

1. Write a program to overload the method print that prints sum of n natural numbers when one variable is passed, & prints the prime no. in a given range when 2 parameters are passed.

Class Overload.

```
{  
    void print(int n)  
    {  
        int sum=0;  
        for(int i=1;i<=n;i++)  
        {  
            sum = sum+i;  
        }  
        System.out.println ("Sum of "+n+" natural nos is "+sum);  
    }  
  
    void print(int m, int n)  
    {  
        System.out.println ("Prime nos in the range are ");  
        for(int i=m; i<=n; i++)  
        {  
            int flag=0;  
            for (int j=2;j<=i/2;j++)  
            {  
                if(i%j==0)  
                {  
                    flag=1;  
                    break;  
                }  
            }  
            if(flag == 0)  
                System.out.println(i);  
        }  
    }  
}  
  
Class Overload Demo  
{  
    public static void main (String [] args)  
    {
```

```
Overload o=new Overload ();  
o.print(5);  
o.print(7,13);
```

Output :

Enter n 5

Enter range 7 13

Sum =15

Prime numbers are

7

11

13 .

2. Grocery.

```
Class Grocery {  
    String c_name;  
    String c_ph;  
    double total;  
  
    Grocery(string c_name, string c_ph){  
        this.c_name = c_name;  
        this.c_ph = c_ph;  
    }  
  
    void calc(double q_dal, double q_pulses, double q_sugar){  
        total = q_dal * 100 + q_pulses * 80 + q_sugar * 50;  
    }  
  
    void display(){  
        System.out.println("Name " + " " + "Phone no " + " " + "Total");  
        System.out.println(c_name + " " + c_ph + " " + total);  
        System.out.println();  
    }  
}
```

Class Demo {

```
public static void main(string[] args){  
    grocery g1 = new grocery("Rama", "8060302010");  
    grocery g2 = new grocery("Shama", "7689632510");  
    grocery g3 = new grocery("Bhama", "9632587412");  
    g1.calc(2, 2, 1);  
    g1.display();  
    g2.calc(3, 5, 2);  
    g2.display();  
    g3.calc(1, 1, 0.5);  
    g3.display();  
}
```

Output:

Name	Phone number	Total
Rama	123456789	4100
Name	Phone number	Total
Shama	123456789	800.0
Name	Phone number	Total
Bhama	123456789	205.0

3. Calculate root & Quad Eqⁿ.

(1)

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

```
class Quad {
```

```
    int a, b, c;
```

```
    double root1, root2, d;
```

```
    Scanner s = new Scanner(System.in);
```

```
    void input()
```

```
{
```

```
    System.out.println("Quadratic Eq is in the form : ax^2 + bx + c");
```

```
    System.out.print("Enter a: ");
```

```
    a = s.nextInt();
```

```
    System.out.print("Enter b: ");
```

```
    b = s.nextInt();
```

```
    System.out.print("Enter c: ");
```

```
    c = s.nextInt();
```

```
}
```

```
void discriminant()
```

```
    d = (b * b) - (4 * a * c);
```

```
,
```

```
void calculateRoots()
```

```
    if (d > 0)
```

```
        System.out.println("Roots are real & unequal");
```

```
        root1 = (-b + Math.sqrt(d)) / (2 * a);
```

```
        root2 = (-b - Math.sqrt(d)) / (2 * a);
```

```
        System.out.println("First root is: " + root1);
```

```
        System.out.println("Second root is: " + root2);
```

```
}
```

```
else if (d == 0)
```

```
{
```

```
    System.out.println("Roots are real and equal");
```

```
    root1 = (-b + Math.sqrt(d)) / (2 * a);
```

```
    System.out.println("Roots" + root1);
```

```
}
```

```
else
```

```

System.out.println("No real solutions. Roots are imaginary");
double real = -b/(2*a);
double imaginary = Math.sqrt(-d)/(2*a);
System.out.println("The equation has two complex roots:");
    " + real + " + "i" + imaginary + "i and " + real + "-" + imaginary + "i";
}
}

class Main {
    public static void main(String[] args) {
        Quad q = new Quad();
        q.input();
        q.discriminant();
        q.calculateRoots();
    }
}

```

Output:

Quadratic Eqⁿ is in the form: $ax^2 + bx + c$

Enter a: 5

Enter b: 4

Enter c: 2

No real solutions. Roots are imaginary

The Eqⁿ has two complex roots:

$0.0 + 0.4898979i$ and $0.0 - 0.4898979i$

8A
22-12-23

(3)

①

```
import java.util.Scanner;  
class Books {  
    String name;  
    String author;  
    int price;  
    int numPages;  
    Books(String name, String author, int price, int numPages){  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages;  
    }  
    public String toString(){  
        return name + author + price + numPages;  
    }  
}
```

class Main
{

```
public static void main (String args[]){  
    Scanner s = new Scanner (System.in);
```

```
    int n;
```

```
    String name;
```

```
    String author;
```

```
    int price;
```

```
    int numPages;
```

```
    System.out.println("Enter no. of books:");
```

```
n = s.nextInt();
```

Books b[]

b = new Books[n];

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

System.out.println ("Book " + (i+1) + ": ");

System.out.print ("Enter name of book: ");

Name = s.next();

System.out.print ("Enter author: ");

Author = s.next();

System.out.print ("Enter price: ");

Price = s.nextInt();

System.out.print ("Enter no. of pages: ");

numPages = s.nextInt();

b[i] = new Books (name, author, price, numPages);

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

System.out.println ("Book: " + (i+1) + ": " + b[i]);

}

}

Output:

Enter no. of books: 2

Book 1:

Enter book: Jungle Book

Enter Author: Ruskin Bond

Enter price: 1000

Enter no. of pages: 500

Book 2:

Enter book: Tale of Fox

Enter Author: Birbal

Enter price: 900

Enter no. of pages: 400

Book 1:

Book name: Jungle Book

Author name: Ruskin Bond

Price: 1000

No. of pages: 500

Book 2:

Book name: Tale of Fox

Author name: Birbal

Price: 900

No. of pages: 400

②

②

```
import java.util.Scanner;
class Student {
    String usn, name;
    double [] marks = new double [6];
    void details() {
        Scanner s = new Scanner (System.in);
        System.out.print("Enter the USN:");
        usn = s.nextLine();
        System.out.print("Enter the name:");
        name = s.nextLine();
        System.out.print("Enter marks for 6 Subjects:");
        for (int i=0; i<6; i++) {
            System.out.print("Subject "+(i+1)+":");
            marks[i] = s.nextDouble();
        }
    }
}
```

```
double perc() {
    double totalmarks = 0;
    for (int i=0; i<marks.length; i++) {
        totalmarks = totalmarks + marks[i];
    }
    return (totalmarks / 6);
}
```

```
void display() {
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Percentage: " + perc() + "%");
}
```

Ques Main {

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

```
Scanner s = new Scanner(System.in);
```

```
System.out.print("Enter the no. of students: ");
int num = s.nextInt();
```

```
Student[] students = new Student[num];
```

```
for (int i=0; i<num; i++) {
```

```
System.out.print("Enter details for student ");
```

```
students[i] = new Student();
```

```
students[i].details();
```

```
System.out.print("Details of students: ");
for (int i=0; i<students.length; i++) {
```

```
    students[i].display();
}
```

```
}
```

```
}
```

Output:

Enter No. of students: 2

Enter details of student 1: Enter USN: 300

Enter the name: Nina

Enter marks for 6 subjects: subject 1: 78

subject 2: 89

subject 3: 67

subject 4: 56

subject 5: 78

subject 6: 45

Enter details of student 2: Enter the USN: 312

Enter the name: Jamie

Enter marks of 6 subjects: subject 1: 78

subject 2: 89

subject 3: 67

subject 4: 90

subject 5: 56

subject 6: 70

details of students: USN: 300 Name: Nina

Percentage: 68.8333%. USN: 312 Name: Jamie Percentage: 75.0%.

12/1/24

①

④

abstract class shape

```

int length;
int width;
public shape(int length, int width)
{
    this.length = length;
    this.width = width;
}

```

```
public abstract void printArea();
```

class rectangle extends shape

```

public rectangle(int length, int width)
{
    super(length, width);
}
public void printArea()
{
    float area = length * width;
    System.out.println("Rectangle Area: " + area);
}

```

class triangle extends shape

```

public triangle(int length, int breadth)
{
    super(length, width);
}
public void printArea()
{
    float area = 0.5 * length * breadth;
    System.out.println("Triangle Area: " + area);
}

```

```
class Circle extends Shape  
{  
    public Circle (int radius)  
    {  
        super (radius);  
    }  
    public void printArea()  
    {  
        float area = 3.14 * radius * radius;  
        System.out.println ("Circle Area: " + area);  
    }  
}
```

Public class Main()

```
{  
    public static void main (String args [])  
    {  
        Rectangle rectangle = new Rectangle (1, 2);  
        Triangle triangle = new Triangle (2, 4);  
        Circle circle = new Circle (5);  
        rectangle.printArea();  
        triangle.printArea();  
        circle.printArea();  
    }  
}
```

Op:
Rectangle Area: 2
Triangle Area: 4
Circle Area: 78.5

✓
8/1/24

(2)

```

import java.util.Scanner;
class Account {
    String cname;
    long accno;
    String acctype;
    double bal;
    public void deposit(double amt) {
        bal += amt;
        System.out.println("Deposited! Balance = " + bal);
    }
    public void display() {
        System.out.println("Balance = " + bal);
    }
}

```

```

class CurrAcct extends Account {
    double servicecharge; minbal;
    public void deposit(double amt) {
        super.deposit(amt);
        minbal();
    }
    public void display() {
        super.display();
        minbal();
    }
    public void minbal() {
        if (bal < minbal) {
            bal -= servicecharge;
            System.out.println("balance");
        }
    }
}

```

```

class Saver extends Account {
    double intrest rate;
    public void checkintrest() {
        double intrest = bal * intrest rate;
        bal += intrest;
        System.out.println("balance");
    }
}

```

```

    checkIntrest();
    super.displayBal();
public void withdraw(double amt)
{
    if(amt <= bal)
    {
        bal -= amt;
        System.out.println("Remaining Balance: " + balance);
    }
    else
    {
        System.out.println("Insufficient balance");
    }
}

public class Bank {
    public static void main (String [] args) {
        CurrentAcc ca = new CurrentAcc ("abc", 1234, 1000, 500, 10);
        ca.deposit(500, 0);
        ca.checkMinBalance();
        ca.displayBal ();
        SavingsAcc sa = new SavingsAcc ("xyz", 12367, 2000, 5);
        sa.DepositIntrest();
        sa.Withdraw(3000);
        sa.displayBal ();
    }
}

```

Output:
 Updated Balance: 1500
 Min Balance maintained
 Balance: 1500
 Interest deposited
 Updated Balance: 2100.0
 Remaining Balance: 1800.0
 Balance: 1800.0

⑦

Exception handling

```
class ErrorAge extends Exception {  
    ErrorAge(String msg) {  
        super(msg);  
    }  
}
```

```
class Father {  
    int age;  
    Father(int age) throws ErrorAge {  
        if (age < 0) {  
            throw new ErrorAge("Age should only be positive");  
        }  
        this.age = age;  
    }  
}
```

```
class Son extends Father {  
    int sonAge;  
    Son(int fatherAge, int sonAge) throws ErrorAge {  
        super(fatherAge);  
        if (sonAge >= fatherAge) {  
            throw new ErrorAge("Son's age should be less than father's");  
        }  
        this.sonAge = sonAge;  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        try {  
            Son son1 = new Son(40, 20);  
            System.out.println(son1.sonAge);  
            Son son2 = new Son(30, 35);  
            System.out.println(son2.sonAge);  
        } catch (ErrorAge e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

⑧

Threads

```
class Threads extends Thread {  
    private String sent;  
    private int intervals;  
    Threads (String sent, int intervals) {  
        this.sent = sent;  
        this.intervals = intervals;  
    }  
    public void run() {  
        try {  
            while (true) {  
                System.out.println(sent);  
                Thread.sleep(1000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println(e);  
        }  
    }  
}
```

```
class main {
```

```
    public static void main (String [] args) {  
        Threads thread1 = new Threads ("BMS College of Engineering", 10);  
        Threads thread2 = new Threads ("CSE", 2);  
        thread1.start();  
        thread2.start();  
    }  
}
```

Op:

BMS College of Engineering

CSE

BMS College of Engineering

CSE

/ / /

⑥ Packages

```
package CIE;
public class Student {
    public String usn;
    public String name;
    public int sem;
    public Student (String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

package CIE;
Public class Internals extends Student {
    public int[] internalMarks;
    public Internals (String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}

package SEE;
import CIE.Student;
public class External extends Student {
    public int[] marks;
    public External (String usn, String name, int sum, int[] marks) {
        super(usn, name, sum);
        this.marks = marks;
    }
}

import CIE.Student;
import SEE.External;
public class Main {
    public static void main (String [] args) {
        Student s1 = new Student ("IBM22CS312", "abc", 3);
        int[], SEE marks = {87, 81, 95, 91, 89};
        External s2 = new External ("IBM22CS333", "pqr", 3, SEE marks);
    }
}
```

```
S.O.Pln ("Student 1:");
S.O.Pln ("USN:" + s1.USN);
S.OPln ("Name :" + s1.name);
S.OPln ("Sem :" + s1.sem);
S.OPln ("Student 2:");
S.OPln ("USN;" + s2.USN);
S.OPln ("Name :" + s2.name);
S.OPln ("Sem :" + s2.sem);
}
```

O/p:

Student1: IBM22CS312

abc

3

Student2: IBM22CS333

Pvt

3