

Name: HRITHIK M NAYAK

Usn:4NM20IS056

1)Sort the list of numbers given by the user using bubble sort.

```
Import java.util.*;

Class BubbleSort

{
    Void bubblesort(int arr[])
    {
        int n=arr.length();
        for(int i;i<n;i++)
        {
            for(j=0;j<n-1;j++)
            {
                if (arr[j]>arr[j+1])
                {
                    int temp=arr[j];
                    arr[j]=arr[j+1];
                    arr[j+1]=temp;
                }
            }
        }
        Void display(int arr[])
        {
            int n=arr.length();
            for(int i=0;i<n;i++)
            {
                System.out.print(arr[i] + " ")
            }
        }
    }

    Public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int n;
        System.out.println("Enter the number of elements")
    }
}
```

```
n=sc.nextInt();

int arr[] = new int[10];

System.out.println("Enter " + n + " integers");

for(int i=0; i < n; i++)

{

    arr[i] = sc.nextInt();

}

BubbleSort ob = new BubbleSort();

ob.bubblesort(arr);

System.out.println("Sorted array is");

ob.display(arr);

}

}
```

Output: Enter the number of elements

6

Enter 6 integers

31 8 16 36 13 27

Sorted array is

8 13 16 27 31 36

JAVA TASK-3

Name - Jeevith D R
USN - 4NM20IS057
Program - 2

Q. Write a Java program to find the factorial of a number.

```
import java.util.Scanner;

public class Example
{
    public static void main(String[] args)
    {
        int i,fact=1,number;

        Scanner s = new Scanner(System.in);

        System.out.print("Enter the number : ");
        number = s.nextInt();

        for(i=1;i<=number;i++)
        {
            fact = fact * i;
        }

        System.out.print("Factorial of the number is :" +fact);
    }
}
```

Output :

```
Enter the number : 5
Factorial of the number is :120
```

```
3.write a java program to print floyd's triangle.  
import java.util.*;  
class GFG {  
    public static void main(String[] args)  
    {  
        int n = 5;  
        int i, j, k = 1;  
        for (i = 1; i <= n; i++) {  
            for (j = 1; j <= i; j++) {  
                System.out.print(k + " ");  
                k++;  
            }  
            System.out.println();  
        }  
    }  
}
```

output:

```
1  
2 3  
4 5 6  
7 8 9 10  
11 12 13 14 15
```

Java Task

Name: Joyson Brian Dcunha

USN: 4NM20IS059

4. Write a Java program for linear search and binary search.

```
import java.util.Scanner;

public class Search {

    void linearSearch(int n, int[] a, int key) {
        int i=0;
        for(i=0; i<n; i++) {
            if(key==a[i]) {
                System.out.println("Key "+key+" found at position "
                    +(i+1));
                break;
            }
        }
        if(i==n) {
            System.out.println("Key "+key+" not found !!!");
        }
    }

    void binarySearch(int n, int[] a, int key) {
        int low=0, high=n, mid;
        while(low<=high) {
            mid=low+(high-low)/2;
            if(key==a[mid]) {
                System.out.println("Key "+key+" found at position "
                    +(mid+1));
                break;
            }
            else if(key<a[mid]) {
                high=mid-1;
            }
            else {
                low=mid+1;
            }
        }
        if(low>high) {
            System.out.println("Key "+key+" not found !!!");
        }
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        int arr[] = new int[20];
        System.out.print("Enter the size of the array: ");
        int n= sc.nextInt();
        System.out.println("Enter array elements in ascending order:");
        for(int i=0; i<n; i++) {
            arr[i]=sc.nextInt();
        }
        System.out.print("Enter the key to be searched: ");
        int key = sc.nextInt();
    }
}
```

```
        Search s=new Search();
        System.out.println("Searching using linear search algorithm:");
        s.linearSearch(n, arr, key);
        System.out.println("Searching using binary search algorithm:");
        s.binarySearch(n, arr, key);
    }

}
```

OUTPUT 1:

```
Enter the size of the array: 5
Enter array elements in ascending order:
3 57 68 90 99
Enter the key to be searched: 57
Searching using linear search algorithm:
Key 57 found at position 2
Searching using binary search algorithm:
Key 57 found at position 2
```

OUTPUT 2:

```
Enter the size of the array: 5
Enter array elements in ascending order:
1 2 3 4 5
Enter the key to be searched: 6
Searching using linear search algorithm:
Key 6 not found !!!
Searching using binary search algorithm:
Key 6 not found !!!
```

17) Transpose of a matrix

Name: Nishanth r alva

Usn: 4nm20is090

```
package transposemat;

import java.util.Scanner;

public class Transpose

{
    public static void main(String args[])
    {
        int i, j;
        System.out.println("Enter total rows and columns: ");
        Scanner s = new Scanner(System.in);
        int row = s.nextInt();
        int column = s.nextInt();
        int array[][] = new int[row][column];
        System.out.println("Enter matrix:");
        for(i=0;i<row;i++)
        {
            for(j=0;j<column;j++)
            {
                array[i][j] = s.nextInt();
                System.out.print(" ");
            }
        }
        System.out.println("The above matrix before Transpose is ");
        for(i=0;i<row;i++)
        {
            for(j=0;j<column;j++)
            {
                System.out.print(array[i][j]+" ");
            }
        }
        System.out.println(" ");
    }
    System.out.println("The above matrix after Transpose is ");
    for(i=0;i<column;i++)
    {
        for(j=0;j<row;j++)
        {
            System.out.print(array[j][i]+" ");
        }
        System.out.println(" ");
    }
}}
```

Problems @ Javadoc Declaration Console X
<terminated> Transpose [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (12-Jul-2023)
Enter total rows and columns:
4 4
Enter matrix:
1 2 3 4
4 5 6 7
6 7 5 4
9 8 7 2
The above matrix before Transpose is
1 2 3 4
4 5 6 7
6 7 5 4
9 8 7 2
The above matrix after Transpose is
1 4 6 9
2 5 7 8
3 6 5 7
4 7 4 2

4NM20IS061-SHRAVYA KAMATH

Q).Write a java program to get ip address and current date.

```
package shravya;

import java.net.InetAddress;

public class javaP
{
    public static void main(String args[]) throws Exception
    {
        InetAddress localhost = InetAddress.getLocalHost();
        System.out.println("system IP
Address:"+localhost.getHostAddress().trim());
        System.out.println("the current date
is:"+java.time.LocalDate.now());
    }
}
```

The screenshot shows the Eclipse IDE interface with the Java application 'javaP' running. The console tab displays the following output:

```
<terminated> javaP [Java Application] C:\Users\hp\p2\pool\plugins\org.eclipse.jst.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v202201-1208\jre\bin\javaw.exe (13-Jun-2022, 7524)
system IP Address:192.168.2.9
the current date is:2022-06-13
```

Usn:- 4nm20is092

7. Write a java program to perform garbage collection and to generate random numbers.

```
import java.util.Random;
public class TestGarbage1
{
    String ob_name;
    TestGarbage1(String ob_name)
    {
        this.ob_name=ob_name;
        System.out.println("object "+this.ob_name+" is created");
    }

    public void finalize()
    {
        System.out.println(this.ob_name+" object is garbage collected");
    }

    public static void main(String args[])
    {
        Random rand=new Random();
        System.out.println("Generating random numbers....");
        for(int i=0;i<=5;i++)
        {
            int item1=rand.nextInt(50);
            System.out.println("random no. is: "+item1);
        }
        TestGarbage1 s1=new TestGarbage1("t1");
        TestGarbage1 s2=new TestGarbage1("t2");
        s1=null;
        s2=null;
        System.gc();
    }
}
```

```
Generating random numbers....
random no. is: 29
random no. is: 41
random no. is: 28
random no. is: 26
random no. is: 42
random no. is: 29
object t1 is created
object t2 is created
t2 object is garbage collected
t1 object is garbage collected
```

4NM20IS063-KOYEL SUDANI

8). Write a java program to multiply two matrices.

```
import java.util.Scanner;

class MatrixMultiplication
{
    public static void main(String args[])
    {
        int m, n, p, q, sum = 0, c, d, k;

        Scanner in = new Scanner(System.in);
        System.out.println("Enter the number of rows and columns of first matrix");
        m = in.nextInt();
        n = in.nextInt();

        int first[][] = new int[m][n];

        System.out.println("Enter elements of first matrix");

        for (c = 0; c < m; c++)
            for (d = 0; d < n; d++)
                first[c][d] = in.nextInt();

        System.out.println("Enter the number of rows and columns of second matrix");
        p = in.nextInt();
        q = in.nextInt();

        if (n != p)
            System.out.println("The matrices can't be multiplied with each other.");
        else
        {
            int second[][] = new int[p][q];
            int multiply[][] = new int[m][q];

            System.out.println("Enter elements of second matrix");

            for (c = 0; c < p; c++)
                for (d = 0; d < q; d++)
                    second[c][d] = in.nextInt();

            for (c = 0; c < m; c++)
            {
                for (d = 0; d < q; d++)
                {
                    for (k = 0; k < p; k++)
                    {
                        sum = sum + first[c][k]*second[k][d];
                    }

                    multiply[c][d] = sum;
                    sum = 0;
                }
            }
        }
    }
}
```

```

        System.out.println("Product of the matrices:");

        for (c = 0; c < m; c++)
        {
            for (d = 0; d < q; d++)
                System.out.print(multiply[c][d]+"\t");

            System.out.print("\n");
        }
    }
}

```

The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

```

Problems Javadoc Declaration Call Hierarchy
<terminated> MatrixMultiplication [Java Application] C:\Users\hp\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (12-Jun-2022, 5:25:07 pm - 5:25:43 pm) [pid
Enter the number of rows and columns of first matrix
3 3
Enter elements of first matrix
1 2 3
4 5 6
7 8 9
Enter the number of rows and columns of second matrix
3 3
Enter elements of second matrix
9 8 7
6 5 4
3 2 1
Product of the matrices:
30      24      18
84      69      54
138     114     90

```

USN- 4NM20IS094

NAME- Peram Naga Ajay

PROGAM-9

The Fibonacci sequence is defined by the following rule. The first 2 values in the sequence are 1, 1. Every subsequent value is the sum of the 2 values preceding it. Write a Java program that uses both recursive and non-recursive functions to print the nth value of the Fibonacci sequence?

```
package mse_1;
```

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

```
class Fibonacci
```

```
{
```

```
    int fib(int n)
```

```
{
```

```
    if(n==1)
```

```
        return (1);
```

```
    else if(n==2)
```

```
        return (1);
```

```
    else
```

```
        return (fib(n-1)+fib(n-2));
```

```
}
```

```
}
```

```
public class Main
```

```
{  
public static void main(String args[ ])  
  
{  
Scanner input=new Scanner(System.in);  
int i,a=1,b=1,c=0,n;  
System.out.println("Recursive method");  
System.out.print("Enter the value of n: ");  
n=input.nextInt();  
System.out.print(a);  
System.out.print(" "+b);  
for(i=0;i<n-2;i++) {  
c=a+b;  
a=b;  
b=c;  
System.out.print(" "+c);  
}  
System.out.println();  
System.out.println(n+"th number of the series is: "+c);
```

```
System.out.println("Non recursive method");  
Scanner sc = new Scanner(System.in);  
System.out.print("Enter value of n1 : ");  
int n1=sc.nextInt();
```

2/3

```
Fibonacci ob=new Fibonacci();  
System.out.print("Fibonaccie Series is:");  
int res1=0;  
for(int i1=1;i1<=n1;i1++)
```

```

{
    res1 = ob.fib(i1);
    System.out.print(" "+res1);
}
System.out.println();

System.out.println(n1+"th number of the series is: "+res1);

}

}

```

OUTPUT

```

<terminated> Main [Java Application] C:\Users\ajayp\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\bin\java.exe
Recursive method
Enter the value of n: 10
1 1 2 3 5 8 13 21 34 55
10th number of the series is: 55
Non recursive method
Enter value of n1 : 12
Fibonacci Series is: 1 1 2 3 5 8 13 21 34 55 89 144
12th number of the series is: 144

```

PRAJNA PAI

4NM20IS095

10. Write a Java program that prompts the user to enter for an integer and then prints out all the Prime numbers up to that integer.

```
package mypackage;  
import java.util.Scanner;  
class PrimeNumber  
{  
    public static void main(String args[])  
    {  
        int n,p;  
        Scanner s=new Scanner(System.in);  
        System.out.print("Enter a number: ");  
        n=s.nextInt();  
        for(int i=2;i<n;i++)  
        {  
            p=0;  
            for(int j=2;j<i;j++)  
            {  
                if(i%j==0)  
                    p=1;  
            }  
            if(p==0)  
                System.out.print(i+"\t");  
        }  
        if(n<=1)  
        {  
            System.out.println("Sorry, Prime number doesn't exist!!");  
        }  
    }  
}
```

OUTPUT:

Case 1: When a number is less than or equal to 1

```
Problems @ Javadoc Declaration
<terminated> PrimeNumber [Java Application]
Enter a number: -10
Sorry, Prime number doesn't exist!!
```

Case 2: When a number is greater or equal to 2:

```
Problems @ Javadoc Declaration Console X
<terminated> PrimeNumber [Java Application] C:\Users\prajn\p2\poc\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_18.0.1
Enter a number: 45
2      3      5      7      11     13     17     19     23     29     31     37     41     43
```

4NM20IS096

PRAJNA SHETTY

11. Write a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

```
package java_task;

import java.util.Scanner;

class Assignment
{
    public static void main (String args[])
    {
        System.out.println("Enter the coefficients a,b,c of quadratic equation ax2 + bx
+ c = 0 and where a not equal to 0 ");
        Scanner sc = new Scanner(System.in);
        double a=sc.nextInt();
        if (a==0)
        {
            System.out.println("please enter the value of a except 0");
        }
        else
        {
```

```
double b=sc.nextInt();
double c=sc.nextInt();
double z=b*b-4*a*c;
Equation ob=new Equation();

if (z<0)
{
    System.out.println("In this there are no real solutions");
}

else if(z==0)
{
    System.out.println("The solutions are real and equal");
    ob.checking(a,b,c);
    ob.display();
}

else
{
    System.out.println("The solutions are real and distinct");
    ob.checking(a,b,c);
    ob.display();
}

}
```

```
class Equation
{
    double a;
    double b;
    double c;
    double x;
    double y;
    void checking(double a,double b,double c)
    {
        this.a=a;
        this.b=b;
        this.c=c;
        double z=Math.pow( b*b-4*a*c , 0.5 );
        x=(-b-z)/(2*a);
        y=(-b+z)/(2*a);
    }
    void display()
    {
        System.out.println("The value of x is"+x);
        System.out.println("The value of y is"+y);
    }
}
```

OUTPUT:

```
Enter the cofficients a,b,c of quadratic equation ax2 + bx + c = 0 and where a not equal to 0  
2 -8 3  
The solutions are real and distinct  
The value of x is0.41886116991581024  
The value of y is3.58113883008419
```

```
Enter the cofficients a,b,c of quadratic equation ax2 + bx + c = 0 and where a not equal to 0  
1 1 2  
In this there are no real solutions
```

147M of 257M

```
Enter the cofficients a,b,c of quadratic equation ax2 + bx + c = 0 and where a not equal to 0  
4 -4 1  
The solutions are real and equal  
The value of x is0.5  
The value of y is0.5
```

Enter the coefficients a,b,c of quadratic equation $ax^2 + bx + c = 0$ and where a not equal to 0

0 1 2

please enter the value of a except 0

12. Write a java program that reads on file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of the file and the length of the file in bytes ?

```
package javapgm;
import java.io.*;
import java.util.*;
class AboutFile
{
public static void main(String args[])
{
Scanner input=new Scanner(System.in);
System.out.print("Enter the name of the file: ");
String file_name=input.nextLine();
File f=new File(file_name);
if(f.exists())
System.out.println("The file "+file_name+" exists");
else
System.out.println("The file "+file_name+" does not exists");
System.out.println("Absolute path : "+f.getAbsolutePath());
if(f.canRead())
System.out.println("The file "+file_name+" is readable");
else
System.out.println("The file "+file_name+" is not readable");
if(f.canWrite())
System.out.println("The file "+file_name+" is writable");
else
System.out.println("The file "+file_name+" is not writable");
if(f.isFile())
System.out.println(""+file_name+" is a file");
else
System.out.println(""+file_name+" is not a file");
if(f.isDirectory())
System.out.println(""+file_name+" is a directory");
else
System.out.println(""+file_name+" is not a directory");
System.out.println("The length of the file: "+f.length()+" bytes");
}
}
```

OUTPUT 1:

```
Enter the name of the file: input.txt
The file input.txt exists
Absolute path : C:\Users\User\java\task3\input.txt
The file input.txt is readable
The file input.txt is writable
input.txt is a file
input.txt is not a directory
The length of the file: 8 bytes
```

OUTPUT 2:

```
Enter the name of the file: abc.txt
The file abc.txt does not exists
Absolute path : C:\Users\User\java\task3\abc.txt
The file abc.txt is not readable
The file abc.txt is not writable
abc.txt is not a file
abc.txt is not a directory
The length of the file: 0 bytes
```

JAVA TASK 3
PRAMITH BHANDARY D
4NM20IS098

Q-13) Java program to demonstrate Producer-Consumer Problem

```
import java.util.LinkedList;
public class Pgml
{
    public static void main(String[] args) throws InterruptedException
    {
        final PC pc = new PC();
        Thread t1 = new Thread(new Runnable()
        {
            public void run()
            {
                try
                {
                    pc.produce();
                }
                catch (InterruptedException e)
                {
                    e.printStackTrace();
                }
            }
        });
        Thread t2 = new Thread(new Runnable()
        {
            public void run()
            {
                try
                {
                    pc.consume();
                }
                catch (InterruptedException e)
                {
                    e.printStackTrace();
                }
            }
        });
        t1.start();
        t2.start();

        t1.join();
        t2.join();
    }
    public static class PC
    {

        LinkedList<Integer> list = new LinkedList<>();
        int capacity = 2;
        void produce() throws InterruptedException
        {
            int value = 0;
            while (true)
            {
                synchronized (this)
```

JAVA TASK 3
PRAMITH BHANDARY D
4NM20IS098

```
        {
            while (list.size() == capacity)
                wait();
            System.out.println("Producer produced-"+value);
            list.add(value++);
            notify();
            Thread.sleep(1000);
        }
    }
void consume() throws InterruptedException
{
    while (true)
    {
        synchronized (this)
        {
            while (list.size() == 0)
                wait();
            int val = list.removeFirst();
            System.out.println("Consumer consumed- " + val);
            notify();
            Thread.sleep(1000);
        }
    }
}
}
```

OUTPUT:

```
Producer produced- 0
Producer produced- 1
Consumer consumed- 0
Consumer consumed- 1
Producer produced- 2
Producer produced- 3
Consumer consumed- 2
Producer produced- 4
Consumer consumed- 3
Consumer consumed- 4
Producer produced- 5
Producer produced- 6
Consumer consumed- 5
Consumer consumed- 6
Producer produced- 7
Producer produced- 8
Consumer consumed- 7
```

14. Define a class IntArr which hosts array of integers. Provide the following methods:

1. A **default constructor**.
2. A **parameterized constructor**.
3. A method called **display** to display the array contents.
4. A method called **search** to search for an element in the array.
5. A method called **compare** which compares 2 IntArr objects for equality.

```
package pck1;
import java.util.Scanner;

public class IntArr
{
    int a[];
    IntArr()
    {
        System.out.println("Array Demonstration");
    }
    IntArr(int arr[])
    {
        a=arr;
    }
    void display()
    {
        System.out.println("Array contents:");
        for(int i=0;i<a.length;i++)
            System.out.print(a[i]+"\t");
        System.out.println();
    }
    void search(int k)
    {
        int flag=-1;
        for(int i=0;i<a.length;i++)
```

```
        {
            if(a[i]==k)
            {
                flag=i;
                break;
            }
        }
        if(flag==-1)
        {
            System.out.println(k+" is not found.");
        }
        else
        {
            System.out.println(k+" is found at index
                                "+flag);
        }
    }
    boolean compare(int a[],int b[])
    {
        for(int i=0;i<a.length;i++)
        {
            if(a[i]!=b[i])
            {
                return false;
            }
        }
        return true;
    }
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int arr1[] = {10,20,30,40,45};
        IntArr ob=new IntArr();
        IntArr ob1=new IntArr(arr1);
        System.out.println("First array:");
        ob1.display();
        System.out.println("Enter the element to search:");
        int k=sc.nextInt();
        ob1.search(k);
```

```
    int arr2[] = {50,60,45,80,70};  
    IntArr ob2=new IntArr(arr2);  
    System.out.println("Second array:");  
    ob2.display();  
    System.out.println("Two arrays are equal:  
                      "+ob2.compare(arr1,arr2));  
    sc.close();  
}  
}
```

Output:

```
Array Demonstration  
First array:  
Array contents:  
10  20  30  45  60  
Enter the element to search:  
45  
45 is found at index 3  
Second array:  
Array contents:  
30  55  60  90  80  
Two arrays are equal: false
```

15. Define a class called Customer that holds private fields for a customer ID number, name, and credit limit. Include appropriate constructors to initialize the instance variables of the Customer class. Write a main() method that declares an array of 5 customer objects. Prompt the user for values for each Customer and display all 5 Customer objects.

```
import java.util.*;
public class Customer
{
    String ID;
    String name;
    int Climit;
    Customer (String ID,String name,int Climit)
    {
        this.ID=ID;
        this.name=name;
        this.Climit=Climit;
    }
    public static void main(String a[])
    {
        Scanner s1=new Scanner(System.in);
        Customer s[]=new Customer [5];
        String ID;
        String name;
        int Climit;
        int n;
        System.out.println("Enter the number of Customer ");
        n=s1.nextInt();
        for(int i=0;i<n;i++)
        {
            System.out.println("Enter Customer Details: ID NAME credit
limit ");
            ID=s1.next();
            name=s1.next();
            Climit=s1.nextInt();
            s[i]=new Customer (ID,name,Climit);
        }
        for(int i=0;i<n;i++)
        {
            System.out.println("Customer "+(i+1)+" Details: "+s[i].ID+" "+s[i].name+" "+s[i].Climit);
        }
    }
}
```

```

name=s1.next();

Climit=s1.nextInt();

s[i]=new Customer(ID,name,Climit);

}

System.out.println("Customer Details");

System.out.println("ID\tNAME\tCLIMIT ");

for(int i=0;i<n;i++)

{

System.out.println(s[i].ID+"\t"+s[i].name+"\t"+s[i].Climit);

}

}

}

```

The screenshot shows the Eclipse IDE interface with the following details:

- Package Explorer View:** Shows the project structure under the package A. It includes a JRE System Library [JavaSE-17], a src folder containing a Customer.java file, and other files like aj, fib, fib1, K, mse, n, new, module-info.java, and Task-3.
- Customer.java Content:** The code defines a Customer class with ID, name, and Climit fields, and a constructor that initializes them.
- Console Output:**

```

Enter the number of Customer
2
Enter Customer Details: ID NAME creditlimit
1234
LESTON
123
Enter Customer Details: ID NAME creditlimit
2345
AJAY
245
Customer Details
ID      NAME    CLIMIT
1234    LESTON   123
2345    AJAY     245

```

Leston Mathias

4nm20is070

4NM20IS101

PRARTHANA M KUNDER

16. Create a **Person** class with private instance variables for the person's name and birth date. Add appropriate accessor methods for these variables. Then create a subclass **College Graduate** with private instance variables for the student's GPA and year of graduation and appropriate accessors for these variables. Include appropriate constructors or your class. Then create a class with **main()** method that demonstrates your classes.

```
package task1;
import java.util.Scanner;
class Person
{
    Person(String name,int dob)
    {
        this.name=name;
        this.dob=dob;
    }
    private String name;
    private int dob;
    public String getname()
    {
        return name;
    }
    public int getdob()
    {
        return dob;
    }
}
class CollegeGraduate extends Person
{
    private float GPA;
    private int year;
    CollegeGraduate(String name,int dob,float GPA,int year)
    {
        super(name,dob);
        this.GPA=GPA;
        this.year=year;
    }
    public float getgpa()
    {
        return GPA;
    }
    public int getyear()
    {
        return year;
    }
}
public class Demo
{
    public static void main(String args[])
    {
        String n;
        int dob,year;
```

```
float gpa;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the name:");
n=sc.next();
System.out.println("Enter the date of birth(ddmm/yyyy):");
dob=sc.nextInt();
System.out.println("Enter the GPA:");
gpa=sc.nextFloat();
System.out.println("Enter the year of graduation");
year=sc.nextInt();
CollegeGraduate c=new CollegeGraduate(n,dob,gpa,year);
n=c.getname();
dob=c.getdob();
gpa=c.getgpa();
year=c.getyear();
System.out.println("Details Entered\nName : "+n+"\nDate of
birth"+dob+"\nGPA : "+gpa+"\nYear of Graduation : "+year);
}
}
```

OUTPUT:

```
Enter the name:
John
Enter the date of birth(ddmm/yyyy):
10082002
Enter the GPA:
8.98
Enter the year of graduation
2024
Details Entered
Name : John
Date of birth : 10082002
GPA : 8.98
Year of Graduation : 2024
```

4NM20IS072

M S YASHASWINI

17) Create a Building class and two subclasses, House and School. The Building class contains fields for square footage and stories. The House class contains additional fields for number of bedrooms and baths. The School class contains additional fields for number of classrooms and grade level(for example, elementary or junior high). All the class contain appropriate get and set methods. Place the Building , House and School classes in a package named com.course.structure. Create a main method that declares objects of each type and uses the package.

```
package com.course.structure;

class Building
{
    double sq;
    int stories;
    void get(double sq,int stories)
    {
        this.sq=sq;
        this.stories=stories;
    }
    void set()
    {
        System.out.println("Square footage= "+sq);
        System.out.println("Number of stories= "+stories);
    }
}
class House extends Building
{
    int bedroom,baths;
    void get(double sq,int stories,int bedroom,int baths)
    {
        super.get(sq,stories);
        this.bedroom=bedroom;
        this.baths=baths;
    }
    void set()
    {
        System.out.println("No. of bedrooms= "+bedroom);
        System.out.println("No. of bathroom ="+baths);
    }
}
class School extends Building
{
    int classroom;
    String grade;
    void get(double sq,int stories,String grade,int classroom)
    {
        super.get(sq,stories);
        this.grade=grade;
        this.classroom=classroom;
    }
    void set()
    {
        System.out.println("No. of classroom= "+classroom);
    }
}
```

```
        System.out.println("grade =" + grade);
    }
}

class Demo
{
    public static void main(String args[])
    {
        Building b=new Building();
        House h=new House();
        School s=new School();
        s.get(23.6,3,"A",5);
        s.set();
        h.get(23.6,3,2,2);
        h.set();
        b.get(23.6,3);
        b.set();
    }
}
```

OUTPUT

No. of classroom= 5

grade =A

No. of bedrooms= 2

No. of bathroom =2

Square footage= 23.6

Number of stories= 3

NAME-M SHASHANK DEV PADAKANNAYA

USN-4NM20IS073

18. Define a class CurrentDate with data members day, month and year. Define a method createDate() to create date object by reading values from keyboard. Throw a user defined exception by the name InvalidDayException if the day is invalid and InvalidMonthException if month is found invalid and display current date if the date is valid. Write a test program to illustrate the functionality.

```
package proj1;

import java.util.*;
import java.io.*;

class InvalidDayException extends Exception

{

    InvalidDayException()

    {

        super("Day is invalid");

    }

}

class InvalidMonthException extends Exception

{

    InvalidMonthException()

    {

        super("Month is invalid");

    }

}

public class CurrentDate

{

    void createDate()

    {
```

```
int arr[]= {31,28,31,30,31,30,31,31,30,31,30,31};

int x;

int y;

int z;

Scanner sc=new Scanner(System.in);

System.out.println("Enter the day");

x=sc.nextInt();

System.out.println("Enter the month");

y=sc.nextInt();

System.out.println("Enter the year");

z=sc.nextInt();

try

{

    if(y>12)

    {

        throw new InvalidMonthException();

    }

    else

    {

        try

        {

            if(x>arr[y-1])

            {

                throw new InvalidDayException();

            }

            else

            {

                System.out.println("Date entered is valid");
            }
        }
    }
}
```

```

        Date date=new Date();

        System.out.println(date);

    }

}

catch(InvalidDayException e)

{

    System.out.println(e);

}

}

catch(InvalidMonthException e)

{

    System.out.println(e);

}

}

public static void main(String args[]) throws Exception

{
    CurrentDate cur=new CurrentDate();

    cur.createDate();

}
}

```

Output

1)

Enter the day

22

Enter the month

44

Enter the year

2002

proj1.InvalidMonthException: Month is invalid

2)

Enter the day

33

Enter the month

11

Enter the year

2020

proj1.InvalidDayException: Day is invalid

3)

Enter the day

29

Enter the month

2

Enter the year

2002

proj1.InvalidDayException: Day is invalid

(above output is for february month)

4)

Enter the day

12

Enter the month

5

Enter the year

2021

DATE entered is valid

Sun Jun 12 17:56:33 IST 2022

19) Design a class Student with the methods, get Number and put Number to read and display the Roll No. of each student and getMarks() and putMarks() to read and display their marks. Create an interface called Sports with a method put Grade() that will display the grade obtained by a student in Sports. Design a class called Result that will implement the method put Grade() and generate the final result based on the grade in sports and the marks obtained from the superclass Student.

```
import java.util.*;  
  
class Student  
{  
    protected static int roll_num;  
    Scanner s = new Scanner(System.in);  
    public void getNumber()  
    {  
        System.out.println("Enter Roll no. : ");  
        roll_num = s.nextInt();  
    }  
    public void putNumber()  
    {  
        System.out.println(roll_num);  
    }  
}  
  
class Test extends Student  
{  
    protected static int marks1;  
    protected static int marks2;  
    protected static int marks3;  
  
    public void getMarks()  
    {  
        super.getNumber();  
        System.out.println("Enter marks of subject 1");  
        marks1=s.nextInt();  
        System.out.println("Enter marks of subject 2");  
        marks2=s.nextInt();  
        System.out.println("Enter marks of subject 3");  
        marks3=s.nextInt();  
    }  
    public void putMarks()  
    {  
        System.out.println("Student Details:-\n ");  
        System.out.println("Rollnumber:-");  
        super.putNumber();  
        System.out.println("Subject 1 marks: "+marks1);  
        System.out.println("subject 2 marks: "+marks2);  
        System.out.println("Subject 3 marks: "+marks3);  
    }  
}
```

```
interface Sports
{
    public static final int s1 =40 ;
    public void putGrade();
}

class Result extends Test implements Sports
{
    public void putGrade()
    {
        System.out.println("The grade obtained by the student in sports
is "+Sports.s1);
    }

    public void calculate()
    {
        int tot_marks;
        tot_marks=marks1+marks2+marks3+Sports.s1;
        System.out.println("Total marks obtained by roll number "+
Roll_num+ ":"+tot_marks);
        System.out.println("\n\n");
    }
}

public class College
{
    public static void main(String args[])
    {
        Scanner s1 = new Scanner(System.in);
        Result r = new Result();
        int n;
        System.out.println("Enter number of student ");
        n=s1.nextInt();
        for(int i=0;i<n;i++)
        {
            r.getMarks();
            r.putMarks();
            r.putGrade();
            r.calculate();
        }
    }
}
```

```
Problems @ Javadoc Declaration Console X
<terminated> College [Java Application] C:\Users\Maaz\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64
Enter number of student:-
3
Enter Roll no:-
01
Enter marks of subject 1
45
Enter marks of subject 2
47
Enter marks of subject 3
44
Student Details:-

Rollnumber:-
1
Subject 1 marks: 45
subject 2 marks: 47
Subject 3 marks: 44
The grade obtained by the student in sports :- 40
Total marks Obtained by roll number 1:- 176

Enter Roll no:-
02
Enter marks of subject 1
40
Enter marks of subject 2
44
Enter marks of subject 3
39
Student Details:-

Rollnumber:-
2
Subject 1 marks: 40
subject 2 marks: 44
Subject 3 marks: 39
4
```

```
Subject 1 marks: 40
subject 2 marks: 44
Subject 3 marks: 39
The grade obtained by the student in sports :- 40
Total marks Obtained by roll number 2:- 163
```

```
Enter Roll no:-
3
Enter marks of subject 1
39
Enter marks of subject 2
37
Enter marks of subject 3
49
Student Details:-

Rollnumber:-
3
Subject 1 marks: 39
subject 2 marks: 37
Subject 3 marks: 49
The grade obtained by the student in sports :- 40
Total marks Obtained by roll number 3:- 165
```

NAME:MAHIMA
USN:4NM20IS075

20.create a class by extending thread class to print a multiplication table of a number supplied as parameter. create another class table which will instantiate two objects of the above class to print multiplication table of 5 and 7.

```
class Use
{
    void printUse(int n)
    {
        synchronized(this)
        {
            for(int i=1;i<=10;i++)
            {
                System.out.println(+n+"*"+i+"="++(n*i));
            }
            try
            {
                Thread.sleep(400);
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
        }
    }
}

class Mythread1 extends Thread
{
    Use t;
    Mythread1(Use t)
    {
```

```
this.t=t;  
}  
  
public void run()  
{  
    t.printUse(5);  
}  
}  
  
class Mythread2 extends Thread  
{  
    Use t;  
    Mythread2(Use t)  
    {  
        this.t=t;  
    }  
  
    public void run()  
    {  
        t.printUse(7);  
    }  
}  
  
public class Tables {  
  
    public static void main(String[] args) {  
        Use obj=new Use();  
        Mythread1 th1=new Mythread1(obj);  
        Mythread2 th2=new Mythread2(obj);  
        th1.start();  
        th2.start();  
    }  
}
```



```
}
```

```
32    }
33 }
```

```
Problems @ Javadoc Declaration Console X
<terminated> Tables [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_18.0.1
5*1=5
5*2=10
5*3=15
5*4=20
5*5=25
5*6=30
5*7=35
5*8=40
5*9=45
5*10=50
7*1=7
7*2=14
7*3=21
7*4=28
7*5=35
7*6=42
7*7=49
7*8=56
7*9=63
7*10=70
```

PRUTHVI R K

4NM20IS106

21. Write and execute a java program to create and initialize a matrix of integers. Create n threads(by implementing Runnable interface) where n is equal to the number of rows in the matrix. Each of these threads should compute a distinct row sum. The main thread computes the complete sum by looking into the partial sums given by the threads.

Program:

```
package Matrix1;

public class SumofRowColumn {

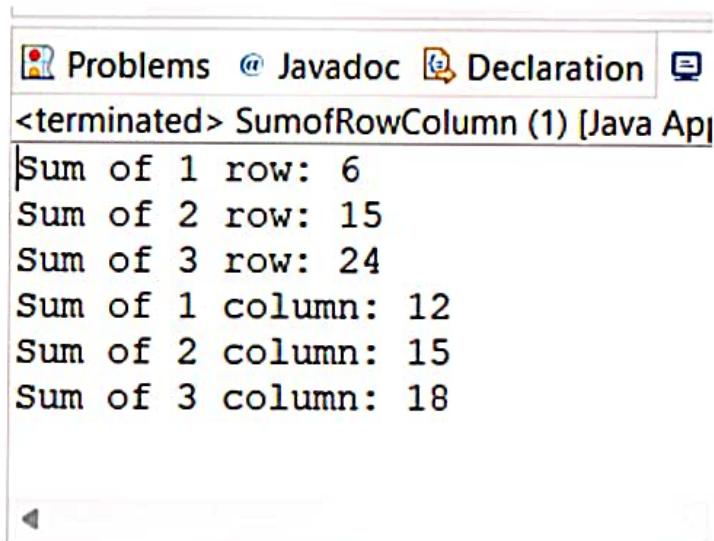
    public static void main(String[] args) {
        int rows, cols, sumRow, sumCol;
        int a[][] = {
            {1, 2, 3},
            {4, 5, 6},
            {7, 8, 9}
        };

        rows = a.length;
        cols = a[0].length;

        for(int i = 0; i < rows; i++){
            sumRow = 0;
            for(int j = 0; j < cols; j++){
                sumRow = sumRow + a[i][j];
            }
            System.out.println("Sum of " + (i+1) +" row: " + sumRow);
        }
    }
}
```

```
for(int i = 0; i < cols; i++){
    sumCol = 0;
    for(int j = 0; j < rows; j++){
        sumCol = sumCol + a[j][i];
    }
    System.out.println("Sum of " + (i+1) + " column: " + sumCol);
}
}
}
```

Output:



The screenshot shows an IDE interface with tabs for Problems, Javadoc, Declaration, and another tab that is partially visible. Below the tabs, the title bar reads "<terminated> SumofRowColumn (1) [Java Ap]". The main pane displays the following output text:

```
Sum of 1 row: 6
Sum of 2 row: 15
Sum of 3 row: 24
Sum of 1 column: 12
Sum of 2 column: 15
Sum of 3 column: 18
```



22.Create a Counter class with a private count instance variable and two methods.

The first method: synchronized void increment() tries to increment count by 1.

If count is already at its maximum of 3, then it waits until count is less than 3 before incrementing

it. The other method: synchronized void decrement() tries to decrement count by 1.

If count is already at its minimum of 0, then it waits until count is greater than 0

before decrementing it. Every time either method has to wait.

it displays a statement saying why it is waiting. Also, every time an increment or decrement occurs, the counter displays a statement that says what occurred and shows count's new value.

A.Create one thread class whose run() method calls the Counter's increment() method 20 times.

In between each call it sleeps for a random amount of time between 0 and 500 milliseconds.

B. Create one thread class whose run() method calls the Counter's decrement() method 20 times.

In between each call, it sleeps for a random amount of time between 0 and 500 milliseconds.

C. Write a Counter User class with a main() method that creates one Counter and the two threads and starts the threads running

Program:

```
package counteruser;
import java.util.Random;
class Counter
{
    private int count;
    synchronized void increment()
    {
        try
        {
            if(count==3)
            {
                System.out.println("Count=3...Wait for
Decrement");
                wait();
            }
        }
    }
}
```

```
        else
        {
            count++;
            System.out.println("Incremented
Count....="+count);
            notify();
        }
    }
    catch(Exception e)
    {

    }
}
synchronized void decrement()
{
    try
    {
        if(count==0)
        {
            System.out.println("Count=0...Wait for
Increment");
            wait();
        }
        else
        {
            count--;
            System.out.println("Decrement
Count.....=" + count);
            notify();
        }
    }
    catch(Exception e)
    {

    }
}
class Thread1 extends Thread
{
    Counter c;
    Thread1(Counter c)
```

```
    {
        this.c=c;
    }
    Random r=new Random();
    public void run()
    {
        for(int i=0;i<20;i++)
        {
            try
            {
                c.increment();
                int ran1=r.nextInt(500);
                Thread.sleep(ran1);
            }
            catch(InterruptedException e)
            {
                System.out.println("Interrupted");
            }
        }
    }
}
class Thread2 implements Runnable
{
    Counter c;
    Thread2(Counter c)
    {
        this.c=c;
    }
    Random r=new Random();
    public void run()
    {
        for(int i=0;i<20;i++)
        {
            try
            {
                c.decrement();
                int ran2=r.nextInt(500);
                Thread.sleep(ran2);
            }
            catch(InterruptedException e)
```

```
        {
            System.out.println("Interrupted");
        }

    }

}

public class CounterUser
{
    public static void main(String [] args)
    {
        Counter c=new Counter();
        Thread1 a=new Thread1(c);
        Thread2 b=new Thread2(c);
        Thread t1=new Thread(a);
        Thread t2=new Thread(b);
        t1.start();
        t2.start();
    }
}
```

Output:

```
Count=0...Wait for Increment
Incremented Count....=1
Decrementated Count.....=0
Incremented Count....=1
Decrementated Count.....=0
Incremented Count....=1
Incremented Count....=2
Decrementated Count.....=1
Incremented Count....=2
Decrementated Count.....=1
Decrementated Count.....=0
Incremented Count....=1
Decrementated Count.....=0
Count=0...Wait for Increment
Incremented Count....=1
Decrementated Count.....=0
Count=0...Wait for Increment
Incremented Count....=1
Incremented Count....=2
Incremented Count....=3
Decrementated Count.....=2
Incremented Count....=3
Count=3...Wait for Decrement
Decrementated Count.....=2
Incremented Count....=3
Decrementated Count.....=2
Decrementated Count.....=1
Decrementated Count.....=0
Incremented Count....=1
Incremented Count....=2
Incremented Count....=3
Decrementated Count.....=2
```

4NM20IS078
MANISH

23) Define a simple generic stack class and show the use of the generic class for two different class types student and employee class object

```
class Test<S, E>
{
    S obj1;
    E obj2;
    Test(S obj1, E obj2)
    {
        this.obj1 = obj1;
        this.obj2 = obj2;
    }
    public void print()
    {
        System.out.println(obj1);
        System.out.println(obj2);
    }
}
public class Main
{
    public static void main (String[] args)
    {
        Test <String, String> obj =
            new Test<String, String>("STUDENT1", "Employee 1");

        obj.print();
    }
}
```

OUTPUT

Result

CPU Time: 0.08 sec(s), Memory: 31700 kilobyte(s)

STUDENT 1
Employee 1

NAME – MANOJ B

USN – 4NM20IS079

```
public class Node<T>
{
    public T data;
    public Node<T> next;
    public Node(T data)
    {
        this.data = data;
    }
    public void displayNode()
    {
        System.out.print( data + " ");
    }
}
class LinkedList<T>
{
    private Node<T> first;
    public LinkedList()
    {
        first = null;
    }
    public void insertFirst(T data)
    {
        Node<T> newNode = new Node<T>(data);
        newNode.next = first;
        first = newNode;
    }
    public Node<T> deleteFirst()
    {
        if(first==null)
        {
            System.out.println("empty");
        }
        Node<T> tempNode = first;
        first = first.next;
        return tempNode;
    }
    public void displayLinkedList()
    {
        System.out.print("Displaying LinkedList [first--->last]: ");
        Node<T> tempDisplay = first;
        while (tempDisplay != null)
        {
            tempDisplay.displayNode();
            tempDisplay = tempDisplay.next;
        }
        System.out.println();
    }
}
class Generic
{
    public static void main(String[] args)
    {
        LinkedList<Integer> linkedList = new LinkedList<Integer>();
        linkedList.insertFirst(11);
```

```

linkedList.insertFirst(21);
linkedList.insertFirst(59);
linkedList.insertFirst(14);
linkedList.insertFirst(39);
linkedList.displayLinkedList();
System.out.print("Deleted Nodes: ");
Node<Integer> deletedNode = linkedList.deleteFirst();
deletedNode.displayNode();
deletedNode = linkedList.deleteFirst();
deletedNode.displayNode();
System.out.println();
linkedList.displayLinkedList();
LinkedList<Double> linkedList1 = new LinkedList<Double>();
linkedList1.insertFirst(11.9);
linkedList1.insertFirst(21.8);
linkedList1.insertFirst(59.1);
linkedList1.insertFirst(14.5);
linkedList1.insertFirst(39.6);
linkedList1.displayLinkedList();
System.out.print("Deleted Nodes: ");
Node<Double> deletedNode1 = linkedList1.deleteFirst();
deletedNode1.displayNode();
deletedNode1 = linkedList1.deleteFirst();
deletedNode1.displayNode();
System.out.println();
linkedList1.displayLinkedList();
}
}

```

OUTPUT :-

```

Problems Javadoc Declaration Console Coverage
<terminated> Generic Java Application C:\Users\HP\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v202201-1208\jre\bin\java.exe (12-Jun-2022, 7:14:52 pm - 7:14:54 pm) [pid: 12]
Displaying Linkedlist [first--->last]: 39 14 59 21 11
Deleted Nodes: 39 14
Displaying Linkedlist [first--->last]: 59 21 11
Displaying Linkedlist [first--->last]: 39.6 14.5 59.1 21.8 11.9
Deleted Nodes: 39.6 14.5
Displaying Linkedlist [first--->last]: 59.1 21.8 11.9

```

```

Displaying LinkedList [first--->last]: 39 14 59 21 11
Deleted Nodes: 39 14
Displaying LinkedList [first--->last]: 59 21 11
Displaying LinkedList [first--->last]: 39.6 14.5 59.1 21.8 11.9
Deleted Nodes: 39.6 14.5
Displaying LinkedList [first--->last]: 59.1 21.8 11.9

```

JAVA TASK – 3

Name: Rahul O G

Usn: 4NM20IS110

25). Write a program to demonstrate the use of wildcard arguments.

i) Wildcard arguments with unknown type:

```
import java.util.Arrays;
import java.util.List;
class unboundedwildcardemo
{
    public static void main(String[] args)
    {
        List<Integer> l1 = Arrays.asList(77, 88, 99);
        List<Double> l2 = Arrays.asList(77.7, 88.8, 99.9);
        printlist(l1);
        printlist(l2);
    }
    private static void printlist(List<?> list)
    {
        System.out.println(list);
    }
}
```

i)Output:

```
[77, 88, 99]  
[77.7, 88.8, 99.9]
```

ii) Wildcard arguments with upper bound:

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

```
import java.util.List;
```

```
class WildcardDemo  
{  
    public static void main(String[] args)  
    {  
        List<Integer> l1 = Arrays.asList(10, 20, 30, 40);  
        System.out.println("Total sum is:" + sum(l1));  
        List<Double> l2 = Arrays.asList(10.2, 20.3, 30.4, 40.5);  
        System.out.print("Total sum is:" + sum(l2));  
    }  
    private static double sum(List<? extends Number> list)  
    {  
        double sum = 0.0;  
        for (Number i : list)  
        {  
            sum += i.doubleValue();  
        }  
        return sum;  
    }  
}
```

ii)Output:

```
Total sum is:100.0  
Total sum is:101.4
```

iii) Wildcard arguments with lower bound:

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

```
import java.util.List;
```

```
class WildcardDemo {  
    public static void main(String[] args)  
    {  
        List<Integer> l1 = Arrays.asList(100, 200, 300, 400);  
        printOnlyIntegerClassorSuperClass(l1);  
        List<Number> l2 = Arrays.asList(59, 69, 79, 89);  
        printOnlyIntegerClassorSuperClass(l2);  
    }  
    public static void printOnlyIntegerClassorSuperClass(  
        List<? super Integer> list)  
    {  
        System.out.println(list);  
    }  
}
```

iii)Output:

```
[100, 200, 300, 400]  
[59, 69, 79, 89]
```

Name:Mithilesh Suresh Moger

Usn:4NM20IS081

26)Write a java program to display the listing of given directories and subdirectories using recursion.

```
package com.company;

import java.io.File;

public class Main {

    public static void main(String[] args) {

        File parentDirectory=new File("C:\\\\Users\\\\Mithilesh\\\\Downloads\\\\birthday-wisher-extrahard-
start");

        listdir(parentDirectory);

    }

    public static void listdir(File dir){

        File elements[]=dir.listFiles();

        for(File element:elements){

            if(element.isFile()){

                System.out.println(element);

            }

            else if(element.isDirectory()){

                listdir(element.getAbsoluteFile());

            }

        }

    }

}
```

Output:

The screenshot shows the IntelliJ IDEA interface with a terminal window open. The terminal is running a command to list files in a directory. The output of the command is displayed in the terminal window.

```
"C:\Program Files\Java\jdk-18\bin\java.exe" "-javaagent:C:\Users\Mithilesh\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2021.3.3\lib\idea_rt.jar" -Dfile.encoding=UTF-8 C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\Main.java
```

The terminal output lists the following files:

- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\.idea\gitignore
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\.idea\birthday-wisher-extrahard-start.iml
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\.idea\inspectionProfiles\profiles_settings.xml
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\.idea\misc.xml
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\.idea\modules.xml
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\.idea\workspace.xml
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\birthdays.csv
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\college_email_list
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\email-creator.py
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\letter_templates\letter_1.txt
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\letter_templates\letter_2.txt
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\letter_templates\letter_3.txt
- C:\Users\Mithilesh\Downloads\birthday-wisher-extrahard-start\main.py

At the bottom of the terminal output, it says "Process finished with exit code 0".

Below the terminal, the IntelliJ IDEA navigation bar is visible, showing tabs for Version Control, Run, TODO, Problems, Terminal, and Build. The Microsoft Edge browser tab is also open. The system tray at the bottom right shows the date and time as 6/12/2022 6:33 PM, along with other system icons.

27. Write a JavaFX application program to do the following:

- a. Display the message "Welcome to JavaFX programming" using Label in the scene.
- b. Set the text color of the Label to Magenta.
- c. Set the title of the Stage to "This is the first JavaFX Application".
- d. Set the width and height of the Scene to 500 and 200 respectively.
- e. Use FlowPane layout and set the hgap and vgap of the FlowPane to desired values.

```
import javafx.application.Application;
import javafx.scene.*;
import javafx.scene.control.Label;
import javafx.scene.layout.FlowPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

public class Main extends Application {
    public static void main(String[] args) {
        launch(args);
    }

    public void start(Stage s) {
        s.setTitle("This is the first JavaFX Application.");

        FlowPane root = new FlowPane(10,10);

        Scene myScene = new Scene(root,500,200);

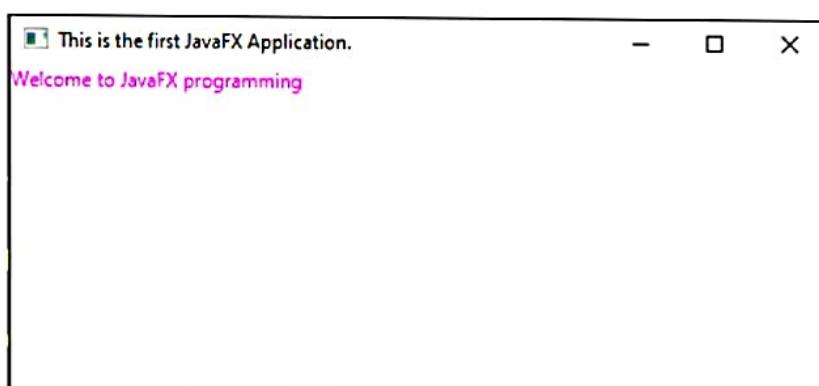
        s.setScene(myScene);

        Label l1 = new Label("Welcome to JavaFX programming");

        l1.setTextFill(Color.MAGENTA);

        root.getChildren().add(l1);

        s.show();
    }
}
```



MOHAMMED AAZAM U T 4NM20IS083

28. Write a JavaFX program to display a window as shown below. Use TextField for User Name and Password Field for Password inputs. On click of "Sign in" Button the message "Welcome UserName" should be displayed in a Text Control. Use Grid Layout for the application.

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Text;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

public class Main extends Application {
    public void start(Stage stage) {

        Text uname = new Text("Username");

        Text pass = new Text("Password");

        Label l = new Label("");

        TextField unameField = new TextField();

        TextField passField = new TextField();
```

```
Button button1 = new Button("Sign In");

GridPane gridPane = new GridPane();

gridPane.setVgap(5);
gridPane.setHgap(5);

gridPane.setAlignment(Pos.CENTER);

gridPane.add(uname, 0, 0);
gridPane.add(unameField, 1, 0);
gridPane.add(pass, 0, 1);
gridPane.add(passField, 1, 1);
gridPane.add(button1, 0, 2);
gridPane.add(l, 1, 2);

button1.setOnAction(new EventHandler<ActionEvent>() {
    public void handle(ActionEvent ae) {
        l.setText("Welcome "+unameField.getText());
    }
});

Scene scene = new Scene(gridPane,400,200);

stage.setTitle("Welcome Username");

stage.setScene(scene);

stage.show();
}
```

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



29. Write a JavaFx program that obtains two positive integers passed from the user and display the numbers and their GCD as the result.

```
package application;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.Dialog;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.FlowPane;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;

public class gcdDemo extends Application {
    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) {

        primaryStage.setTitle("FIND GCD");
        Label l1= new Label("Enter 1st number:");
        TextField t1=new TextField();
        t1.setPrefColumnCount(15);
        Label l2= new Label("Enter 2nd number:");
        TextField t2=new TextField();
        t2.setPrefColumnCount(15);
        Button btn = new Button();
        btn.setText("Submit");

        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                int v1, v2,gcd=0;
                v1 = Integer.parseInt(t1.getText());
                v2 = Integer.parseInt(t2.getText());

                for(int i = 1; i <= v1 && i <= v2; i++)
                {
                    if(v1%i==0 && v2%i==0)
                        gcd = i;
                }

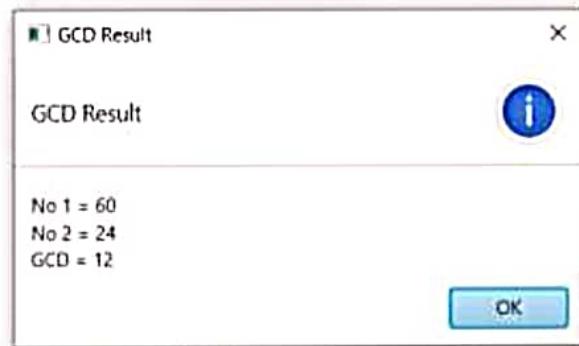
                Dialog d=new Alert(Alert.AlertType.INFORMATION,
                    "No 1 =" +Integer.toString(v1) +
                    "\nNo 2 = " + Integer.toString(v2) +
                    "\nGCD = " + Integer.toString(gcd));
                d.setTitle("GCD Result");
            }
        });
    }
}
```

```
        d.setHeaderText("GCD Result");
        d.show();
    });

    FlowPane root = new FlowPane();
    root.getChildren().addAll(l1,t1,l2,t2,btn);
    primaryStage.setScene(new Scene(root, 300, 250));
    primaryStage.show();
}
}
```

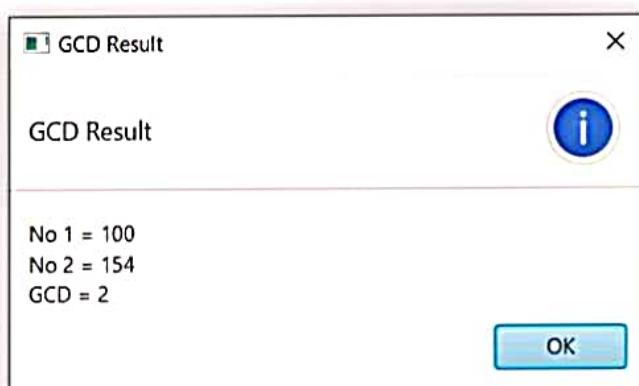
FIND GCD

Enter 1st number: Enter 2nd number: Submit



FIND GCD

Enter 1st number: Enter 2nd number: Submit



NAVYA RAO
4NM20IS085

30. Define a class called Employee with attributes name, empid, designation, basicPay, DA, HRA, PF, LIC, netSalary. DA is 40% of basicPay, HRA is 15% of basicPay, PF is 12% of basicPay. Display all the employee information in a JavaFX application.

```
package application;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.FlowPane;
import javafx.stage.Stage;

class EmpDemo1
{
    String name,designation,empid;
    double basicpay,da,hra,pf,netsal;

    double da(double basicpay) {
        return basicpay*40/100;
    }
    double hra(double basicpay) {
        return basicpay*15/100;
    }
    double pf(double basicpay) {
        return basicpay*12/100;
    }
    double netsal(double basicpay,double da,double hra,double pf)
    {
        return (basicpay+da+hra)-pf;
    }
    String getDetails(String name,String empid,
                      String designation,double basicpay,
                      double da, double hra, double pf, double netsal)
    {
        return "EMPLOYEE DETAILS\nEmployee Name: "+name+
```

```

"\nEmployee ID: "+empid+"\nEmployee Designation:
"+designation+"\nBasic Pay: "+basicpay+"\nDearness
Allowance (DA): "+da+"\nHouse Rent Allowance (HRA):
"+hra+"\nProvident Fund (PF): "+pf+"\nNet Salary:
"+netsal;
}
}

public class EmpDetails extends Application
{
    String name1,empid1,desig1;
    double bspay1,da1,hra1,pf1,netsal1;

    Label namelb,empidlb,desiglb,bspaylb;
    TextField nametf,empidtf,desigtf,bspaytf;
    Label response;
    String Details="";
    Button b;

    public void start(Stage mystage) throws Exception {
        mystage.setTitle("Employee Details");
        FlowPane root=new FlowPane(30,30);
        Scene scene=new Scene(root,1300,500);
        mystage.setScene(scene);

        root.setAlignment(Pos.TOP_CENTER);

        namelb=new Label("Name:");
        nametf=new TextField();

        empidlb=new Label("Employee ID:");
        empidtf=new TextField();

        desiglb=new Label("Designation:");
        desigtf=new TextField();

        bspaylb=new Label("Basic Pay:");
        bspaytf=new TextField();

        response=new Label("");

        b=new Button("Get Employee Details");
        b.setOnAction(new EventHandler<ActionEvent>() {

```

```
public void handle(ActionEvent ae) {
    name1=nametf.getText();
    empid1=empidtf.getText();
    desig1=desigtf.getText();
    try
    {
        bspay1=Double.parseDouble(bspaytf.getText());
    }
    catch(Exception e) {System.out.println(e);}
    EmpDemo1 e =new EmpDemo1();
    da1=e.da(bspay1);
    hra1=e.hra(bspay1);
    pf1=e.pf(bspay1);
    netsal1=e.netsal(bspay1,da1,hra1,pf1);
    Details=e.getDetails(name1,empid1,desig1,bspay1,da1,hra1,pf1,netsal1);
    response.setText(Details);
    }});
    root.getChildren().addAll(namelb,nametf,empidlb,
        empidtf,desiglb,desigtf,bspaylb,bspaytf,b,response
    );
    mystage.show();
}
public static void main(String[] args)
{
    Launch(args);
}
}
```

OUTPUT:

Employee Details

Name:	Amar	Employee ID:	12345	Designation:	Manager	Basic Pay:	100000	Get Employee Details
-------	------	--------------	-------	--------------	---------	------------	--------	--------------------------------------

EMPLOYEE DETAILS
Employee Name: Amar
Employee ID: 12345
Employee Designation: Manager
Basic Pay: 100000.0
Dearness Allowance (DA): 40000.0
House Rent Allowance (HRA): 15000.0
Provident Fund (PF): 12000.0
Net Salary: 143000.0