```java
public class Geomatry_Units {
    1 usage
    public static double Calculate_Area(double radius) {
        return Math.PI * (radius * radius);
    }

    1 usage
    public static double Calculate_Area(double length, double width) {
        return length * width;
    }

    1 usage
    public static double Calculate_perimeter(double radius) {
        return 2 * Math.PI * radius;
    }

    1 usage
    public static double Calculate_perimeter(double length, double width) {
        return (length + width) * 2;
    }

}
```

Geomatry_Units

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\In
6361.725123519331
31.41592653589793
20.0
20.0
8.0

Process finished with exit code 0
```

src > Geomatry_Units

```java
    public static double Calculate_volume(double side) {
        return side * side * side;

    }

    public static void main(String[] args) {
        double circle=Calculate_Area( radius: 45);
        double rec=Calculate_Area( length: 4, width: 5);
        double circleP=Calculate_perimeter( radius: 5);
        double rectp=Calculate_perimeter( length: 4, width: 6);
        double cube=Calculate_volume( side: 2);
        System.out.println(circle);
        System.out.println(circleP);
        System.out.println(rec);
        System.out.println(rectp);
        System.out.println(cube);

    }


}
```

Geomatry_Units ×

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program File
6361.725123519331
31.41592653589793
20.0
20.0
8.0

Process finished with exit code 0
```

rc > Geomatry_Units

```
    {
        this.account_num=account_num;
        this.Holder_name=holder_name;
        this.Balance=Balance;
    }
    1 usage
    public void transfer_Amount(Bank account,double balance){
        if(balance>0&& this.Balance>=balance) {
            this.Balance=Balance-balance;
            account.Balance+=balance;
            System.out.println("trasnfer succesfull");
        }
        else{
            System.out.println("Insufficent amount");
        }

    }
    3 usages
    public void display()
    {
        System.out.println("account holder name"+Holder_name);
        System.out.println("account num"+account_num);
        System.out.println("account balance"+Balance);
    }

    public static void main(String[] args) {
        Bank b1=new Bank( account_num: 5656, holder_name: "yousuf", Balance: 1000);
        Bank b2=new Bank( account_num: 9898, holder_name: "Amar", Balance: 500);

        b1.display();
        b2.display();
        b1.transfer_Amount(b2, balance: 500);
        b2.display();
```

Geomatry_Units ×

```java
    1 usage
    public BMI_anlyzer(double weight , double height)
    {
        this.weight=weight;
        this.height=height;
    }
    1 usage
    public double calculateBMI()
    {
        return (weight)/(height*height)*703;

    }
    1 usage
    public void Findstatus(double bmi) {
        if (bmi <= 18.5) {
            System.out.println("underweight");
        } else if (bmi < 24.9 & bmi >= 18.5) {
            System.out.println("normal");
        } else if (bmi < 29.9 & bmi > 25.0) {
            System.out.println("overweight");
        } else {
            System.out.println("obese");
        }
     }

    public static void main(String[] args) {
        BMI_anlyzer b=new BMI_anlyzer( weight: 63, height: 163);
        double bmi= (b.calculateBMI());
        System.out.println(bmi);
        b.Findstatus(bmi);

    }}
```

```java
public class Factorial {
    public static void main(String[] args) {
        int number=5;
        int result=factorial(number);
        System.out.println(result);


    }
    2 usages
    public static int factorial(int number)
    {
        if (number==0)
        {
            return 1;
        }
        else{
            return number * factorial( number: number-1);
        }
    }

}
```

```java
public class Re_power {

2 usages
public static int recursive(int base,int exponent)
{
    if(exponent==0)
    {
        return 1;
    }
    else{
        return  base*recursive(base,  exponent: exponent-1);
    }
}

    public static void main(String[] args) {
    int base=4;
    int exponent=4;

    int result=recursive(base,exponent);
        System.out.println(result);

    }



}
```

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


    1 usage
    public int sumeven(int a) {

        if(a % 2 == 0) {
         return a;}

        return 0;
    }

    1 usage
    public int sumodd(int b) {

        if (b % 2 != 0) {
            return b;
        }

        return 0;
    }

    public static void main(String []args){
        Compute c= new Compute();
        Scanner sc=new Scanner(System.in);
        int input = 0;

        int sum=0;
        int sum1=0;

        while(input >=0){
            System.out.println("enter the value");
```