ASSIGNMENT 01

1) Write a Java program to check whether an input number is Palindrome or not.

```
import java.util.Scanner;

public class Pali {
    public static void main(String args[])
    {
        int x,number, y,temp=0;
        Scanner in=new Scanner(System.in);
        System.out.println("Enter any number: ");
        number=in.nextInt();
        y = number;
        while(number!=0)
        {
            x=number%10;
            temp=temp*10+x;
            number=number/10;
        }
        if(temp==y)
        {
            System.out.println("Number is Palindrome");
        }
        else
        {
            System.out.println("not Palindrome");
        }
    }
}
```

OUTPUT:

```
Enter any number:
121
Number is Palindrome
```

2) Write a Java program to check whether an input number is Armstrong or not.

```
import java.util.*;
class Ang{
    public static void main(String[] args) {
        int num;
        try (Scanner in = new Scanner(System.in)) {
            System.out.println("Enter the number");
            num = in.nextInt();
        }
        int d=0, comp = num, temp = num;
        while(num!=0){
            d++;
            num = num / 10;
        }
        int s, sum = 0;
        while(comp!=0){
            s = comp % 10;
            sum = sum + (int)Math.pow(s, d);
            comp = comp / 10;
        }
        if(sum == temp){
            System.out.println("It is a Angstrome");
        }
        else{
            System.out.println("It is not a Angstrome number");
        }
    }
}
```

OUTPUT:

```
Enter the number
153
It is a Angstrome
```

3) Write a Java program to find factorial of a number.

```
import java.util.Scanner;

public class Fact {
    public static void main(String[] args) {
        int number, res=1;
        Scanner in=new Scanner(System.in);
        System.out.println("Enter any number: ");
        number=in.nextInt();
        for(int i = 1; i<=number; i++){
            res = res * i;
        }
        System.out.println(number+"! = "+res);
    }
}
```

OUTPUT:

```
Enter any number:
5
5! = 120
```

4) Write a Java program to find GCD and LCM of two number.

```
import java.util.*;

public class GCD {
    public static void main(String[] args) {
        int num1, num2, gcd = 1;
        try (Scanner in = new Scanner(System.in)) {
            System.out.println("Enter first number");
            num1 = in.nextInt();
            System.out.println("Enter second number");
            num2 = in.nextInt();
        }
        int n1 = num1;int n2 = num2;
        while(num1!=num2){
            if(num1>num2){
                num1 = num1 - num2;
            }
            else if(num2>num1){
                num2 = num2-num1;
            }
            else{
                gcd = num1;
            }
        }
        int lcm = (n1*n2)/gcd;
        System.out.println("GCD of " + n1 + " and " + n2 + " is " + gcd);
        System.out.println("LCM of " + n1 + " and " + n2 + " is " + lcm);

    }
}
```

OUTPUT:

```
Enter first number
2
Enter second number
4
GCD of 2 and 4 is 1
LCM of 2 and 4 is 8
```

5) Write a Java program to check whether a input is prime or not.

```
import java.util.Scanner;

public class Prime {

  public static void main(String[] args) {

    int num;
    try (Scanner in = new Scanner(System.in)) {
      System.out.println("Enter the number");
      num = in.nextInt();
    }
    boolean flag = false;
    for (int i = 2; i <= num / 2; ++i) {
      // condition for nonprime number
      if (num % i == 0) {
        flag = true;
        break;
      }
    }

    if (!flag)
      System.out.println(num + " is a prime number.");
    else
```

```
      System.out.println(num + " is not a prime number.");
  }
}
```

OUTPUT:

```
Enter the number
2
2 is a prime number.
```

NAME – PRIYANSHU MALLICK
SIC – 21BCSF11
ROLL. NO – 30
SEC – B, GROUP – B2
BRANCH - CSE

ASSIGNMENT 02

1) Write a Java program to find all prime numbers between given pair of range.

```
import java.util.Scanner;

public class PrimeRange {
    public static void main(String[] args) {
        int num1, num2;
        try (Scanner in = new Scanner(System.in)) {
            System.out.println("Enter first number");
            num1 = in.nextInt();
            System.out.println("Enter second number");
            num2 = in.nextInt();
        }
        while (num1 < num2) {
            boolean flag = false;

            for(int i = 2; i <= num1/2; ++i) {
                // condition for nonprime number
                if(num1 % i == 0) {
                    flag = true;
                    break;
                }
            }

            if (!flag && num1 != 0 && num1 != 1)
                System.out.print(num1 + " ");

            ++num1;
        }
    }
}
```

OUTPUT:

```
Enter any number:
121
Number is Palindrome
```

2) Write a Java program to find all Armstrong numbers between a given pair of range.

```
import java.util.*;

class AngRange {
    public static void main(String[] args) {
        int num1, num2;
        try (Scanner sc = new Scanner(System.in)) {
            System.out.println("Enter the first number ::");
            num1 = sc.nextInt();
            System.out.println("Enter the second number ::");
            num2 = sc.nextInt();
        }
        for(int number = num1 + 1; number < num2; ++number) {
            int digits = 0;
            int result = 0;
            int originalNumber = number;

            // number of digits calculation
            while (originalNumber != 0) {
              originalNumber /= 10;
              ++digits;
            }

            originalNumber = number;

            // result contains sum of nth power of its digits
            while (originalNumber != 0) {
              int remainder = originalNumber % 10;
              result += Math.pow(remainder, digits);
              originalNumber /= 10;
            }

            if (result == number) {
              System.out.print(number + " ");
            }
        }
    }
}
```

OUTPUT:

```
Enter the number
153
It is a Angstrome
```

3) Write a Java program to find all Palindrome numbers within a given pair of range.

```
import java.util.Scanner;

public class Pali {
    public static void main(String args[]) {
        int x, number, start, end, y, temp = 0;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter The Range : ");
        start = in.nextInt();
        end = in.nextInt();
        for (number = start; number <= end; number++) {
            temp=0;
            y = number;
            while (number != 0) {
                x = number % 10;
                temp = temp * 10 + x;
                number = number / 10;
            }
            if (temp == y)
                System.out.print(y + ", ");
        }
    }
}
```

OUTPUT:

```
Enter The Range:
10 20
11
```

4) Write a Java program to Fibonacci series up to a given range.

```
import java.util.*;
class Fibo{
    public static void main(String args[]){
        int n;
        try (Scanner in = new Scanner(System.in)) {
            System.out.println("Enter the last number");
            n = in.nextInt();
        }
        System.out.print("0, 1, ");
        int n1 = 0, n2 = 1;
        for(int i = 2; i<=n-1; i++, n1++,n2++){
            int sum = n1 + n2;
            System.out.print(sum+", ");
        }
    }
}
```

OUTPUT:

```
Enter the last number
13
0, 1, 1, 3, 5, 7, 9, 11, 13,
```

5) Write a Java program to check if the square root of reverse of a number is equal to reverse of that original number.

```
import java.util.Scanner;

public class rev {
    public static void main(String[] args) {
        int n;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the number");
        n = in.nextInt();
        int t = n;
        int t1, t2, rev_n = 0, rev_sq = 0;
        while(n!=0){
            t1 = n%10;
            rev_n = rev_n * 10 + t1;
            n = n / 10;
        }
        int sq_of_n = t * t;

        while(sq_of_n != 0){
            t2 = sq_of_n % 10;
            rev_sq = rev_sq * 10 + t2;
            sq_of_n = sq_of_n/10;
        }

        int sqr_rev_sqn = (int) Math.sqrt(rev_sq);

        if(sqr_rev_sqn == rev_n){
            System.out.println("Yes");
        }
        else{
            System.out.println("No");
        }
    }
}
```

OUTPUT:

```
Enter the number
12
Yes
```

6) Write a Java program to display the multiplication table up to a given number.

```
import java.util.*;
class MultiTable{
    public static void main(String args[]){
        int n;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the last number");
        n = in.nextInt();
        int r;
        System.out.println("");
        System.out.println("Multiplication  Table of 1 is: ");
        for(int j = 1; j<=n; j++){
            for(int i = 1; i<=10; i++){
                r = j * i;
                System.out.println(j+" X "+i+" = "+r);
            }
            System.out.println("");
            System.out.println("Multiplication  Table of "+(j+1)+"is: ");
        }
    }
}
```

OUTPUT:

```
Enter the last number
3
```

```
Multiplication  Table of 1 is:
1 X 1 = 1
1 X 2 = 2
1 X 3 = 3
1 X 4 = 4
1 X 5 = 5
1 X 6 = 6
1 X 7 = 7
1 X 8 = 8
1 X 9 = 9
1 X 10 = 10

Multiplication  Table of 2is:
2 X 1 = 2
2 X 2 = 4
2 X 3 = 6
2 X 4 = 8
2 X 5 = 10
2 X 6 = 12
2 X 7 = 14
2 X 8 = 16
2 X 9 = 18
2 X 10 = 20

Multiplication  Table of 3is:
3 X 1 = 3
3 X 2 = 6
3 X 3 = 9
3 X 4 = 12
3 X 5 = 15
3 X 6 = 18
3 X 7 = 21
3 X 8 = 24
3 X 9 = 27
3 X 10 = 30
```

NAME – PRIYANSHU MALLICK
SIC – 21BCSF11
ROLL. NO – 30
SEC – B, GROUP – B2
BRANCH - CSE

ASSIGNMENT 03

1) Write a Java program to display the following pattern.

```
*
*   *
*   *   *
*   *   *   *
*   *   *   *   *
```

```java
import java.util.*;
class P1{
    public static void main(String args[]){
        int n;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the row size");
        n = in.nextInt();

        for(int i = 1; i<=n; i++){
            for(int j = 1; j<=i; j++){
                System.out.print("* ");
            }
            System.out.println("");
        }
    }
}
```

OUTPUT:

```
Enter the row size

5

*

* *

* * *

* * * *

* * * * *
```

2) Write a Java program to display the following pattern.

```
            1
         2  1
      3  2  1
   4  3  2  1
5  4  3  2  1
   4  3  2  1
      3  2  1
         2  1
            1
```

```
public class P2 {

    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            for (int j = 4; j >= i; j--)
                System.out.print(" ");
            for (int k = i; k >= 1; k--)
                System.out.print(k);
            System.out.println();
        }
        for (int x = 4; x >= 1; x--) {
            for(int space=5-x;space>0;space--)
                System.out.print(" ");
            for (int y = x; y > 0; y--)
                System.out.print(y);
            System.out.println();
        }
    }
}
```

OUTPUT:

```
    1
   21
  321
 4321
54321
 4321
  321
   21
    1
```

3) Write a Java program to display the following pattern.

```
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
  1 2 3 4 3 2 1
    1 2 3 2 1
      1 2 1
        1
```

```
public class P3{
    public static void main(String[] args){
        int n = 5;
        // Scanner in = new Scanner(System.in);
        int i, j, k, l;
        //Syste.out.println("Enter the number of row");
        for(i=1;i<=n;i++) {
            for(j=i;j<=n-1;j++) {
                System.out.print(" ");
                System.out.print(" ");
```

```
      }
      for(j=1;j<=i;j++) {
        System.out.print(j);
        System.out.print(" ");
      }
      for(j=i-1;j>=1;j--) {
        System.out.print(j);
        System.out.print(" ");
      }
      System.out.println();
    }
    k=1;
    for(i=4;i>=n-k && k<=4 ;i--,k++) {
      for(j=i;j<=n-1;j++) {
        System.out.print(" ");
        System.out.print(" ");
      }
      for(j=1;j<=i;j++) {
        System.out.print(j);
        System.out.print(" ");

      }
      for(j=i-1;j>=1;j--) {
        System.out.print(j);
        System.out.print(" ");
      }
      System.out.println();
    }
  }
}
```

OUTPUT:

```
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
  1 2 3 4 3 2 1
    1 2 3 2 1
      1 2 1
        1
```

4) Write a Java program to display the following pattern.

```
    *
   * *
  *   *
 *     *
*       *
 *     *
  *   *
   * *
    *
```

```
import java.util.Scanner;
public class P4
{
    public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);

    System.out.println("Enter the number of rows: ");

    int rows = sc.nextInt();
    for (int i=1; i<= rows ; i++) { for (int j = rows; j > i ; j--) {
            System.out.print(" ");
        }
        System.out.print("*");
        for (int k = 1; k < 2*(i -1) ;k++) { System.out.print(" "); } if( i==1) {
System.out.println(""); } else { System.out.println("*"); } } for (int i=rows-1; i>= 1 ;
i--)
        {
        for (int j = rows; j > i ; j--) {
            System.out.print(" ");
        }
        System.out.print("*");
        for (int k = 1; k < 2*(i -1) ;k++) {
            System.out.print(" ");
        }
        if( i==1)
            System.out.println("");
        else
            System.out.println("*");
    }
    sc.close();
}
}
```

OUTPUT:

```
Enter the number of rows:
5
    *
   * *
  *   *
 *     *
*       *
 *     *
  *   *
   * *
    *
```

5) Write a Java program to display the following pattern.

```
        A
      B A B
    C B A B C
  D C B A B C D
E D C B A B C D E
```

```
import java.util.*;

class P5
{
    public static void main(String args[])
    {
        int i,j,n;
        Scanner sc = new Scanner(System.in);
            System.out.println("Enter the no of lines");
        n=sc.nextInt();
            for(i=1;i<=n;i++)
            {

                for(j=1;j<=n-i;j++)
                {
                    System.out.printf("  ");
            }
                for(j=i;j>0;j--)
                {
                    System.out.printf(" "+(char)(j+64));
            }
                for(j=2;j<=i;j++)
                {
                    System.out.printf(" "+(char)(j+64));
            }
                System.out.println();
            }
      }
}
```

OUTPUT:

```
Enter the no of lines
5
         A
       B A B
     C B A B C
   D C B A B C D
 E D C B A B C D E
```

6) Write a Java program to display the following pattern.

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

```java
class P6{
    public static void main(String[] args){
        int i, j, k=1, n = 5, s=0;
        for(i=1; i<=n; i++){
            if(i%2!=0){
                for(j=1; j<=i;j++,k++){
                    s = k;
                }
                for(j=1; j<=i;j++, s--){
                    System.out.print(s);
                    System.out.print(" ");
                }
            }
            else{
                for(j=1; j<=i;j++,k++){
                    System.out.print(k);
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
    }
}
```

OUTPUT:

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

NAME – PRIYANSHU MALLICK
SIC – 21BCSF11
ROLL. NO – 30
SEC – B, **GROUP** – B2
BRANCH - CSE

ASSIGNMENT 04

1) Design a class Time having data members hour, minute and second. It should also have methods displayTime() to display the time in HH:MM:SS format and addTime() to add two Time objects. Test these methods by creating a main() method in another class. Implement it by using parameterized constructor to initialize its data members.

```
import java.util.*;
class Time{
    public int hour;
    public int minute;
    public int second;
    public int day = 0;

    Time(){
    }
    Time(int h, int m, int s){
        hour = h;
        minute = m;
        second = s;
    }


    void add(Time ob1, Time ob2){
        hour = ob1.hour + ob2.hour;
        minute = ob1.minute + ob2.minute;
        second = ob1.second + ob2.second;

        if(second>=60){
            minute = minute + (second/60);
            second = second % 60;
        }
        if(minute>=60){
            hour = hour + (minute/60);
            minute = minute % 60;
        }
        if(hour>=24){
            day = day + (hour/24);
            hour = hour % 24;
        }
    }
}
```

OUTPUT:

```
Result Time is: 1 D 10 H 56 M 23 S
```

2) Create a class Complex for performing arithmetic with complex numbers. Complex numbers have the form realPart + imaginaryPart * i. Write a program to test your class. Use floating point variables to represent the private data of the class. Provide a constructor that enables an object of this class to be initialized when it is declared.

Provide a no argument constructor with default values in case no initializers are provided. Provide public methods that perform the following operations:

a. Add two complex numbers: The real parts are added together and the imaginary parts are added together.

b. Subtract two complex numbers: The real parts of the right operand is subtracted from the real part of the left operand, and the imaginary part of the right operand is subtracted from the imaginary part of the left operand.

c. Print complex numbers in the form (realPart, imaginaryPart)

```java
import java.util.*;
class Complex{
    public int real;
    public int img;

    Complex(){
        real = 1;
        img = 1;
    }
    Complex(int r, int i){
        real = r;
        img = i;
    }

    void add(Complex ob1, Complex ob2){
        real = ob1.real + ob2.real;
        img = ob1.img + ob2.img;
    }
    void sub(Complex ob1, Complex ob2){
        real = ob1.real - ob2.real;
        img = ob1.img - ob2.img;
    }
}

class Driver4{
    public static void main(String args[]){
        /*Complex ob1 = new Complex();
        Complex ob2 = new Complex();
        Complex ob3 = new Complex();
        ob1.input();
        ob2.input();
        ob3.add(ob1, ob2);*/
        int real1, real2, img1, img2;
        Complex ob1 = new Complex();

        Scanner in = new Scanner(System.in);
        System.out.println("Enter the real part of num1");
        real1 = in.nextInt();
        System.out.println("Enter the imaginary of num1");
        img1 = in.nextInt();
        System.out.println("Enter the real part of num2");
        real2 = in.nextInt();
        System.out.println("Enter the imaginary of num2");
        img2 = in.nextInt();

        Complex ob2 = new Complex(real1, img1);
        Complex ob3 = new Complex(real2, img2);

        Complex ob4 = new Complex();
        Complex ob5 = new Complex();

        ob4.add(ob2, ob3);
        ob5.sub(ob2, ob3);

        System.out.println("Addition Result: "+ob4.real+" + "+ob4.img+"i");
        System.out.println("Subtraction Result: "+ob5.real+" + "+ob5.img+"i");
    }
}
```

OUTPUT:

```
Enter the real part of num1
4
Enter the imaginary of num1
5
Enter the real part of num2
2
Enter the imaginary of num2
3
Addition Result: 6 + 8i
Subtraction Result: 2 + 2i
```

3) Design a class named Rectangle to represent a rectangle. The class contains:

a. Two double data fields names width and height that specify the width and height of the rectangle. The default values are 1 for both width and height.

b. A no-argument constructor that creates a default rectangle.

c. A constructor that creates a rectangle with the specified width and height.

d. A method named getArea() that returns the area of this rectangle. e. A method named getPerimeter() that returns the perimeter.

```
class Rectangle{
    double width;
    double height;
    Rectangle(){
        width = 1;
        height = 1;
    }
    Rectangle(double w, double h){
        width = w;
        height = h;
    }
    double getArea(){
        return width*height;
    }
    double getPerimeter(){
        return 2*(width+height);
    }
}
class Driver{
    public static void main(String args[]){

        Rectangle ob1 = new Rectangle(20,30);
        double area = ob1.getArea();
        double perimeter = ob1.getPerimeter();
        System.out.println("The area of rectangle is "+area+" m^2 and the perimeter is 
"+perimeter+" m");
    }
}
```

OUTPUT:

```
The area of rectangle is 600.0 m^2 and the perimeter is 100.0 m
```

4) Write a program to find the sum and average if values of an array.

```
import java.util.Scanner;

public class Arr {
    public static void main(String[] args) {
        int arr[] = new int[10];
        int n;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the size of the array");
        n = in.nextInt();
        System.out.println("Enter the array");
        for(int i = 0; i<n; i++){
            arr[i] = in.nextInt();
        }
        float sum = 0;
        for(int i = 0; i<n; i++){
            sum = sum + arr[i];
        }
        float avg = sum/n;
        System.out.println("Sum is: "+sum+" Avarage is: "+avg);
    }
}
```

OUTPUT:

```
Enter the size of the array
5
Enter the array
1 3 2 4 1
Sum is: 11.0 Avarage is: 2.2
```

5) Write a program to implement linear search. If present, display the number of times it is present.

```
import java.util.*;
public class LS {
    public static void main(String[] args) {
        int arr[] = new int[10];
        int n, c = 0;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the size of the array");
        n = in.nextInt();
        System.out.println("Enter the array");
        for(int i = 0; i<n; i++){
            arr[i] = in.nextInt();
        }
        System.out.println("Enter the key");
        int k = in.nextInt();
        for(int i = 0; i<n; i++){
            if(k == arr[i]){
                c++;
            }
        }
        if(c==0){
            System.out.println("The key is not present");
        }
        else{
            System.out.println(c+" number of times the key is present");
        }
    }
}
```

OUTPUT:

```
Enter the size of the array
5
Enter the array
1 1 1 2 3
Enter the key
1
3 number of times the key is present
```

6) Design a class Student having data members age and mark. It should have methods input() to take details of student and show() method to display the details of the student. Test these methods by creating an array of object of Student.

```
import java.util.Scanner;

class Student{
    int age;
    float mark;
    void input(){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the age of the student");
        age = in.nextInt();
        System.out.println("Enter the mark");
        mark = in.nextFloat();
    }
    void show(){
        System.out.println("Age: "+age);
        System.out.println("Mark: "+mark);
    }

}
public class Driver {
    public static void main(String[] args) {
        Student st[];
        st = new Student[10];
        int n;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the numebr of student");
        n = in.nextInt();

        for(int i=0; i<n; i++){
            st[i] = new Student();
        }

        for(int i=0; i<n; i++){
            System.out.println("Student "+(i+1));
            st[i].input();
        }
        for(int i=0; i<n; i++){
            System.out.println("Details of student "+(i+1));
            st[i].show();
        }

    }
}
```

OUTPUT:

```
Enter the numebr of student
3
Student 1
Enter the age of the student
12
Enter the mark
25
Student 2
Enter the age of the student
23
Enter the mark
24
Student 3
Enter the age of the student
12
```

```
Enter the mark
45
Details of student 1
Age: 12
Mark: 25.0
Details of student 2
Age: 23
Mark: 24.0
Details of student 3
Age: 12
Mark: 45.0
```

NAME – PRIYANSHU MALLICK
SIC – 21BCSF11
ROLL. NO – 30
SEC – B, GROUP – B2
BRANCH - CSE

ASSIGNMENT 05

1) Write a Java program to implement Single Inheritance.

```
class Base{
    void show(){
        System.out.println("Base class");
    }
}
class Derived extends Base{
    void display(){
        System.out.println("Derived class");
    }
}
class SingleInheritance{
    public static void main(String args[]){
        Derived ob = new Derived();
        ob.show();
        ob.display();
    }
}
```

OUTPUT:

```
Base class
Derived class
```

2) Write a Java program to implement Multi-Level Inheritanc.

```
class Base{
    void show(){
        System.out.println("Base class");
    }
}
class Derived1 extends Base{
    void display(){
        System.out.println("Derived1 class");
    }
}
class Derived2 extends Derived1{
    void print(){
        System.out.println("Derived2 class");
    }
}
class MultiLevelInheritance{
    public static void main(String args[]){
        Derived2 ob = new Derived2();
        ob.show();
        ob.display();
        ob.print();
    }
}
```

OUTPUT:

```
Base class
Derived1 class
Derived2 class
```

3) Write a Java program to implement Hierarchical Inheritance.

```java
class Base{
    void show(){
        System.out.println("Base class");
    }
}
class Derived1 extends Base{
    void display(){
        System.out.println("Derived1 class");
    }
}
class Derived2 extends Base{
    void print(){
        System.out.println("Derived2 class");
    }
}
class HierarchicalInheritance{
    public static void main(String args[]){
        Derived1 ob1 = new Derived1();
        Derived2 ob2 = new Derived2();
        ob1.show();
        ob1.display();
        ob2.show();
        ob2.print();
    }
}
```

OUTPUT:

```
Base class
Derived1 class
Base class
Derived2 class
```

4) Write a Java program to find the largest number between two numbers using Single Inheritance.

```java
class Base{
    int no1;
}
class Derived extends  Base{
    int no2;
    Derived(int n1, int n2){
        no1 = n1;
        no2 = n2;
    }
    void compare(){
        if(no1>no2){
            System.out.println(no1+" is the larger");
        }
        else if(no2>no1){
            System.out.println(no2+" is the larger");
        }
    }
}
public class SIEx {
    public static void main(String[] args) {
        Derived ob = new Derived(2, 3);
        ob.compare();
    }
}
```

OUTPUT:

```
3 is the larger
```

5) Write a Java program to find the smallest number between three numbers using Multi Level Inheritance.

```
import java.util.*;
class Base{
    int no1;
}
class Derived1 extends Base{
    int no2;
}
class Derived2 extends Derived1{
    int no3, c, s;
    void input(){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the first num");
        no1 = in.nextInt();
        System.out.println("Enter the 2nd num");
        no2 = in.nextInt();
        System.out.println("Enter the 3rd num");
        no3 = in.nextInt();
    }
    void findSmall(){
        if(no1<no2){
            c = no1;
        }
        else if(no1>no2){
            c = no2;
        }
        if(c<no3){
            s = c;
        }
        else if(c>no3){
            s = no3;
        }
    }
    void display(){
        System.out.println(s+" is the smallest");
    }
}
class MultiLevelInheritanceEx{
    public static void main(String args[]){
        Derived2 ob = new Derived2();
        ob.input();
        ob.findSmall();
        ob.display();
    }
}
```

OUTPUT:

```
Enter the first num
2
Enter the 2nd num
7
Enter the 3rd num
9
2 is the smallest
```

6) Write a Java program to achieve Multiple Inheritance in Java through Interface.

```
interface Backend {
    public void connectServer();
  }
  class Frontend {
    public void responsive(String str) {
      System.out.println(str + " can also be used as frontend.");
    }
  }
  class Language extends Frontend implements Backend {
    String language = "Java";
    public void connectServer() {
      System.out.println(language + " can be used as backend language.");
    }
    public static void main(String[] args) {
      Language java = new Language();
      java.connectServer();
      java.responsive(java.language);
    }
  }
```

OUTPUT:

```
Java can be used as backend language.
Java can also be used as frontend.
```

NAME – PRIYANSHU MALLICK
SIC – 21BCSF11
ROLL. NO – 30
SEC – **B, GROUP** – B2
BRANCH - CSE

ASSIGNMENT 06

1) Write a Java program to implement the Method Overloading concept.

```java
class Test{
    void add(int no1, int no2){
        int res;
        res = no1 + no2;
        System.out.println("Sum is "+res);
    }
    void add(float value1, float value2){
        float res;
        res = value1 + value2;
        System.out.println("Sum is "+res);
    }
    void add(int no1, int no2, int no3){
        int res;
        res = no1 + no2;
        System.out.println("Sum is "+res);
    }
    void add(float value){
        float res;
        res = value + 5;
        System.out.println("Sum is "+res);
    }
}
public class Demo {
    public static void main(String[] args) {
        Test ob = new Test();
        ob.add(5,6);
        ob.add(5.0f,6.0f);
        ob.add(10,20,30);
        ob.add(5.0f);
    }
}
```

OUTPUT:

```
Sum is 11
Sum is 11.0
Sum is 30
Sum is 10.0
```

2) Design the following class hierarchies with appropriate driver class and main() method. You may add more member functions if required. Also add the parameterized constructor.

Employee

String empName

int empID

double basicSalary

static int count

double DA

double HRA

Employee()

double grossSalary()

protected void finalize()

void empDetails()

↑

<u>Manager</u>

double bonus

Manager()//All constructor

void empDetails()

```
class Employee{
    String empName;
    int empID;
    double basicSalary;
    static int count;
    double DA;
    double HRA;

    Employee(){

    }
    Employee(String n, int i, double bs, int c){
        empName = n;
        empID = i;
        basicSalary = bs;
        count = c;
    }
    double grossSalary(){
        DA = basicSalary*0.15;
        HRA = basicSalary*0.6;
        return basicSalary+DA+HRA;
    }
    protected void finalize(){

    }
    void empDetails(){
        System.out.println("Name: "+empName);
        System.out.println("ID: "+empID);
        System.out.println("Basic Salary: "+basicSalary);
    }
}
class Manager extends Employee{
    double bonus;
    Manager(String n, int i, double bs, int b){
        empName = n;
        empID = i;
        basicSalary = bs;
        bonus = b;
    }
    double grossSalary(){
        DA = basicSalary*0.15;
        HRA = basicSalary*0.6;
        return basicSalary+DA+HRA+bonus;
    }
    void empDetails(){
        System.out.println("Name: "+empName);
        System.out.println("ID: "+empID);
        System.out.println("Basic Salary: "+basicSalary);
    }
}
public class Driver15 {
    public static void main(String[] args) {
        Employee ob;
        ob = new Manager("Priyanshu", 123, 12000.00, 1);
        double gs = ob.grossSalary();
        ob.empDetails();
        System.out.println("Gross Salary: "+gs);
    }
}
```

OUTPUT:

```
Name: Priyanshu
ID: 123
Basic Salary: 12000.0
Gross Salary: 21001.0
```

3) Use the concept of abstract class and abstract method for the above class. Create an object reference of Employee class using which access the necessary member function of Manager class.

```
abstract class Employee {
    String Name;
    int number;

    Employee(String n, int i) {
        Name = n;
        number = i;
    }
    abstract void printData();
}

class Manager extends Employee {
    String title, name;
    double clubDues;
    Manager(String t, double cd, String n, int i){
        super(n, i);
        title = t;
        clubDues = cd;
    }
    void printData(){
        System.out.println(Name+" "+number+" "+title+" "+clubDues);
    }
}
public class Driver10 {
    public static void main(String[] args) {
        Employee ob;
        ob = new Manager("Mr",2345,"Priyanshu",45679);
        ob.printData();
    }
}
```

OUTPUT:

```
Priyanshu 45679 Mr 2345.0
```

4) Design a interface Shape and implement in Square, Rectangle and Triangle class from Shape class. By creating the object reference of Shape to access the member area() of different shape.

```
import java.util.*;
interface Shape{
    void input();
    public abstract void area();
    abstract void show();
}
class Square implements Shape{
    int side, result;
    public void input(){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the side of the square");
        side = in.nextInt();
    }
    public void area(){
        result = side * side;
    }
    public void show(){
        System.out.println("Area of square is "+result);
    }
}
class Rectangle implements Shape{
    int side1, side2, result;
    public void input(){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the side of the rectangle");
        side1 = in.nextInt();
        side2 = in.nextInt();
    }
    public void area(){
        result = side1 * side2;
    }
    public void show(){
        System.out.println("Area of Rectangle is "+result);
    }
}
class Triangle implements Shape{
```

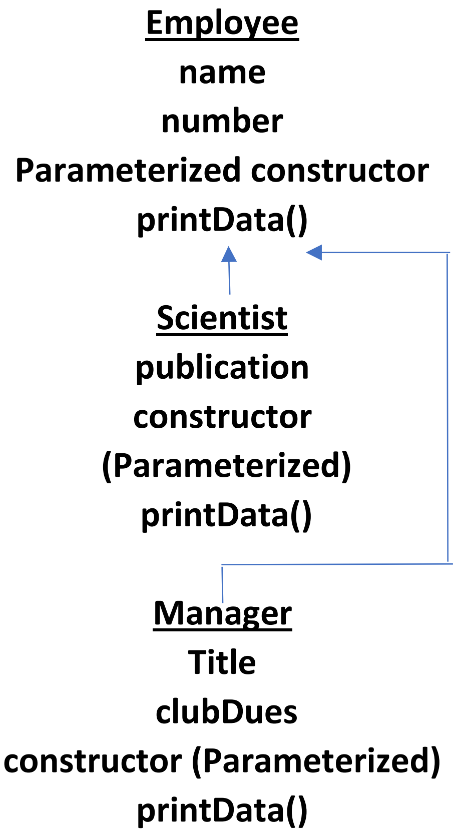
```java
    double side1, side2, side3, result;
    public void input(){
        System.out.println("FOR TRIANGLE");
        System.out.println();
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the three sides of the triangle");
        side1 = in.nextDouble();
        side2 = in.nextDouble();
        side3 = in.nextDouble();
    }
    public void area(){
        double s;
        s = (side1+side2+side3)/2;
        result = Math.sqrt(s*(s-side1)*(s-side2)*(s-side3));
    }
    public void show(){
        System.out.println("Area of square is "+result);
    }
}
class DemoInterface{
    public static void main(String args[]){
        Shape ob;
        ob = new Square();
        ob.input();
        ob.area();
        ob.show();
        ob = new Rectangle();
        ob.input();
        ob.area();
        ob.show();
        ob = new Triangle();
        ob.input();
        ob.area();
        ob.show();
    }
}
```

OUTPUT:

```
Enter the side of the square
2
Area of square is 4
Enter the side of the rectangle
2 4
Area of Rectangle is 8
FOR TRIANGLE

Enter the three sides of the triangle
4 2 6
Area of square is 0.0
```

5) Design following class hierarchies with appropriate main() function. You may add more member functions if required.



Employee

name

number

Parameterized constructor

printData()

Scientist

publication

constructor

(Parameterized)

printData()

Manager

Title

clubDues

constructor (Parameterized)

printData()

```
abstract class Employee {
    String Name;
    int number;

    Employee(String n, int i) {
        Name = n;
        number = i;
    }
    abstract void printData();
}

class Manager extends Employee {
    String title, name;
    double clubDues;
    Manager(String t, double cd, String n, int i){
        super(n, i);
        title = t;
        clubDues = cd;
    }
    void printData(){
        System.out.println(Name+" "+number+" "+title+" "+clubDues);
    }
}
class Scientist extends Employee {
    String publication, name;
    Scientist(String p, String n, int i){
        super(n, i);
        publication = p;
    }
    void printData(){
        System.out.println(Name+" "+number+" "+publication);
    }
}
public class Driver10 {
    public static void main(String[] args) {
        Employee ob;
        ob = new Manager("Mr",2345,"Priyanshu",45679);
        ob.printData();
        ob = new Scientist("KP","Chandan",23456);
        ob.printData();
    }
}
```

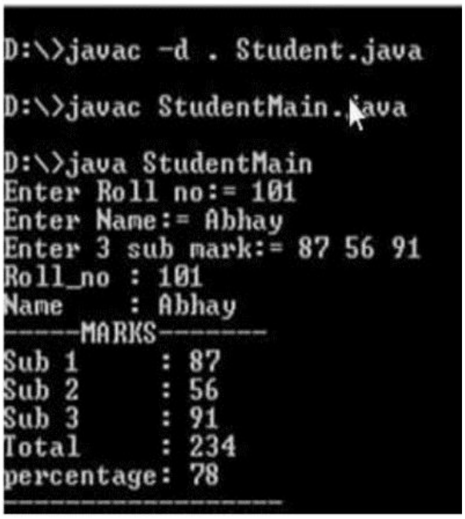
OUTPUT:

```
Priyanshu 45679 Mr 2345.0
Chandan 23456 KP
```

NAME – PRIYANSHU MALLICK
SIC – 21BCSF11
ROLL. NO – 30
SEC – B, GROUP – B2
BRANCH - CSE

ASSIGNMENT 07

1) Create a package btech which has one class Student. Accept student details through parameterized constructor of Student class. Write a method display() to display the student details. Create another class Test containing the main() method which will use the package btech and calculate total marks and percentage of marks. One sample output is shown below.



```
D:\>javac -d . Student.java

D:\>javac StudentMain.java

D:\>java StudentMain
Enter Roll no:= 101
Enter Name:= Abhay
Enter 3 sub mark:= 87 56 91
Roll_no  : 101
Name     : Abhay
-----MARKS-------
Sub 1      : 87
Sub 2      : 56
Sub 3      : 91
Total      : 234
percentage: 78
------------------
```

```
package btech;

public class Student{
    public String name;
    public int roll_no;
    public int sub1, sub2, sub3;
    public Student(){

    }
    public Student(String n, int r, int s1, int s2, int s3){
        name = n;
        roll_no = r;
        sub1 = s1;
        sub2 = s2;
        sub3 = s3;
    }
    public void display(){
        System.out.println("Roll No "+roll_no);
        System.out.println("Name: "+name);
        System.out.println("------MARK------ ");
        System.out.println("Sub 1: "+sub1);
        System.out.println("Sub 2: "+sub2);
        System.out.println("Sub 3: "+sub3);
        System.out.println("Total: "+(sub1+sub2+sub3));
        System.out.println("percentage "+((sub1+sub3+sub3)*0.3));
    }
}
```

```
import btech.Student;
public class Test {
    public static void main(String[] args) {
        Student ob = new Student("Ram", 12, 92, 94, 98);
        ob.display();
     }
}
```

OUTPUT:

```
                    Roll No 12
                    Name: Ram
                  ------MARK------
                    Sub 1: 92
                    Sub 2: 94
                    Sub 3: 98
                    Total: 284
                  percentage 86.399
```

2) Create a sub-package called arithmetic under the package btech. The arithmetic package should contain a class MyMath having methods to deal with different arithmetic operations (addition, subtraction, multiplication, division and mod). Create a class Test containing the main method which will use the methods of sub-package arithmetic.

```
package btech.arithmetic;

public class MyMath {
    public int add(int x, int y)
    {
        return x + y;
    }
    public int sub(int x, int y)
    {
        return x - y;
    }
    public int mul(int x, int y)
    {
        return x * y;
    }
    public double div(int x, int y)
    {
        return (double) x / y;
    }
    public int mod(int x, int y)
    {
        return x % y;
    }
}
```

```
import btech.arithmetic.*;
class Test
{
    public static void main(String as[])
    {
        MyMath m = new MyMath();
        System.out.println(m.add(8, 5));
        System.out.println(m.sub(8, 5));
        System.out.println(m.mul(8, 5));
        System.out.println(m.div(8, 5));
        System.out.println(m.mod(8, 5));
    }
}
```

OUTPUT:

```
13
3
40
1.6
3
```

3) Create a sub-package named shapes under package org. Create some classes in the package representing some common geometric shapes like Square, Triangle, Circle and so on. The classes should the area() and perimeter() methods in them. Compile the package. Use this package to find area and perimeter of different shapes as chosen by the user.

```
package Package.shapes;

public class Triangle {
    public int a, b, c;
    public Triangle(int s1, int s2, int s3){
        a = s1;
        b = s2;
        c = s3;
    }
    public void area(){
        double s = (a+b+c)/2;
        double area = Math.sqrt(s*(s-a)*(s-b)*(s-c));
        System.out.println("Area "+area);
    }
    public void perimeter(){
        int peri = a + b + c;
        System.out.println("Perimeter "+peri);
    }
}

package Package.shapes;

public class Circle {
    public int radius;
    public Circle(int r){
        radius = r;
    }
    public void area(){
        double area = 3.141*radius*radius;
        System.out.println("Area: "+area);
    }
    public void perimeter(){
        double peri = 2*3.141*radius;
        System.out.println("Perimeter: "+peri);
    }
}

package Package.shapes;

public class Square {
    public int side;
    public Square(int s){
        side = s;
    }
    public void area(){
        int area = side*side;
        System.out.println("Area: "+area);
    }
    public void perimeter(){
        int perimeter = 4*side;
        System.out.println("Perimeter "+perimeter);
    }
}

import Package.shapes.Triangle;
import Package.shapes.Circle;
import Package.shapes.Square;

class Test
{
    public static void main(String as[])
    {
       Triangle t = new Triangle(2, 6, 4);
       Circle c = new Circle(7);
       Square s = new Square(4);
       System.out.println("TRIANGLE");
       System.out.println("----------");
       t.area();
       t.perimeter();
       System.out.println("CIRCLE");
       System.out.println("----------");
       c.area();
       c.perimeter();
       System.out.println("SQUARE");
       System.out.println("----------");
       s.area();
       s.perimeter();
    }
}
```

OUTPUT:

```
TRIANGLE
----------
Area 0.0
Perimeter 12
CIRCLE
----------
Area: 153.90900000000002
Perimeter: 43.974000000000004
SQUARE
----------
Area: 16
Perimeter 16
```

4) Write a program to create a package named folder1 to take a number and create another package name folder2 in which you find the largest between two numbers.

```java
package folder1;

import java.util.Scanner;

public class Demo {
    public int n1;
    public void getOne(){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the 1st number");
        n1 = in.nextInt();
    }
}
```

```java
package folder2;

import folder1.Demo;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        int no2;
        Scanner in = new Scanner(System.in);
        Demo ob = new Demo();
        ob.getOne();
        System.out.println("Enter the 2nd number");
        no2 = in.nextInt();
        if(ob.n1>no2){
            System.out.println(ob.n1+" is larger");
        }
        else{
            System.out.println(no2+" is larger");
        }
    }
}
```

OUTPUT:

```
Enter the 1st number
3
Enter the 2nd number
4
4 is larger
```

ASSIGNMENT 08

1) Write a Java program to read two numbers a and b and calculate a/(a-b). The program should check the value of a-b before dividing with a, it should throw an exception if a-b is zero. In the exception handler the program should display appropriate message to the user.

Protected Access Specifiers

```
class Animal {
    protected void display() {
        System.out.println("I am an animal");
    }
}

class Dog extends Animal {
    public static void main(String[] args) {
        Dog dog = new Dog();
        dog.display();
    }
}
```

Public Access Specifier

```
class Animal {
    // public variable
    public int legCount;

    // public method
    public void display() {
        System.out.println("I am an animal.");
        System.out.println("I have " + legCount + " legs.");
    }
}

public class Main {
    public static void main( String[] args ) {
        Animal animal = new Animal();
        animal.legCount = 4;
        animal.display();
    }
}
```

OUTPUT:

```
Protected
I am an animal.
Public
I am an animal.
I have 4 legs.
```

2) Write a Java program to implement Stack using class and object.

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

```
class Stk{
    public int SIZE = 10;
    public int stack[]=new int[SIZE];
    public int top = -1;

    public boolean isFull(){
        if(top==SIZE-1){
            return true;
        }
        else{
            return false;
        }
    }
    public boolean isEmpty(){
        if(top==-1){
            return true;
        }
        else{
            return false;
        }
    }
    public void push(int x){
        if(isFull()){
            System.out.println("Stack is full");
            System.exit(1);
        }
        else{
            top = top + 1;
            stack[top] = x;
            System.out.println("The inserted element is: "+stack[top]);
        }
    }
    public int pop(){
        int de=0;
        if(isEmpty()){
            System.out.println("Stack is empty");
            System.exit(1);
        }
        else{
            de = stack[top];
            top = top - 1;
        }
        return de;
    }
    public void display(){
        for(int i = 0; i<=top;i++){
            System.out.print(stack[i]+", ");
            //System.out.println();
        }
    }
}
class StackImp{
    public static void main(String args[]){
        int v, d;
        Scanner in = new Scanner(System.in);
        Stk ob = new Stk();
        int ch;
        while(true){
            System.out.println();
            System.out.println("Choose from following options");
            System.out.println("1. Push");
            System.out.println("2. Pop");
            System.out.println("3. Display");
            ch = in.nextInt();
            switch(ch){
                case 1: System.out.println("Enter the value");
                        v = in.nextInt();
                        ob.push(v);
                        break;
                case 2: d = ob.pop();
                        System.out.println("The poped element is: "+d);
                        break;
                case 3: ob.display();
                        break;
                default: System.out.println("Invalid Input");
            }
        }
    }
}
```

OUTPUT:

```
Choose from following options
1. Push
2. Pop
3. Display
1
Enter the value
2
The inserted element is: 2

Choose from following options
1. Push
2. Pop
3. Display
1
Enter the value
3
The inserted element is: 3

Choose from following options
1. Push
2. Pop
3. Display
1
Enter the value
5
```

```
The inserted element is: 5

Choose from following options
1. Push
2. Pop
3. Display
3
2, 3, 5,
Choose from following options
1. Push
2. Pop
3. Display
2
The poped element is: 5

Choose from following options
1. Push
2. Pop
3. Display
3
2, 3,
Choose from following options
1. Push
2. Pop
3. Display
```

3) Write a Java program to implement Linear Queue by using class and object.

```java
import java.util.*;
class Qu{
    public int SIZE = 10;
    public int q[]=new int[SIZE];
    public int front = -1;
    public int rear = -1;

    public boolean isFull(){
        if(rear==SIZE-1){
            return true;
        }
        else{
            return false;
        }
    }
    public boolean isEmpty(){
        if(front==-1 && front == rear +1){
            return true;
        }
        else{
            return false;
        }
    }
    public void insert(int x){
        if(isFull()){
            System.out.println("Queue is full");
            System.exit(1);
        }
        else{
            if(front == -1){
                front = 0;
            }
            rear = rear + 1;
            q[rear] = x;
            System.out.println("The inserted element is: "+q[rear]);
        }
    }
    public int delete(){
        int de=0;
        if(isEmpty()){
            System.out.println("Queue is empty");
            System.exit(1);
        }
        else{
            de = q[front];
            front = front + 1;
        }
        return de;
    }
    public void display(){
        if(isEmpty()){
            System.out.println("Queue is empty");
        }
        System.out.println("Queue is");
        for(int i=front;i<=rear;i++){
            System.out.print(q[i]+", ");
        }
        System.out.println();
    }
}
class QueueImp{
    public static void main(String args[]){
        int v, d;

        Scanner in = new Scanner(System.in);
        Qu ob = new Qu();
        int ch;
        while(true){
            System.out.println();
            System.out.println("Choose from following options");
            System.out.println("1. Insert");
            System.out.println("2. Delete");
            System.out.println("3. Display");
            ch = in.nextInt();
            switch(ch){
                case 1: System.out.println("Enter the value");
                        v = in.nextInt();
                        ob.insert(v);
                        break;
                case 2: d = ob.delete();
```

```
                        System.out.println("The poped element is: "+d);
                        break;
                case 3: ob.display();
                        break;
                default: System.out.println("Invalid Input");
            }
        }
    }
}
```

OUTPUT:

```
Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
2
The inserted element is: 2

Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
3
The inserted element is: 3

Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
5
The inserted element is: 5

Choose from following options
1. Insert
2. Delete
3. Display
3
Queue is
2, 3, 5,

Choose from following options
1. Insert
2. Delete
3. Display
2
The poped element is: 2
```

```
Choose from following options
1. Insert
2. Delete
3. Display
3
Queue is
3, 5,

Choose from following options
1. Insert
2. Delete
3. Display
```

4) Write a Java program to implement Circular Queue by using class and object.

```java
package folder1;


import java.util.Scanner;

public class Demo {
    public int n1;
    public void getOne(){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the 1st number");
        n1 = in.nextInt();
    }
}
```

```java
package folder2;

import folder1.Demo;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        int no2;
        Scanner in = new Scanner(System.in);
        Demo ob = new Demo();
        ob.getOne();
        System.out.println("Enter the 2nd number");
        no2 = in.nextInt();
        if(ob.n1>no2){
            System.out.println(ob.n1+" is larger");
        }
        else{
            System.out.println(no2+" is larger");
        }
    }
}
```

OUTPUT:

```
Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
2
The inserted element is: 2

Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
3
The inserted element is: 3

Choose from following options
1. Insert
2. Delete
3. Display
1
Enter the value
4
The inserted element is: 4

Choose from following options
1. Insert
2. Delete
3. Display
3
Queue is
2,
3,
4,

Choose from following options
1. Insert
2. Delete
3. Display
2
The poped element is: 2

Choose from following options
1. Insert
2. Delete
3. Display
3
```

```
Queue is
3,
4,

Choose from following options
1. Insert
2. Delete
3. Display
```

ASSIGNMENT 09

1) Write a Java program to read two numbers a and b and calculate a/(a-b). The program should check the value of a-b before dividing with a, it should throw an exception if a-b is zero. In the exception handler the program should display appropriate message to the user.

```java
import java.util.Scanner;

public class Ex1 {
    public static void main(String[] args) {
        int a, b;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter two numbers");
        a = in.nextInt();
        b = in.nextInt();
        int r;
        int d = a-b;
        try{
            r = a/d;
            System.out.println("Result of a/a-b = "+r);
        }
        catch(ArithmeticException e){
            System.out.println(e);
        }
    }
}
```

OUTPUT:

```
Enter two numbers
2
2
java.lang.ArithmeticException: / by zero
```

2) Write a class Account with the following properties and methods: Properties: String name, int acc_no, double balance Methods: void deposit(int num), void withdraw(int num), void transfer(Account acc1, Account acc2, int amt) Assume that an account needs to have a minimum balance of 500. If an attempt is made to withdraw or transfer, which results in balance going below 500, throw a user-defined exception called MinimunBalanceException. Use throw and throw wherever necessary

```java
import java.util.Scanner;

class MinBalEx extends Exception {
    String str;

    MinBalEx(String str) {
        this.str = str;
    }

    public String toString() {
        return str;
    }
}

class Account {
    String name;
    int acc_no;
    double balance = 500;

    void deposit(int num) {
        balance = balance + num;
        System.out.println("After deposite the account balance is: " + balance);
    }

    void withdraw(int num) throws MinBalEx {
        double d = balance - num;
        if (d <= 500) {
            throw new MinBalEx("MinimunBalanceException");
        } else {
            balance = balance - num;
            System.out.println("After withdral your account balance is: " + balance);
        }
    }

    void transfer(Account acc1, Account acc2, int amt) throws MinBalEx {
        double d = acc1.balance - amt;
        if (d <= 500) {
            throw new MinBalEx("MinimunBalanceException");
        } else {
            acc1.balance = acc1.balance - amt;
            System.out.println("Your trasaction is successfull, After tranfer your accont balance is: " +
acc1.balance);
        }
        acc2.balance = acc2.balance + amt;
        System.out.println(
                "Your bank account has credited by " + amt + " rupees, Now your bank balance is: " +
acc2.balance);
    }
}

public class Bank {
    public static void main(String[] args) throws MinBalEx {
        Scanner in = new Scanner(System.in);
        Account ac1 = new Account();
        Account ac2 = new Account();
        Account ac3 = new Account();
        int ch;
        int f = 0;
        while (f==0) {
            System.out.println("Enter your choice");
            System.out.println("0. Terminate the transaction");
            System.out.println("1. Deposite");
            System.out.println("2. Withdrawal");
            System.out.println("3. Transfer");
            ch = in.nextInt();

            switch (ch) {
                case 0:
                    f=1;
                    break;
                case 1:
                    int amt;
                    System.out.println("Enter the amount to deposit");
                    amt = in.nextInt();
                    ac1.deposit(amt);
                    break;
                case 2:
                    int am;
                    System.out.println("Enter the amount to withdraw");
```

```java
                    am = in.nextInt();
                    try{
                        ac1.withdraw(am);
                    }
                    catch(MinBalEx e){
                        System.out.println(e);
                    }
                    break;
                case 3:
                    int amnt;
                    System.out.println("Enter the amount to tranfer");
                    amnt = in.nextInt();
                    ac3.transfer(ac1, ac2, amnt);
                    break;
                default:
                    System.out.println("Invalid Choice");
                    break;
            }
        }
    }
}
```

OUTPUT:

```
Enter your choice
0. Terminate the transaction
1. Deposite
2. Withdrawal
3. Transfer
1
Enter the amount to deposit
2500
After deposite the account balance is: 3000.0
Enter your choice
0. Terminate the transaction
1. Deposite
2. Withdrawal
3. Transfer
2
Enter the amount to withdraw
2600
MinimunBalanceException
Enter your choice
0. Terminate the transaction
1. Deposite
2. Withdrawal
3. Transfer
3
Enter the amount to tranfer
1500
Your trasaction is successfull, After tranfer your accont balance is: 1500.0
Your bank account has credited by 1500 rupees, Now your bank balance is: 2000.0
Enter your choice
0. Terminate the transaction
1. Deposite
2. Withdrawal
3. Transfer
0
```

4) Write Java programs to implement all cases of exception handling:

a. ArithmeticException

b. ArrayIndexOutOfBoundsException

c. NullPointerException

d. NumberFormatException

```
// a. Arithmatic Exception
public class ExTy {
    public static void main(String[] args) {
        int no=10;
        try{
            int res = no/0;
            System.out.println(res);
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
OUTPUT:
Java.lang.ArithmeticException
```

```
// b. ArrayIndexOfBoundsException
public class ExTy {
    public static void main(String[] args) {
        int arr[] = new int[1];
        try{
            System.out.println(arr[1]);
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
```

OUTPUT:

```
java.lang.ArrayIndexOutOfBoundsException: Index 1 out of bounds for length 1
```

```
// c. NullPointerException
public class ExTy {
    public static void main(String[] args) {
        String str = null;
        try{
            int no = str.length();
            System.out.println(no);
        }
        catch(Exception e){
            System.out.println(e);
        }

    }
}
OUTPUT:
java.lang.NullPointerException: Cannot invoke "String.length()" 
because "<local1>" is null
```

```
// d. NumberFormatException
public class ExTy {
    public static void main(String[] args) {
        String str = "A";
        try{
            int no = Integer.parseInt(str);
            System.out.println(no);
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
OUTPUT:
java.lang.NumberFormatException: For input string: "A"
```

4) Write Java programs to implement exception handling by using:

a. try block and multiple catch block

b. Nested try-catch block

c. finally block

d. throw keyword

e. throws keyword

f. User-defined Exception/ Custom Exception

g. Generic Exception

a.

```
import java.util.Scanner;
public class ExTy {
    public static void main(String[] args) {
        int no1, no2, res;
        System.out.println("Welcome");
        Scanner in = new Scanner(System.in);
        try{
            System.out.println("Enter valu1");
            no1 = in.nextInt();
            System.out.println("Enter value");
            no2 = in.nextInt();
            res = no1/no2;
            System.out.println("Result is "+res);
        }
        catch(NullPointerException e){
            System.out.println(e);
        }
        catch(ArithmeticException e){
            System.out.println(e);
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println(e);
        }
        System.out.println("Over");
    }
}
```

OUTPUT:

```
Welcome
Enter valu1
4
Enter value
0
java.lang.ArithmeticException: / by zero
```

b.

```
class ExTy {
    public static void main(String args[])
    {
        try {
            int a[] = { 1, 2, 3, 4, 5 };
            System.out.println(a[5]);
            try {
                int x = a[2] / 0;
            }
            catch (ArithmeticException e2) {
                System.out.println("division by zero is not possible");
            }
        }
        catch (ArrayIndexOutOfBoundsException e1) {
            System.out.println("ArrayIndexOutOfBoundsException");
            System.out.println("Element at such index does not exists");
        }
    }
}
```

OUTPUT :

```
ArrayIndexOutOfBoundsException
Element at such index does not exists
```

c.

```
class ExTy {
    public static void main(String[] args)
    {
        try {
            System.out.println("inside try block");
            System.out.println(34 / 2);
        }
        catch (ArithmeticException e) {
            System.out.println("Arithmetic Exception");
        }
        finally {
            System.out.println("finally : i execute always.");
        }
    }
}
```

OUTPUT:

```
inside try block
17
finally : i execute always.
```

d.

```
class ExTy {
    static void fun()
    {
        try
        {
            throw new NullPointerException("demo");
        }
        catch(NullPointerException e)
        {
            System.out.println("Caught inside fun().");
            throw e;
        }
    }

    public static void main(String args[])
    {
        try
        {
            fun();
        }
        catch(NullPointerException e)
        {
            System.out.println("Caught in main.");
        }
    }
}
```

OUTPUT:

```
Caught inside fun().
Caught in main.
```

e.

```
class ExTy {
    public static void main(String[] args)throws InterruptedException
    {
        Thread.sleep(1000);
        System.out.println("Hello World");
    }
}
```

OUTPUT:

```
Hello World
```

f.

```
import java.util.*;
class VotingApplication extends Exception{
    String str;
    VotingApplication(String str){
        this.str = str;
    }
    public String toString(){
        return str;
    }
}
class ExTy {
    static void validatee(int age) throws VotingApplication{
        if(age>=18){
            System.out.println("Welcome to voting");
        }
        else{
            throw new VotingApplication("Not alowed");
        }
    }
    public static void main(String[] args)throws InterruptedException
    {
        int age;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter your age");
        age = in.nextInt();
        try{
            validatee(age);
        }
        catch(VotingApplication e){
            System.out.println("As age is less than 18 "+e);
        }
    }
}
```

OUTPUT:

```
Enter your age
17
As age is less than 18 Not allowed
```

g.

```
import java.util.*;
class ExTy {
    public static void main(String[] args)
    {
        int no1, no2, res;
        System.out.println("Welcome");
        Scanner in = new Scanner(System.in);
        try{
            int arr[] = new int[1];
            System.out.println(arr[1]);
            no1 = 20; no2 = 0;
            res = no1/no2;
            System.out.println("Result is "+res);
        }
        catch(Exception ob){
            System.out.println("Generic Exception");
        }
    }
}
```

OUTPUT:

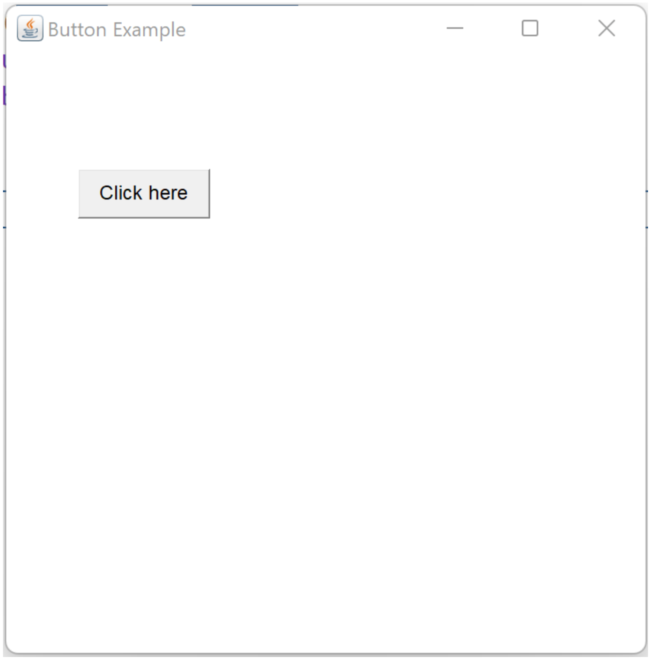
```
Welcome
Generic Exception
```

ASSIGNMENT 11

1) Write a Java program to create a button by implementing AWT(Abstract Window Toolkit).

```
// AWT example
import java.awt.*;
class AWT{
    public static void main(String args[]){
        Frame fr = new Frame("Button Example");
        Button bt = new Button("Click here");
        bt.setBounds(50, 100, 80, 30);
        fr.add(bt);
        fr.setSize(400, 400);
        fr.setLayout(null);
        fr.setVisible(true);
    }
}
```

OUTPUT:



2) Write a Java program to create a Calculator by implementing AWT.

```
import java.awt.*;
import java.awt.event.*;

class MyCalc extends WindowAdapter implements ActionListener {
    Frame f;
    Label l1;
    Button b1, b2, b3, b4, b5, b6, b7, b8, b9, b0;
    Button badd, bsub, bmult, bdiv, bmod, bcalc, bclr, bpts, bneg, bback;
    double xd;
    double num1, num2, check;
```

```java
    MyCalc() {
        f = new Frame("MY CALCULATOR");
        // INSTANTIATING COMPONENETS
        l1 = new Label();
        l1.setBackground(Color.LIGHT_GRAY);
        l1.setBounds(50, 50, 260, 60);

        b1 = new Button("1");
        b1.setBounds(50, 340, 50, 50);
        b2 = new Button("2");
        b2.setBounds(120, 340, 50, 50);
        b3 = new Button("3");
        b3.setBounds(190, 340, 50, 50);
        b4 = new Button("4");
        b4.setBounds(50, 270, 50, 50);
        b5 = new Button("5");
        b5.setBounds(120, 270, 50, 50);
        b6 = new Button("6");
        b6.setBounds(190, 270, 50, 50);
        b7 = new Button("7");
        b7.setBounds(50, 200, 50, 50);
        b8 = new Button("8");
        b8.setBounds(120, 200, 50, 50);
        b9 = new Button("9");
        b9.setBounds(190, 200, 50, 50);
        b0 = new Button("0");
        b0.setBounds(120, 410, 50, 50);
        bneg = new Button("+/-");
        bneg.setBounds(50, 410, 50, 50);
        bpts = new Button(".");
        bpts.setBounds(190, 410, 50, 50);
        bback = new Button("back");
        bback.setBounds(120, 130, 50, 50);

        badd = new Button("+");
        badd.setBounds(260, 340, 50, 50);
        bsub = new Button("-");
        bsub.setBounds(260, 270, 50, 50);
        bmult = new Button("*");
        bmult.setBounds(260, 200, 50, 50);
        bdiv = new Button("/");
        bdiv.setBounds(260, 130, 50, 50);
        bmod = new Button("%");
        bmod.setBounds(190, 130, 50, 50);
        bcalc = new Button("=");
        bcalc.setBounds(245, 410, 65, 50);
        bclr = new Button("CE");
        bclr.setBounds(50, 130, 65, 50);

        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
        b6.addActionListener(this);
        b7.addActionListener(this);
        b8.addActionListener(this);
        b9.addActionListener(this);
        b0.addActionListener(this);

        bpts.addActionListener(this);
        bneg.addActionListener(this);
        bback.addActionListener(this);

        badd.addActionListener(this);
        bsub.addActionListener(this);
        bmult.addActionListener(this);
        bdiv.addActionListener(this);
        bmod.addActionListener(this);
        bcalc.addActionListener(this);
        bclr.addActionListener(this);

        f.addWindowListener(this);
        // ADDING TO FRAME
        f.add(l1);
        f.add(b1);
        f.add(b2);
        f.add(b3);
        f.add(b4);
        f.add(b5);
        f.add(b6);
        f.add(b7);
        f.add(b8);
        f.add(b9);
        f.add(b0);

        f.add(badd);
        f.add(bsub);
        f.add(bmod);
        f.add(bmult);
        f.add(bdiv);
        f.add(bmod);
        f.add(bcalc);

        f.add(bclr);
        f.add(bpts);
        f.add(bneg);
        f.add(bback);

        f.setSize(360, 500);
        f.setLayout(null);
        f.setVisible(true);
    }

    // FOR CLOSING THE WINDOW
    public void windowClosing(WindowEvent e) {
        f.dispose();
    }
```

```
public void actionPerformed(ActionEvent e) {
    String z, zt;
    // NUMBER BUTTON
    if (e.getSource() == b1) {
        zt = l1.getText();
        z = zt + "1";
        l1.setText(z);
    }
    if (e.getSource() == b2) {
        zt = l1.getText();
        z = zt + "2";
        l1.setText(z);
    }
    if (e.getSource() == b3) {
        zt = l1.getText();
        z = zt + "3";
        l1.setText(z);
    }
    if (e.getSource() == b4) {
        zt = l1.getText();
        z = zt + "4";
        l1.setText(z);
    }
    if (e.getSource() == b5) {
        zt = l1.getText();
        z = zt + "5";
        l1.setText(z);
    }
    if (e.getSource() == b6) {
        zt = l1.getText();
        z = zt + "6";
        l1.setText(z);
    }
    if (e.getSource() == b7) {
        zt = l1.getText();
        z = zt + "7";
        l1.setText(z);
    }
    if (e.getSource() == b8) {
        zt = l1.getText();
        z = zt + "8";
        l1.setText(z);
    }
    if (e.getSource() == b9) {
        zt = l1.getText();
        z = zt + "9";
        l1.setText(z);
    }
    if (e.getSource() == b0) {
        zt = l1.getText();
        z = zt + "0";
        l1.setText(z);
    }
```

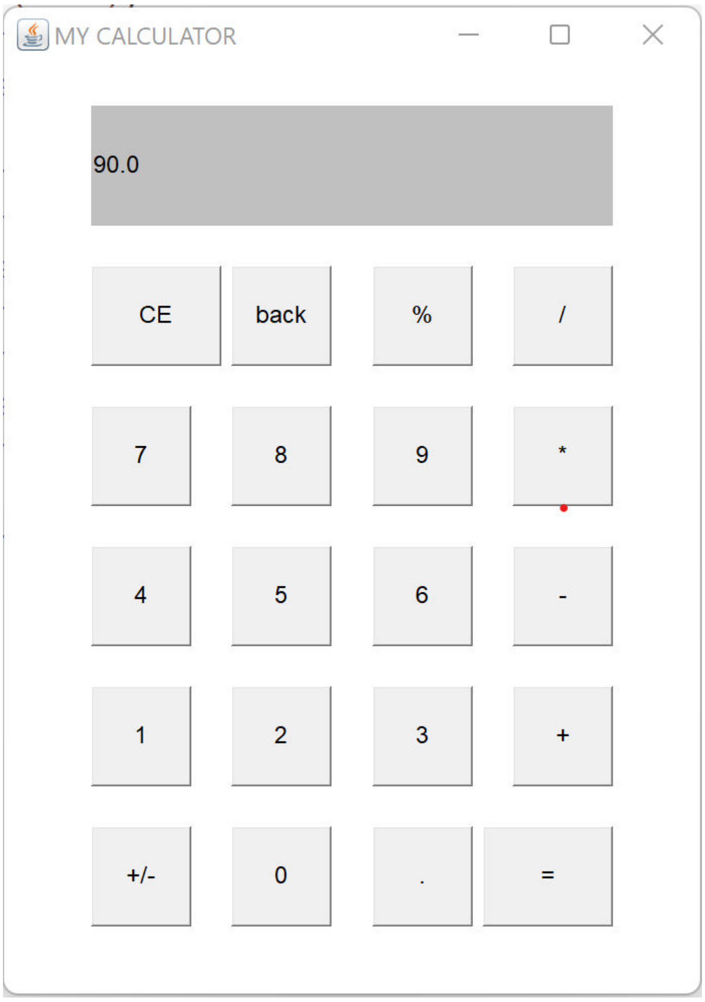
```
if (e.getSource() == bpts) { // ADD DECIMAL PTS
    zt = l1.getText();
    z = zt + ".";
    l1.setText(z);
}
if (e.getSource() == bneg) { // FOR NEGATIVE
    zt = l1.getText();
    z = "-" + zt;
    l1.setText(z);
}

if (e.getSource() == bback) { // FOR BACKSPACE
    zt = l1.getText();
    try {
        z = zt.substring(0, zt.length() - 1);
    } catch (StringIndexOutOfBoundsException f) {
        return;
    }
    l1.setText(z);
}
// AIRTHMETIC BUTTON
if (e.getSource() == badd) { // FOR ADDITION
    try {
        num1 = Double.parseDouble(l1.getText());
    } catch (NumberFormatException f) {
        l1.setText("Invalid Format");
        return;
    }
    z = "";
    l1.setText(z);
    check = 1;
}
if (e.getSource() == bsub) { // FOR SUBTRACTION
    try {
        num1 = Double.parseDouble(l1.getText());
    } catch (NumberFormatException f) {
        l1.setText("Invalid Format");
        return;
    }
    z = "";
    l1.setText(z);
    check = 2;
}
if (e.getSource() == bmult) { // FOR MULTIPLICATION
    try {
        num1 = Double.parseDouble(l1.getText());
    } catch (NumberFormatException f) {
        l1.setText("Invalid Format");
        return;
    }
    z = "";
    l1.setText(z);
    check = 3;
```

```
        }
        if (e.getSource() == bdiv) { // FOR DIVISION
            try {
                num1 = Double.parseDouble(l1.getText());
            } catch (NumberFormatException f) {
                l1.setText("Invalid Format");
                return;
            }
            z = "";
            l1.setText(z);
            check = 4;
        }
        if (e.getSource() == bmod) { // FOR MOD/REMAINDER
            try {
                num1 = Double.parseDouble(l1.getText());
            } catch (NumberFormatException f) {
                l1.setText("Invalid Format");
                return;
            }
            z = "";
            l1.setText(z);
            check = 5;
        }
        // RESULT BUTTON
        if (e.getSource() == bcalc) {
            try {
                num2 = Double.parseDouble(l1.getText());
            } catch (Exception f) {
                l1.setText("ENTER NUMBER FIRST ");
                return;
            }
            if (check == 1)
                xd = num1 + num2;
            if (check == 2)
                xd = num1 - num2;
            if (check == 3)
                xd = num1 * num2;
            if (check == 4)
                xd = num1 / num2;
            if (check == 5)
                xd = num1 % num2;
            l1.setText(String.valueOf(xd));
        }
        // FOR CLEARING THE LABEL and Memory
        if (e.getSource() == bclr) {
            num1 = 0;
            num2 = 0;
            check = 0;
            xd = 0;
            z = "";
            l1.setText(z);
        }

    }
    // MAIN METHOD where objects of MyCalc is instantaiated
    public static void main(String args[]) {
        new MyCalc();
    }
}
```

OUTPUT:



NAME – PRIYANSHU MALLICK
SIC – 21BCSF11
ROLL. NO – 30
SEC – **B, GROUP** – B2
BRANCH - CSE

ASSIGNMENT 12

1) Write a java program to create two threads. First thread should find the square of the number, second thread should find the sum of the digits of the squared number.

```java
import java.util.Scanner;

class Global {
    public static int num;

    public static void update(int n) {
        num = n;
    }
}

class Child1 extends Thread {
    Child1() {
        start();
    }

    public void run() {
        int n = Global.num;
        Global.update(n * n);
    }
}

class Child2 extends Thread {
    Child2() {
        start();
    }

    public void run() {
        int n = Global.num;
        int r, sum = 0;
        while (n > 0) {
            r = n % 10;
            n = n / 10;
            sum += r;
        }
        Global.update(sum);
    }
}

class GlobalDemo {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number:");
        int n = sc.nextInt();
        System.out.print("Sum of the digits of square of " + n + "=");
        Global.update(n);
        Child1 c1 = new Child1();
        Child2 c2 = new Child2();
        try {
            c1.join();
            c2.join();
        } catch (InterruptedException e) {
            System.out.print(e);
        }
        System.out.println(Global.num);
    }
}
```

OUTPUT:

```
Enter a number:
5
Sum of the digits of square of 5=7
```

2) Write a java program that will compute product of two vector (1D array) using multithreading. The program should read two vectors (of same size) from the user. First thread should multiply the corresponding elements present in the odd index position and second thread should multiply the corresponding elements present in the even index position. Main thread should display the result.

```
import java.util.Scanner;

class Global {
    public static int R[];

    Global(int s) {
        R = new int[s];
        for (int i = 0; i < s; i++)
            R[i] = 0;
    }

    public static void update(int index, int n) {
        R[index] = n;
    }
}

class Child1 extends Thread {
    int V1[], V2[];
    int size;

    Child1(int V1[], int V2[], int size) {
        this.V1 = V1;
        this.V2 = V2;
        this.size = size;
        start();
    }

    public void run() {
        try {
            for (int i = 1; i < size; i += 2) {
                Global.update(i, V1[i] * V2[i]);
                Thread.sleep(500);
            }
        } catch (InterruptedException e) {
            System.out.println("Child 1 interrupted.");
        }
    }
}

class Child2 extends Thread {
    int V1[], V2[];
    int size;

    Child2(int V1[], int V2[], int size) {
        this.V1 = V1;
        this.V2 = V2;
        this.size = size;
        start();
    }

    public void run() {
        try {
            for (int i = 0; i < size; i += 2) {
                Global.update(i, V1[i] * V2[i]);
                Thread.sleep(500);
            }
        } catch (InterruptedException e) {
            System.out.println("Child 2 interrupted.");
        }
    }
}

class VectorDemo {
    public static void main(String args[]) {
        int size;
        int m[], n[];
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the size of the vectors:");
        size = sc.nextInt();
        m = new int[size];
        n = new int[size];
        new Global(size);
        System.out.println("Enter the first vector:");
        for (int i = 0; i < size; i++)
            m[i] = sc.nextInt();
        System.out.println("Enter the second vector:");
        for (int i = 0; i < size; i++)
            n[i] = sc.nextInt();
        Child1 c1 = new Child1(m, n, 5);
        Child2 c2 = new Child2(m, n, 5);
        try {
            c1.join();
            c2.join();
        } catch (InterruptedException e) {
            System.out.print(e);
        }
        System.out.println("Product of the vectors:");
        for (int i = 0; i < size; i++) {
            System.out.print(Global.R[i] + "\t");
        }
    }
}
```

OUTPUT:

```
Enter the size of the vectors:
5
Enter the first vector:
1 2 3 4 5
Enter the second vector:
5 4 3 2 1
```

```
Product of the vectors:
5    8    9    8    5
```

3) Write a simple Java thread program to compute the sum of n natural numbers. The program should read the number of threads m and value of n from the user. Each of the threads should add its share of assigned number to a global variable. When all the threads are done, the global variable should contain the result. The program should use a Synchronized block to make sure that only one thread is updating the global variable at a given time

```java
import java.util.Scanner;

class Global {
    public static int sum;

    public static synchronized void add(int n) {
        sum += n;
    }
}

class ChildThread extends Thread {
    int m, n;

    ChildThread(int n1, int n2) {
        m = n1;
        n = n2;
        start();
    }

    public void run() {
        for (int i = m; i <= n; i++)
            Global.add(i);
    }
}

class SumDemo {
    public static void main(String args[]) {
        int m, n, i;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the value of n:");
        n = sc.nextInt();
        System.out.println("Enter the number of threads:");
        m = sc.nextInt();
        ChildThread ct[] = new ChildThread[m];
        for (i = 0; i < m - 1; i++)
            ct[i] = new ChildThread((n / m) * i + 1, (n / m) * (i + 1));
        ct[i] = new ChildThread((n / m) * i + 1, n);
        try {
            for (i = 0; i < m; i++)
                ct[i].join();
        } catch (InterruptedException e) {
            System.out.print(e);
        }
        System.out.println("Sum of the " + n + "numbers:" + Global.sum);
    }
}
```

```
OUTPUT:
Enter the value of n:
5
Enter the number of threads:
3
Sum of the 5numbers:15
```

4) Write a Java thread program to search the minimum number in a given array. The program should read the number of elements in the array, number of threads to be created and the array elements from the user. Each thread should find minimum element in an assigned block of elements and compare to global minimum element. When all the threads are done, the global variable should contain the minimum element. It should use a Synchronized block to make sure that only one thread is updating the global minimum variable at any given time

```
import java.util.Scanner;

class Global {
    public static int min, tno;

    public static synchronized void update(int n) {
        if (tno == 0)
            min = n;
        else {
            if (n < min)
                min = n;
        }
        tno++;
    }
}

class ChildThread extends Thread {
    int m, n, arr[];

    ChildThread(int n1, int n2, int a[]) {
        m = n1;
        n = n2;
        arr = new int[n2 - n1 + 1];
        for (int i = 0; i < n - m + 1; i++)
            arr[i] = a[m + i];
        start();
    }

    public void run() {
        int small = arr[0];
        for (int i = 1; i < n - m + 1; i++) {
            if (arr[i] < small)
                small = arr[i];
        }
        System.out.println(m + "\t" + n + "\t" + small);
        Global.update(small);
    }
}

class MinDemo {
    public static void main(String args[]) {
        int m, n, i, arr[];
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of elements of the array:");
        n = sc.nextInt();
        arr = new int[n];
        System.out.println("Enter the number of threads to be created:");
        m = sc.nextInt();
        ChildThread ct[] = new ChildThread[m];
        System.out.println("Enter " + n + " elements:");
        for (i = 0; i < n; i++)
            arr[i] = sc.nextInt();
        for (i = 0; i < m - 1; i++)
            ct[i] = new ChildThread((n / m) * i, (n / m) * (i + 1) - 1, arr);
        ct[i] = new ChildThread((n / m) * i, n - 1, arr);
        try {
            for (i = 0; i < m; i++)
                ct[i].join();
        } catch (InterruptedException e) {
            System.out.print(e);
        }
        System.out.println("Minimum element of the array : " + Global.min);
    }
}
```

OUTPUT:

```
Enter the number of elements of the array:
5
Enter the number of threads to be created:
3
Enter 5 elements:
1 2 3 4 5
2       4       3
1       1       2
0       0       1
Minimum element of the array : 1
```

NAME – PRIYANSHU MALLICK
SIC – 21BCSF11
ROLL. NO – 30
SEC – B
GROUP – B2
BRANCH - CSE