

1.STUDENT DETAILS

```
import java.io.*;
import java.util.*;
public class Student{
    String name;
    String ph_no;
    int m1,m2,m3,m4,m5,total;
    float avg;
    void Student1(){
        inter();
        out();
    }
    void inter(){
        System.out.println("");
        System.out.print("ENTER THE STUDENT NAME:");
        Scanner A=new Scanner(System.in);
        name=A.nextLine();
        System.out.println("");
        System.out.print("Enter The Student Mobile Number: ");
        ph_no= A.nextLine();
        System.out.println("");
        System.out.print("Enter The Student Firts Mark:");
        m1=A.nextlnt();
        System.out.println("");
        System.out.print("Enter The Student Second Mark:");
        m2=A.nextlnt();
        System.out.println("");
        System.out.print("Enter The Student Third Mark:");
        m3=A.nextlnt();
        System.out.println("");
        System.out.print("Enter The Student Fourth Mark:");
        m4=A.nextlnt();
        System.out.println("");
        System.out.print("Enter The Student Fifth Mark:");
        m5=A.nextlnt();
    }
    void out(){
        total=m1+m2+m3+m4+m5;
        avg=total/5.0f;
        System.out.println("");
        System.out.println("-----");
        System.out.println("| STUDENT NAME: "+name+" ..");
        System.out.println("| STUDENT MOBILE NUMBER: "+ph_no+" ..");
        System.out.println("| STUDENT TOTAL MARK: "+total+" ..");
        System.out.println("| STUDENT AVERAGE MARK: "+avg+" ..");
        System.out.println("-----");
    }
    public static void main(String[] arg){
        Student s1=new Student();
        s1.Student1();
    }
}
```

2.CONSTRUCTOR OVERLOADING (A)

```
import java.util.Scanner;

class Shape {
    Shape(float length, float breadth) {
        float area= length * breadth;
        System.out.println("Area of Rectangle = "+ area);
    }

    Shape(floatside) {
        float area= side * side;
        System.out.println("Area of Square = "+ area);
    }

    Shape(double base, double height) {
        double area=0.5* base * height;
        System.out.println("Area of Triangle = "+ area);
    }
}

public class Area {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);

        System.out.print("Enter length of Rectangle: ");
        float l=sc.nextFloat();
        System.out.print("Enter breadth of Rectangle: ");
        float b=sc.nextFloat();
        new Shape(l, b);

        System.out.println();

        System.out.print("Enter side of Square: ");
        float s=sc.nextFloat();
        new Shape(s);

        System.out.println();

        System.out.print("Enter base of Triangle: ");
        double base=sc.nextDouble();
        System.out.print("Enter height of Triangle: ");
        double h=sc.nextDouble();
        new Shape(base, h);

        sc.close();
    }
}
```

2.METHOD OVERLOADING (B)

```
import java.util.Scanner;

class Shape {

    void area(float length, float breadth) {
        System.out.println("AREA OF THE RECTANGLE = "+ (length * breadth));
    }

    void area(float side) {
        System.out.println("AREA OF THE SQUARE = "+ (side * side));
    }

    void area(double base, double height) {
        System.out.println("AREA OF THE TRIANGLE = "+ (0.5* base * height));
    }
}

public class AREA {
    public static void main(String[] args) {
        Shape s=new Shape();
        Scanner R=new Scanner(System.in);

        System.out.println("ENTER THE LENGTH OF THE RECTANGLE:");
        float l=R.nextFloat();
        System.out.println("ENTER THE BREADTH OF THE RECTANGLE:");
        float b=R.nextFloat();
        s.area(l, b);

        System.out.println("ENTER THE SIDE OF THE SQUARE:");
        float side=R.nextFloat();
        s.area(side);

        System.out.println("ENTER THE HEIGHT OF THE TRIANGLE:");
        double h=R.nextDouble();
        System.out.println("ENTER THE BASE OF THE TRIANGLE:");
        double base=R.nextDouble();
        s.area(base, h);

        R.close();
    }
}
```

3 . COMPLEX NUMBER

```
public class Complex
{
    double real;
    double img;
    public Complex(double real,double img)
    {
        this.real=real;
        this.img=img;
    }
    public static void main(String[] args)
    {
        Complex n1=new Complex(2.3,4.5),
        n2=new Complex(3.4,5.0),
        temp;
        temp=add(n1,n2);
        System.out.println();
        System.out.printf("Sum =%.1f + %.1f\n",temp.real,temp.img);
        System.out.println("Total Sum : "+(temp.real+temp.img));
    }
    public static Complex add(Complex n1,Complex n2)
    {
        Complex temp=new Complex(0.0,0.0);
        temp.real=n1.real+n2.real;
        temp.img=n1.img+n2.img;
        return(temp);
    }
}
```

4 . SUPER CLASS METHOD

```
class Animal {
    String name;
    public Animal(String name) {
        this.name= name;
        System.out.println("Animal constructor called for: "+this.name);
    }

    public void output() {
        System.out.println("This is an animal named: " + name);
    }
}

class Dog extends Animal {
    String breed;

    public Dog(String name, String breed) {
        super(name);
        this.breed= breed;
        System.out.println("Dog constructor called for: "+this.name+"
("+this.breed+)");
    }

    public void output() {
        super.output();
        System.out.println("This is a dog of breed: "+ breed);
    }
}

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

        Dog myDog=new Dog("Buddy", "Golden Retriever");
        myDog.output();

    }
}
```

5 . INTERFACE DEMO

```
interface Demo {
    double PI=3.14159;
    double area(double radius);
}

class Circle implements Demo {
    public double area(double radius) {
        return PI * radius * radius;
    }
}

public class Calculate {
    public static void main(String[] args) {
        Circle myCircle=new Circle();
        double circleRadius=5.0;
        double calculatedArea=myCircle.area(circleRadius);
        System.out.println("The area of the circle with radius "+ circleRadius +" is:
"+ calculatedArea);
    }
}
```

6 . EXCEPTION HANDLING

```
import java.util.Scanner;
public class Exception
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        try
        {
            System.out.print("Enter size of array: ");
            int size=sc.nextInt();
            int arr[]=new int[size];
            System.out.println("Array of size "+ size +" created successfully.");
        }
        catch(NegativeArraySizeException e)
        {
            System.out.println("caught Exception: "+ e);
        }

        try
        {
            int arr[] = {10,20,30};
            System.out.println("Trying to access 4th Element...");
            System.out.println(arr[3]);
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("caught Exception: "+ e);
        }
    }
}
```

7 . SUM APPLET

```
import java.applet.Applet;
import java.awt.Graphics;

public class SumApplet extends Applet {
    int num1, num2, sum;
    public void init() {
        String s1=getParameter("a");
        String s2=getParameter("b");

        num1 =Integer.parseInt(s1);
        num2 =Integer.parseInt(s2);

        sum = num1 + num2;
    }
    public void paint(Graphics g) {
        g.drawString("First Number : "+ num1, 20, 40);
        g.drawString("Second Number: "+ num2, 20, 60);
        g.drawString("Sum = "+ sum, 20, 80);
    }
}
```

HTML

```
<html>
<body>
    <appletcode="SumApplet.class"width="300"height="200">
        <paramname="a"value="25">
        <paramname="b"value="15">
    </applet>
</body>
</html>
```


8 . STRING CLASS

```
class StringClass {
    public static void main(String[] args) {
        String str1="ANNAMALAI UNIVERSITY";
        String str2=new String("Java programming");

        System.out.println("String 1: "+ str1);
        System.out.println("String 2: "+ str2);
        System.out.println("String length: "+str1.length());

        String str3=str1.concat(str2);
        System.out.println("String concat: "+ str3);
        System.out.println("String equals: "+str1.equals("Annamalai"));

        System.out.println("Index of 'Annamalai' in str1:
"+str1.indexOf("ANNAMALAI"));
        System.out.println("str1 in Uppercase: "+str1.toUpperCase());
        System.out.println("str2 in Lowercase: "+str2.toLowerCase());

        System.out.println("str1 with 'N' replaced by 'n': "+str1.replace('N', 'n'));
        System.out.println("str1 starts with 'Annam': "+str1.startsWith("Annam"));
        System.out.println("str1 ends with 'UNIVERSITY':
"+str1.endsWith("UNIVERSITY"));
    }
}
```

9 . TCP SERVER (A)

```
import java.io.*;
import java.net.*;

public class Server {
    public static void main(String[] args) {
        try {
            ServerSocket ss=new ServerSocket(5000);
            System.out.println("Server started. Waiting for client...");

            Socket s=ss.accept();
            System.out.println("Client connected!");

            BufferedReader in=new BufferedReader(new
InputStreamReader(s.getInputStream()));

            PrintWriter out=new PrintWriter(s.getOutputStream(), true);

            String clientMessage=in.readLine();
            System.out.println("Client says: "+ clientMessage);

            out.println("Hello from Server! Message received.");

            in.close();
            out.close();
            s.close();
            ss.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

TCP CLIENT(B)

```
import java.io.*;
import java.net.*;

public class Client {
    public static void main(String[] args) {
        try {
            String serverIP="localhost";
            int port=5000;

            Socket s=new Socket(serverIP, port);
            System.out.println("Connected to server.");

            BufferedReader in=new BufferedReader(new
InputStreamReader(s.getInputStream()));
            PrintWriter out=new PrintWriter(s.getOutputStream(), true);

            out.println("Hello Server! This is the Client.");

            String serverMessage=in.readLine();
            System.out.println("Server says: "+ serverMessage);

            in.close();
            out.close();
            s.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```