

## Java Program Note

### Program Name: Find Largest Number Between Three Number

```
import java.util.Scanner;
public class largest {
    public static void main(String[ ] args) {
        int a, b, c;
        Scanner x = new Scanner(System.in);
        System.out.println("Enter the first number:");
        a = x.nextInt();
        System.out.println("Enter the second number:");
        b = x.nextInt();
        System.out.println("Enter the third number:");
        c = x.nextInt();
        System.out.println("Result:");
        if (a > b && a > c){
            System.out.println( a + ", first number is the maximum number.");
        }
        else if (b > a && b >= c){
            System.out.println( b + ", second number is the maximum number.");
        }
        else {
            System.out.println( c + ", third number is the maximum number.");
        }
    }
}
```

### Program Name: Sum Of Series 2+4+6+.....+N

```
import java.util.Scanner;
public class Series {
    public static void main(String[ ] args) {
        Scanner s =new Scanner (System.in);
        System.out.print("enter the value of n: ");
        int n = s.nextInt();
        int i, sum = 0;
        for (i = 2; i <= n; i+=2) {
            sum += i;
        }
        System.out.println("sum of series: "+sum);
    }
}
```

### Program Name: 1 To (100 / N / 25) Print Prime Number Using For Loop

```
import java.util.Scanner;
public class PrimeNumber {
```

```

public static void main(String[ ] args) {
    Scanner s =new Scanner (System.in);
    System.out.print("enter the value of n: ");
    int n = s.nextInt();
    System.out.println("Prime Number = ");
    int i,j;
    for (i =2; i<=n;i++)
    {
        for(j=2;j<=i;j++)
        {
            if (i%j==0)
                break;
        }
        if (i==j)
            System.out.print(i+" ");
    }
}

```

### Program Name: Finding The Root Of Quadratic Equation

```

import java.util.Scanner;
public class QEquation {
    public static void main(String[ ] args) {
        Scanner s = new Scanner(System.in);
        double a,b,c;
        System.out.print("Enter the value of a: ");
        a = s.nextDouble();
        System.out.print("Enter the value of b: ");
        b = s.nextDouble();
        System.out.print("Enter the value of c: ");
        c = s.nextDouble();
        double d= (b*b) - (4*a*c);
        if (d> 0)
        {
            double x1=(-b+Math.sqrt(d))/(2*a);
            double x2=(-b-Math.sqrt(d))/(2*a);
            System.out.println("The roots are real, 1st Root is: " + x1 + " and 2nd Root is: "
+ x2);
        }
        else if (d==0)
        {
            double x = (-b)/(2*a);
            System.out.println("The root is: " + x + " and unique");
        }
        else

```

```

    {
        System.out.println("The roots are not real and imaginary");
    }
}

```

### Program Name: Print The First 10 / N Terms Of The Fibonacci Series

```

import java.util.Scanner;
public class Fibonacci {
    public static void main(String[] args) {
        Scanner s =new Scanner (System.in);
        System.out.print("enter the value of n: ");
        int n = s.nextInt();
        int a = 0, b = 1;
        System.out.println("Fibonacci Series till first 10 terms:");
        for (int i = 1; i <= n; ++i) {
            System.out.print(a + " ");
            int f = a + b;
            a = b;
            b = f;
        }
    }
}

```

### Program Name: Method Overloading

```

public class Sum {
    public void GetAdd(int x, int y) {
        int a = x + y;
        System.out.println("Sum Of 2 Number = " + a);
    }
    public void GetAdd(double x, double y) {
        double b = x + y;
        System.out.println("Sum Of Decimal Number = " + b);
    }
    public void GetAdd(int x, int y, int z) {
        int c = x + y + z;
        System.out.println("Sum Of 3 Number = " + c);
    }
}

class Overloading{
    public static void main(String[ ] args){
Sum S = new Sum();
S.GetAdd(2,4);
S.GetAdd(2,4, 6);
S.GetAdd(2.4509,3.359);
    }
}

```

```
}
```

### **Program Name: Constructor Method**

```
public class Constructor {  
    int a,b;  
    Double x,y;  
    public Constructor()  
    {  
        a= 530264;  
        b= 530263;  
        x= 3.94;  
        y= 3.68;  
        System.out.println("Student ID = " + a);  
        System.out.println("Student Result = " +x);  
        System.out.println("Student ID = " + b);  
        System.out.println("Student Result = " +y);  
    }  
    public static void main (String [ ] args){  
        Constructor x = new Constructor();  
    }  
}
```

### **Program Name: Constructor Overloading / Copy Constructor**

```
public class CopyConstructor {  
    int base, height;  
    public CopyConstructor(int b, int h)  
    {  
        base= b;  
        height=h;  
    }  
    public CopyConstructor(CopyConstructor NEW)  
    {  
        base= NEW.base;  
        height= NEW.height;  
    }  
    public double GetArea(){  
        double Area;  
        Area = 0.5 * base* height;  
        return Area;  
    }  
    public static void main(String[] args) {  
        CopyConstructor x = new CopyConstructor(3,4);  
        CopyConstructor y = new CopyConstructor(x);  
        System.out.println("Parameterized Normal Constructor Result");  
        System.out.println("Area Of Triangle = " +x.GetArea());  
        System.out.println("After Using Copy Constructor Result");  
    }  
}
```

```
System.out.println("Area Of Triangle= " +y.GetArea());
```

```
}  
}  
Program Name: Triangle (Scalene) Area with “TRIANGLE” Name Class
```

```
import java.util.Scanner;  
public class TRIANGLE {  
    public static void main(String[] args) {  
        int a, b, c;  
        Scanner x = new Scanner(System.in);  
        System.out.println("Enter the first number:");  
        a = x.nextInt();  
        System.out.println("Enter the second number:");  
        b = x.nextInt();  
        System.out.println("Enter the third number:");  
        c = x.nextInt();  
        if (a+b>c && a+c>b && b+c>a ) {  
            double s = (a + b + c) / 2;  
            double Area = (Math.sqrt(s * (s - a) * (s - b) * (s - c)));  
            System.out.print("Area Of Scalene Triangle= " + Area);  
        }  
        else{  
            System.out.print("Area Of Scalene Triangle Is Not Possible");  
        }  
    }  
}
```

**Program Name: SuperClass & SubClass / Inheritance / Single Inheritance**

```
class Super {  
    public void triangle(int x, int y){  
        double a = 0.5*x*y;  
        System.out.println("Area Of triangle= "+a);  
    }  
}  
class Sub extends Super{  
    public void rectangle(int x, int y){  
        int b = x*y;  
        System.out.println("Area Of rectangle= "+b);  
    }  
    public void display(){  
        triangle(10,2);  
        rectangle(20,2);  
    }  
}  
public class inheritance {
```

```

    public static void main (String [ ] args){
        Sub s = new Sub();
        s.display();
    }
}

```

### Program Name: Multiple / Multilevel Inheritance

```

class tri {
    public void triangle(int x, int y){
        double a = 0.5*x*y;
        System.out.println("Area Of triangle= "+a);
    }
}

class rec extends tri {
    public void rectangle(int x, int y) {
        int b = x * y;
        System.out.println("Area Of rectangle= " + b);
    }
}

class add extends rec{
    public void addiction(int x, int y){
        int c = x*y;
        System.out.println("Addition Of Two Number= "+c);
    }
    public void display(){
        triangle(10,2);
        rectangle(20,2);
        addiction(5,10);
    }
}

public class MInheritance {
    public static void main (String [ ] args){
        add s = new add();
        s.display();
    }
}

```

### Program Name: Interface

```

interface Rectangle {
    public void getArea(int l, int b);
}

class RArea implements Rectangle {
    public void getArea(int l, int b) {
        int A= l*b;
        System.out.println("The area of the rectangle is " + A);
    }
}

```

```

}
class Main {
    public static void main(String[] args) {
        RArea x = new RArea();
        x.getArea(5, 6);
    }
}

```

### Program Name: MultiTread

```

class Multithreading implements Runnable {
    public void run()
    {
        try {
            System.out.println("Thread " + Thread.currentThread().getId() + " is
running");
        }
        catch (Exception e) {
            System.out.println("Exception is caught");
        }
    }
}

class mainMultithread {
    public static void main(String[] args)
    {
        int n = 5; // Number of threads
        for (int i = 0; i < n; i++) {
            Thread object = new Thread(new Multithreading());
            object.start();
        }
    }
}

```