f) {
        int sum=0;
        for(t=0;t<e;t++) {
            sum+=a[s][t];
        }
        if(sum%2 == 0) {
            b = false;
        }
        else
            b=true;
    }
    else {
        int sum=0;
        for(t=0;t<e;t++) {
            sum+=a[t][s];
        }
        if(sum%2 == 0) {
            b = false;
        }
        else
            b=true;
    }
    return (b);
}
```

Code:

```
public static void display(int[][] m) {

   System.out.println("Entered Matrix is ");

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

public static void enrol(){
   LocalDateTime t1 = LocalDateTime.now();
   System.out.println(t1);
   System.out.println("190130107118 Param Radadiya 4CEB2");
   System.out.println("Practical - 11 (main)\n");
}
 public static void main(String[] args) {

      display(m);

      boolean output_status;
      output_status = validators(m);

      if (output_status){
         System.out.println("\nThe Array is : Valid");
         System.out.println("Every Row and Column has an odd number of 1's.");
      }

      else {
         System.out.println("The Array is : Invalid");
      }
   }
}
```

Output :

```
2021-01-24T13:29:11.977699300
190130107118 Param Radadiya 4CEB2
Practical - 11 (main)

Enter value for 6*6 matrix (only 0-1) :
1
0
0
0
0
0
0
1
0
0
0
1
0
0
1
0
0
0
1
0
0
1
1
```

```
0
0
0
0
0
1
0
1
0
0
0
1
1

Entered Matrix is
100000
010001
001000
100110
000010
100011
The Array is : Invalid
```

Practical 11(Extension-1)

Invalid at row: 0,1,2 at column:1

Code:

```
package com.company;
import java.time.LocalDateTime;
import java.util.Scanner;


public class practical11_1 {

    static int N = 6;
    static int[][] m = input(6,6);;
    static Boolean valid = true;
    static int[] rowin = new int[6];
    static int[] colin = new int[6];
    //static int rowin[], colin[];

    public static int[][] input(int row, int columns) {

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

        int[][] m = new int[row][columns];

        System.out.println("Enter value for 6*6 matrix (only 0-1) : ");
        for (int i = 0; i < m.length; i++) {
            for (int j = 0; j < m[i].length; j++) {
                m[i][j] = input.nextInt();
            }
        }
        System.out.println();

        return m;
    }
```

```
public static void validators(int[][] m) {
    int sum = 0;
    for(int i=0;i<6;i++)
    {
        for(int j=0; j<6;j++)
        {
            System.out.print(m[i][j]);
            sum = sum + m[i][j];
        }
        System.out.println("");

        if(sum%2==1)
        {
            rowin[i] = 1;
        }
        else{
            valid=false;
            rowin[i]=0;
        }
        sum=0;
    }

    for(int i=0;i<6;i++)
    {
        for(int j=0; j<6;j++)
        {
            sum = sum + m[j][i];
        }
```

```
        if(sum%2==1)
        {
            colin[i] = 1;
        }
        else{
            valid=false;
            colin[i]=0;
        }
        sum=0;
    }

}

public static void enrol(){

    LocalDateTime t1 = LocalDateTime.now();
    System.out.println(t1);
    System.out.println("190130107118 Param Radadiya 4CEB2");
    System.out.println("Practical - 11 (extension-1)\n");
}


public static void main(String[] args) {

    validators(m);

    if (valid==true){
        System.out.println("\nThe Array is Valid");
        System.out.println("Every Row and Column has an odd number of 1's.");
    }

else {
    System.out.print("The Array is Invalid at the row : ");

    for(int i=0;i<6;i++){
        if(rowin[i]==0){
            System.out.print(i + " ");
        }
    }
```

```
            System.out.print("and at the column :");

            for(int i=0;i<6;i++){
                if(colin[i]==0){
                    System.out.print(i + " ");
                }
            }

        }
    }
}
```

Output :

```
2021-01-24T15:21:26.446838100
190130107118 Param Radadiya 4CEB2
Practical - 11 (extension-1)

Enter value for 6*6 matrix (only 0-1) :
1
1
1
1
1
1
1
1
0
0
0
0
0
0
1
0
```

Output :

```
0
1
1
0
0
0
0
0
1
0
0
0
1
1
1
0
0
0
1
1

111111
110000
001001
100000
100011
100011
The Array is Invalid at the row : 0 1 2 and at the column :1 2 5
Process finished with exit code 0
```

Practical 11(Extension-2)

Convert this practical in terms of “Tic Tac Toe”

Code:

```
package com.company;
import java.time.LocalDateTime;
import java.util.*;

public class practical11_2 {

    static String[] num;
    static String turn;


    static String result()
    {
        for (int a = 0; a < 8; a++) {
            String line = null;

            switch (a) {
                case 0:
                    line = num[0] + num[1] + num[2];
                    break;
                case 1:
                    line = num[3] + num[4] + num[5];
                    break;
                case 2:
                    line = num[6] + num[7] + num[8];
                    break;
                case 3:
                    line = num[0] + num[3] + num[6];
                    break;
                case 4:
                    line = num[1] + num[4] + num[7];
                    break;
                case 5:
                    line = num[2] + num[5] + num[8];
                    break;
                case 6:
                    line = num[0] + num[4] + num[8];
                    break;
                case 7:
                    line = num[2] + num[4] + num[6];
                    break;
            }
```

```
if (line.equals("XXX")) {
            return "X";
        }

        else if (line.equals("OOO")) {
            return "O";
        }
    }

    for (int a = 0; a < 9; a++) {
        if (Arrays.asList(num).contains(
                String.valueOf(a + 1))) {
            break;
        }
        else if (a == 8) {
            return "draw";
        }
    }


    .
    System.out.println(turn + "'s turn; enter a slot number to place " + turn + " in:");
    return null;
}


static void printnum()
{
    System.out.println("| " + num[0] + " | "+ num[1] + " | " + num[2]+ " |");
    System.out.println(" ----------- ");
    System.out.println("| " + num[3] + " | "+ num[4] + " | " + num[5]+ " |");
    System.out.println(" ----------- ");
    System.out.println("| " + num[6] + " | "+ num[7] + " | " + num[8]+ " |");
}
public static void main(String[] args)
{
    Scanner in = new Scanner(System.in);
    num = new String[9];
    turn = "X";
    String winner = null;
```

```
LocalDateTime t1 = LocalDateTime.now();
System.out.println(t1);
System.out.println("190130107118 Param Radadiya 4CEB2");
System.out.println("Practical - 11 (extension-2)\n");

for (int a = 0; a < 9; a++) {
   num[a] = String.valueOf(a + 1);
}

printnum();

System.out.println("\nPlayer-1 is X");
System.out.println("Player-2 is 0");

System.out.println("\nEnter a slot number to place X at : ");

while (winner == null) {
   int numInput;

   try {
      numInput = in.nextInt();
      if (!(numInput > 0 && numInput <= 9)) {
         System.out.println(
               "\tInvalid input..... \n\tRe-enter slot number : ");
         continue;
      }
   }
   catch (InputMismatchException e) {
      System.out.println("\tInvalid input..... \n\tre-enter slot number:");
      continue;
   }


   if (num[numInput - 1].equals(
         String.valueOf(numInput))) {
      num[numInput - 1] = turn;
num[numInput - 1] = turn;

if (turn.equals("X")) {
   turn = "O";
}
```

```
else {
                turn = "X";
            }

            printnum();
            winner = result();
        }
        else {
            System.out.println("Slot already taken..... \nre-enter slot number:");
        }
    }

    if (winner.equalsIgnoreCase("draw")) {
        System.out.println("\n\nIt's a draw! \n\tThanks for playing.");
    }

    else {
        System.out.println("\n\tCongratulations! " + winner+ " have won the game...
\n\tThanks for playing.");
    }
  }
}
```

Output :

```
2021-01-24T13:38:18.809940900
190130107118 Param Radadiya 4CEB2
Practical - 11 (extension-2)

| 1 | 2 | 3 |
 -----------
| 4 | 5 | 6 |
 -----------
| 7 | 8 | 9 |

Player-1 is X
Player-2 is 0

Enter a slot number to place X at :
1
| X | 2 | 3 |
 -----------
| 4 | 5 | 6 |
 -----------
| 7 | 8 | 9 |
O's turn; enter a slot number to place O in:
2
| X | O | 3 |
 -----------
| 4 | 5 | 6 |
 -----------
| 7 | 8 | 9 |
X's turn; enter a slot number to place X in:
5
```

Output :

```
X's turn; enter a slot number to place X in:
5
| X | O | 3 |
 -----------
| 4 | X | 6 |
 -----------
| 7 | 8 | 9 |
O's turn; enter a slot number to place O in:
3
| X | O | O |
 -----------
| 4 | X | 6 |
 -----------
| 7 | 8 | 9 |
X's turn; enter a slot number to place X in:
9
| X | O | O |
 -----------
| 4 | X | 6 |
 -----------
| 7 | 8 | X |

    Congratulations! X have won the game...
    Thanks for playing.
```

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.

Code:

```
package com.company;
import java.time.LocalDateTime;

public class practical12 {

      int seed=0, n=0;
      int numbers[]= new int[1000];
      practical12(int seed)
      {
         this.seed = seed;
         for(int i=0;i<1000;i++)
         {
            int num = (int) (Math.random()*1000);
            numbers[i]=num;
         }
      }

    int nextInt(int n)
      {
         return(numbers[n]);
      }


   public static void main(String [] args)
      {
         practical12 r = new practical12 (1000);
         int i=0, k=0;
```

```
        LocalDateTime t1 = LocalDateTime.now();
         System.out.println(t1);
         System.out.println("190130107118 Param Radadiya 4CEB2");
         System.out.println("Practical - 12\n");

         while(i<100) {
            if (r.nextInt(k)>= 1 && r.nextInt(k)<=49){
               System.out.println("The random number " + (i + 1) + " : " + r.nextInt(k));
               i++;
            }
            k++;
         }
      }

}
```

Output :

```
2021-01-28T19:58:21.876116500
190130107118 Param Radadiya 4CEB2
Practical - 12

The random number 1 : 44
The random number 2 : 8
The random number 3 : 8
The random number 4 : 45
The random number 5 : 48
The random number 6 : 36
The random number 7 : 35
The random number 8 : 2
The random number 9 : 20
The random number 10 : 18
The random number 11 : 49
The random number 12 : 45
The random number 13 : 39
The random number 14 : 38
The random number 15 : 38
The random number 16 : 41
The random number 17 : 25
The random number 18 : 42
```

```
The random number 19 : 21
The random number 20 : 43
The random number 21 : 14
The random number 22 : 37
The random number 23 : 15
The random number 24 : 41
The random number 25 : 23
The random number 26 : 23
The random number 27 : 5
The random number 28 : 16
The random number 29 : 14
The random number 30 : 14
The random number 31 : 12
The random number 32 : 31
The random number 33 : 47
The random number 34 : 26
The random number 35 : 40
The random number 36 : 18
The random number 37 : 42
The random number 38 : 25
The random number 39 : 45
```

Practical 13

Write a program for calculators to accept an expression as a string in which the operands and operators are separated by zero or more spaces.

For ex: 3+4 and 3 + 4 are acceptable expressions.

Code:

```
package com.company;
import java.time.LocalDateTime;

import java.util.Scanner;

public class practical13 {

    public static boolean validation(char arg[]) {

        char[] ch = new char[3];
        int j=0;

        for(int i=0;i< arg.length;i++){
            if(arg[i]!=' ')
            {
                ch[j++] = arg[i];
            }
        }

        return ch[1] == '+' || ch[1] == '-' || ch[1] == '*' || ch[1] == '/';
    }
```

```
    public static void main (String[]args) {
        LocalDateTime t1 = LocalDateTime.now();
        System.out.println(t1);
        System.out.println("190130107118 Param Radadiya 4CEB2");
        System.out.println("Practical - 13\n");

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string : ");
        String str = sc.nextLine();

        char[] ch = new char[str.length()];

        for (int i = 0; i < str.length(); i++) {
            ch[i] = str.charAt(i);
        }

        boolean result  = validation(ch);

        if(result)
            System.out.println("Valid");
        else
            System.out.println("Invalid");
    }
}
```

Code:

```
2021-02-06T21:59:40.554838200
190130107118 Param Radadiya 4CEB2
Practical - 13

Enter a string : 1    +   9
Valid
```

```
2021-02-06T22:04:15.130232
190130107118 Param Radadiya 4CEB2
Practical - 13

Enter a string : *98
Invalid
```

```
2021-02-06T22:06:53.112806800
190130107118 Param Radadiya 4CEB2
Practical - 13

Enter a string : 4    6-
Invalid
```

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.

Code:

```
package com.company;
import java.util.*;

public class practical14 {

   public static void main(String[] args){

      ArrayList<Object> arr_list = new
ArrayList<Object>();
      arr_list.add(new Date());
      System.out.println("190130107118 Param
Radadiya 4CEB2");
      System.out.println("Practical - 14\n");

      arr_list.add(new Loan(26958));
      arr_list.add(new String("This is Param Radadiya"));
      arr_list.add(new Circle(36.9));
      for (int i = 0; i < arr_list.size(); i++)
      {
         System.out.println((arr_list.get(i)).toString());
      }
   }
}
```

```
class Circle
{
   double radius;
   Circle(double r)
   {
      this.radius=r;
   }
   public String toString()
   {
      return "Circle with Radius "+this.radius;
   }
}
class Loan
{
   double amount;
   Loan(double amt)
   {
      this.amount=amt;
   }
   public String toString()
   {
      return "Amount of the = "+this.amount;
   }
}
```

Output :

```
190130107118 Param Radadiya 4CEB2
Practical - 14

Sun Mar 21 11:28:07 IST 2021
Amount of the = 26958.0
This is Param Radadiya
Circle with Radius 36.9
```

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 NumberFormatException if the string is not a binary string.

Code:

```
package com.company;
import java.time.LocalDateTime;
import java.util.Scanner;

public class practical15 {

   public static int bin2Dec(String binaryString) throws
NumberFormatException
   {
      int decimal = 0;
      int strLength=binaryString.length();
      for (int i = 0; i < strLength; i++)
      {
         if (binaryString.charAt(i) < '0' || binaryString.charAt(i) > '1')
         {
            throw new NumberFormatException("The Input String is
not Binary");
         }

         decimal += (binaryString.charAt(i)-'0') * Math.pow(2,
strLength-1-i);
      }
      return decimal;
   }
```

```
    public static void main(String[] args){
        LocalDateTime t1 = LocalDateTime.now();
        System.out.println(t1);
        System.out.println("190130107118 Param Radadiya 4CEB2");
        System.out.println("Practical - 15\n");

        Scanner input = new Scanner(System.in);
        System.out.print("Enter Binary Value : ");
        String str = input.nextLine();
        try
        {
            System.out.println("Decimal Value = " + bin2Dec(str));
        }
        catch(NumberFormatException e)
        {
            System.out.println(e);
        }
    }
}
```

Output :

```
2021-03-21T11:43:21.310192200
190130107118 Param Radadiya 4CEB2
Practical - 15

Enter Binary Value : 1101
Decimal Value = 13
```

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.

Code:

```
package com.company;
import java.time.LocalDateTime;
import java.util.Scanner;

public class practical16 {

    public static void main(String args[])
    {
        LocalDateTime t1=LocalDateTime.now();
        System.out.println(t1);
        System.out.println("190130107118 Param Radadiya 4CEB2");
        System.out.println("Practical - 16\n");

        Scanner input=new Scanner(System.in);

        System.out.println("Enter a decimal number : ");
        String[] str=input.nextLine().split("\\.");

        System.out.println("Integer part: "+str[0]);
        System.out.println("Fractional Part: "+str[1]);
    }
}
```

Output :

```
2021-03-25T13:07:02.064533600
190130107118 Param Radadiya 4CEB2
Practical - 16

Enter a decimal number :
26.369
Integer part: 26
Fractional Part: 369
```

Practical 17

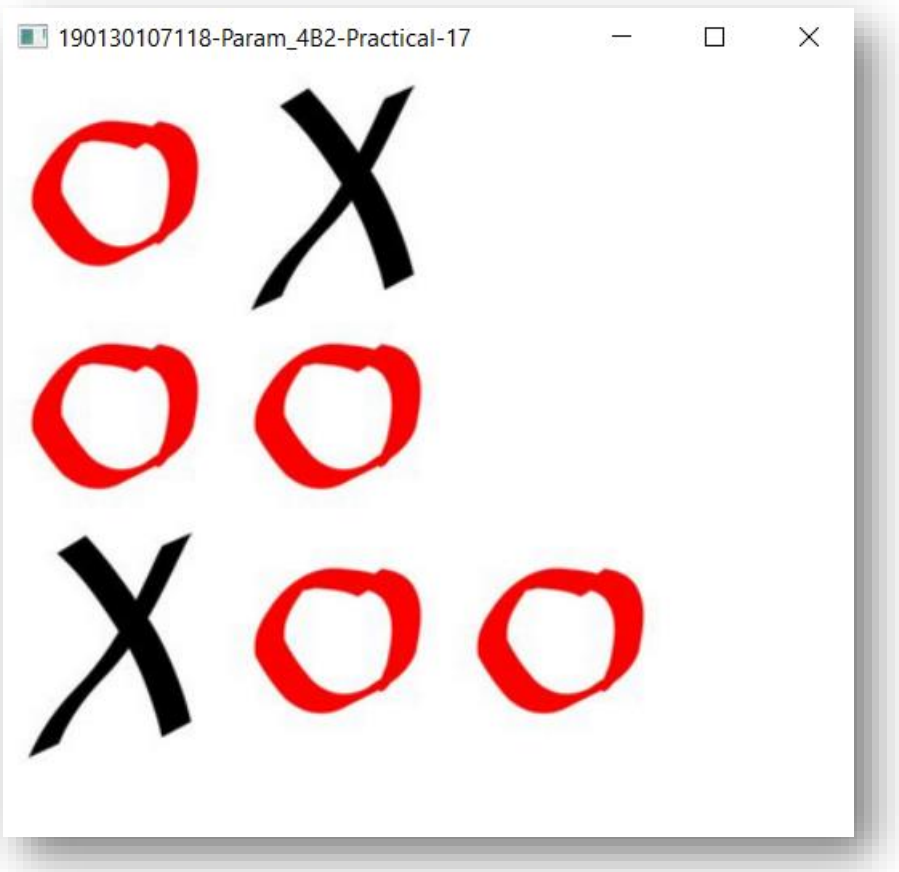
Write a program that displays a tic-tac-toe board. A cell may be X, O, or empty. What to display at each cell is randomly decided. The X and O are images in the files X.gif and O.gif.

Code:

```
package sample;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.GridPane;
import javafx.scene.control.Label;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.stage.Stage;

public class Practical17 extends Application
{
   @Override
   public void start(Stage primaryStage)
   {

      GridPane pane = new GridPane();

      for (int i = 0; i < 3; i++)
      {
         for (int j = 0; j < 3; j++)
         {
            int n = (int)(Math.random() * 3);
            if (n == 0)
               pane.add(new ImageView(new Image("images/X.png")), j, i);
            else if (n == 1)
```

```
    pane.add(new ImageView(new Image("images/O.png")), j, i);
            else {
               continue;
            }
         }
      }

      Scene scene = new Scene(pane, 120, 130);
      primaryStage.setTitle("190130107118-Param_4B2-Practical-17");
      primaryStage.setScene(scene);
      primaryStage.show();
   }
}
```

Output :



Practical 18

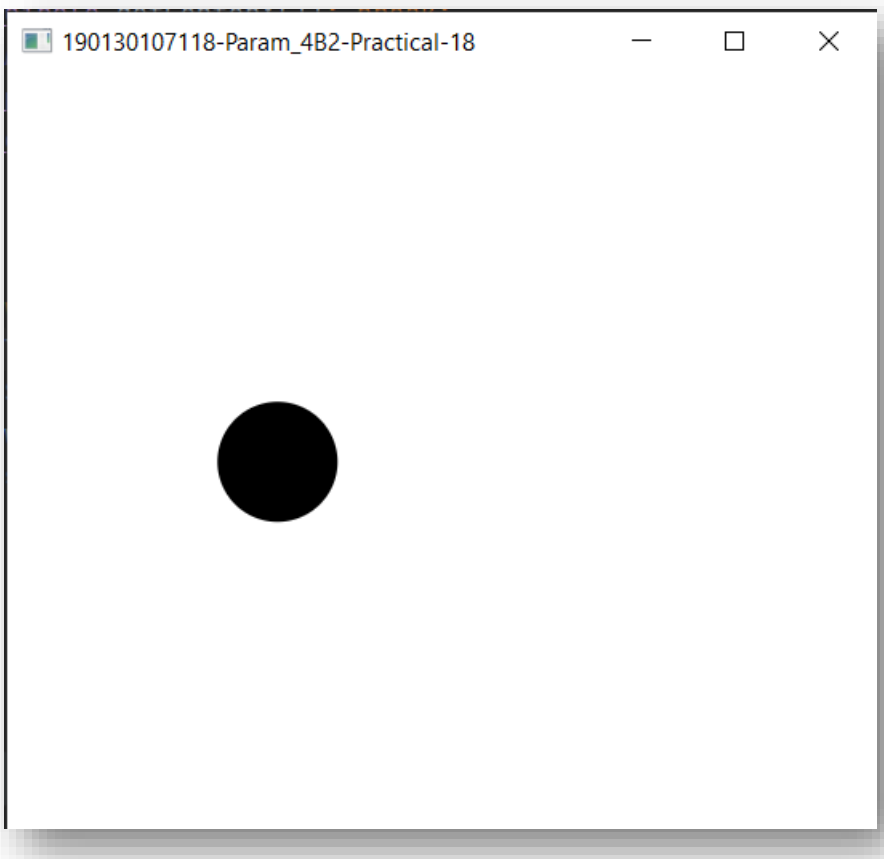
Write a program that moves a circle up, down, left or right using arrow keys.

Code:

```
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(30, 30, 30, 30));
      Circle circle = new Circle(30, 30, 30);
      pane.getChildren().add(circle);

      pane.setOnKeyPressed(e -> {
         switch (e.getCode()) {
            case UP : circle.setCenterY(circle.getCenterY() >
                  circle.getRadius() ? circle.getCenterY() - 15 :
                  circle.getCenterY()); break;
            case DOWN : circle.setCenterY(circle.getCenterY() <
                  pane.getHeight() - circle.getRadius() ?
                  circle.getCenterY() + 15 : circle.getCenterY());
               break;
```

```
            case LEFT : circle.setCenterX(circle.getCenterX() >
                        circle.getRadius() ? circle.getCenterX() - 15 :
                        circle.getCenterX()); break;
                case RIGHT : circle.setCenterX(circle.getCenterX() <
                        pane.getWidth() - circle.getRadius() ?
                        circle.getCenterX() + 15: circle.getCenterX());
            }
        });

        Scene scene = new Scene(pane, 200, 200);
        primaryStage.setTitle("190130107118-Param_4B2-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.

Code:

```
package sample;

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 Practical19 extends Application
{
   @Override
   public void start(Stage primaryStage)
   {
      double width = 450;
      double height = 450;
      Circle c = new Circle(width / 2, height / 2, Math.min(width, height) /
10, Color.BLUE);

      c.setStroke(Color.WHITE);

      StackPane pane = new StackPane(c);

      primaryStage.setScene(new Scene(pane, width, height));
      pane.setOnMousePressed(e -> c.setFill(Color.RED));
      pane.setOnMouseReleased(e -> c.setFill(Color.BLUE));
      primaryStage.setTitle("190130107118-Param_4B2-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.

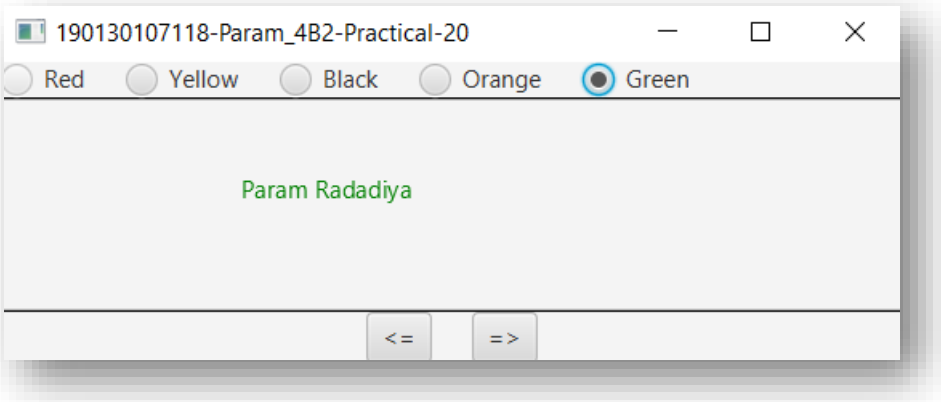
Code:

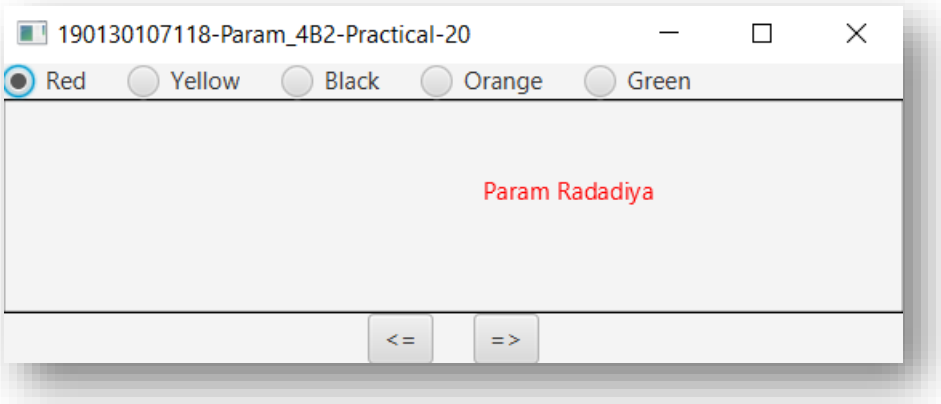
```
package sample;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.geometry.Pos;
import javafx.scene.control.Button;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.scene.layout.BorderPane;
import javafx.scene.text.Text;
import javafx.scene.control.RadioButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.paint.Color;

public class Practical20 extends Application
{
   protected Text text = new Text(50, 50, "Param Radadiya");

   @Override
   public void start(Stage primaryStage) {
      HBox paneForButtons = new HBox(20);
      Button btLeft = new Button("<=");
      Button btRight = new Button("=>");
      paneForButtons.getChildren().addAll(btLeft, btRight);
      paneForButtons.setAlignment(Pos.CENTER);
      BorderPane pane = new BorderPane();
      pane.setBottom(paneForButtons);
```

```
HBox paneForRadioButtons = new HBox(20);
RadioButton rbRed = new RadioButton("Red");
RadioButton rbYellow = new RadioButton("Yellow");
RadioButton rbBlack = new RadioButton("Black");
RadioButton rbOrange = new RadioButton("Orange");
RadioButton rbGreen = new RadioButton("Green");
paneForRadioButtons.getChildren().addAll(rbRed, rbYellow,
        rbBlack, rbOrange, rbGreen);

ToggleGroup group = new ToggleGroup();
rbRed.setToggleGroup(group);
rbYellow.setToggleGroup(group);
rbBlack.setToggleGroup(group);
rbOrange.setToggleGroup(group);
rbGreen.setToggleGroup(group);

Pane paneForText = new Pane();
paneForText.setStyle("-fx-border-color: black");
paneForText.getChildren().add(text);
pane.setCenter(paneForText);
pane.setTop(paneForRadioButtons);

btLeft.setOnAction(e -> text.setX(text.getX() - 10));
btRight.setOnAction(e -> text.setX(text.getX() + 10));

rbRed.setOnAction(e -> {
    if (rbRed.isSelected()) {
        text.setFill(Color.RED);
    }
});

rbYellow.setOnAction(e -> {
    if (rbYellow.isSelected()) {
        text.setFill(Color.YELLOW);
    }
});
```

```
        rbBlack.setOnAction(e -> {
            if (rbBlack.isSelected()) {
                text.setFill(Color.BLACK);
            }
        });

        rbOrange.setOnAction(e -> {
            if (rbOrange.isSelected()) {
                text.setFill(Color.ORANGE);
            }
        });

        rbGreen.setOnAction(e -> {
            if (rbGreen.isSelected()) {
                text.setFill(Color.GREEN);
            }
        });


        Scene scene = new Scene(pane, 450, 150);
            primaryStage.setTitle("190130107118-Param_4B2-Practical-20");
            primaryStage.setScene(scene);
            primaryStage.show();
    }
}
```

Output :


190130107118-Param_4B2-Practical-20
Red
Yellow
Black
Orange
Green
Param Radadiya
<=
=>


190130107118-Param_4B2-Practical-20
Red
Yellow
Black
Orange
Green
Param Radadiya
<=
=>

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.

Code:

```
package sample;

import java.io.*;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.PrintWriter;
import java.time.LocalDateTime;

public class Practical21
{
   public static void main(String[] args)
   {

      LocalDateTime t1 = LocalDateTime.now();
      System.out.println(t1);
      System.out.println("190130107118 Param Radadiya 4CEB2");
      System.out.println("Practical - 21\n");
      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) + " ");
         }
         System.out.println("File is Created");
      }
```

```
        catch (FileNotFoundException fnfe)
        {
            System.out.println("Cannot create the file.");
            fnfe.printStackTrace();
        }
    }
}
```

Output :

```
2021-04-24T15:04:12.817616100
190130107118 Param Radadiya 4CEB2
Practical - 21

File is Created
```

File 123.txt :

```
122 113 101 83 72 138 114 98 73 149 67 43 35 67 29 131 107 61
 67 120 100 145 29 15 142 6 94 70 97 147 101 21 36 34 4 138 131
 129 22 117 38 147 23 61 58 116 7 19 117 126 71 114 57 104 98
14 80 145 18 101 16 42 147 9 137 63 38 77 113 138 3 127 133 149
118 125 106 23 50 72 54 139 35 43 66 33 125 59 93 69 4 95 139 78
21 9 99 9 83 1 65 31 6 27 83 102 111 108 10 36 121 148 26 70 131
17 44 74 68 28 131 124 8 63 74 135 22 93 144 91 76 77 23 93 54 119
12 138 80 4 40 118 87 104 132 135 139 48 20 42
```

Practical 22(A)

Write a recursive method that returns the smallest integer in an array.

Code:

```
import java.time.LocalDateTime;
import java.util.Scanner;

public class Practical22
{
   public static void main(String[] args)
   {
      LocalDateTime t1 = LocalDateTime.now();
      System.out.println(t1);
      System.out.println("190130107118 Param Radadiya 4CEB2");
      System.out.println("Practical - 22\n");

      Scanner input = new Scanner(System.in);

      System.out.print("Enter five integers: ");
      int[] list = new int[5];
      for (int i = 0; i < list.length; i++)
      {
         list[i] = input.nextInt();
      }

      System.out.println("The smallest element is " + min(list));
   }


   public static int min(int[] list)
   {
      int min = list[list.length - 1];
      int index = list.length - 1;
      return min(list, index, min);
   }
```

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

Output :

```
2021-04-24T15:10:14.970168400
190130107118 Param Radadiya 4CEB2
Practical - 22

Enter five integers: 2 3 6 9 80 9
The smallest element is 2
```

Practical 22(B)

Write a test program that prompts the user to enter an integer and display its product.

Code:

```
import java.time.LocalDateTime;
import java.util.Scanner;

public class Practical22
{
   public static void main(String[] args)
   {
      LocalDateTime t1 = LocalDateTime.now();
      System.out.println(t1);
      System.out.println("190130107118 Param Radadiya 4CEB2");
      System.out.println("Practical - 22\n");

            Scanner input = new Scanner(System.in);
            int product=1;
            System.out.print("Enter five integers: ");
            int[] list = new int[5];
            for (int i = 0; i < list.length; i++)
            {
               list[i] = input.nextInt();
               product *= list[i];
            }
            System.out.println("The Product of elements is " + product);
         }
      }
```

Output :

```
2021-04-24T15:18:18.789759300
190130107118 Param Radadiya 4CEB2
Practical - 22

Enter five integers: 1 9 2 60 7
The Product of elements is 7560
```

Practical 23

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

Code:

```
import java.time.LocalDateTime;

public class Practical23
{
   public static void main(String[] args)
   {
      LocalDateTime t1 = LocalDateTime.now();
      System.out.println(t1);
      System.out.println("190130107118 Param Radadiya 4CEB2");
      System.out.println("Practical - 23\n");

      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 = " + max(list));
   }

   public static <E extends Comparable<E>> E max(E[][] list)
   {
      E max = list[0][0];
```

```
        for (E[] elements : list)
        {
            for (E element : elements)
            {
                if (element.compareTo(max) > 0)
                {
                    max = element;
                }
            }
        }
        return max;
    }
}
```

Output :

```
2021-04-24T15:22:43.211953400
190130107118 Param Radadiya 4CEB2
Practical - 23

Min = 99
```

Practical 24

Define MYPriorityQueue class that extends Priority Queue to implement the Cloneable interface and implement the clone() method to clone a priority queue

Code:

```
import java.time.LocalDateTime;
import java.util.PriorityQueue;

public class Practical24
{
   public static void main(String[] args)
   {
      LocalDateTime t1 = LocalDateTime.now();
      System.out.println(t1);
      System.out.println("190130107118 Param Radadiya 4CEB2");
      System.out.println("Practical - 24\n");
      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 :

```
2021-04-24T15:25:37.789686400
190130107118 Param Radadiya 4CEB2
Practical - 24

[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 passedas a command-line argument.

Code:

```
import java.io.*;
import java.security.InvalidParameterException;
import java.time.LocalDateTime;
import java.util.Arrays;
import java.util.TreeSet;
import java.util.Iterator;

public class Practical25
{
   public static void main(String[] args) throws FileNotFoundException
   {
      LocalDateTime t1 = LocalDateTime.now();
      System.out.println(t1);
      System.out.println("190130107118 Param Radadiya 4CEB2");
      System.out.println("Practical - 25\n");

      if (args.length != 1)
         throw new InvalidParameterException("Usage: fullFilePathName");

      File file = new File(args[0]);

      if (!file.isFile())
         throw new FileNotFoundException(file + " is not a file.");

      try (BufferedReader in = new BufferedReader(new
InputStreamReader(new FileInputStream(file)), 10000))
      {
```

```
        String inputS;
        StringBuilder sb = new StringBuilder(10000);
        while ((inputS = in.readLine()) != null)
            sb.append(inputS);

        String[] words = sb.toString().split("\\s+");

        TreeSet<String> ndWords = new TreeSet<>(Arrays.asList(words));

        Iterator<String> itr = ndWords.descendingIterator();
        String s;
        while (itr.hasNext())
        {
            s = itr.next();
            System.out.println(s);
        }
    }
    catch (IOException e)
    {
        e.printStackTrace();
        System.exit(0);
    }

  }
}
```

Output :

```
2021-04-24T15:28:42.775464900
190130107118 Param Radadiya 4CEB2
Practical - 25

Exception in thread "main" java.security.InvalidParameterException Create breakpoint : Usage: fullFilePathName
    at Practical25.main(Practical25.java:20)

Process finished with exit code 1
```