

Program-1 : Write a Java program to overload the method point that prints sum of n natural numbers when one variable is passed and prints the prime numbers in a given range when 2 parameters are passed

class overload {

void point($\text{int } n$) {

int sum = 0;

for ($\text{int } i=1; i \leq n; i++$)

{

sum = sum + i;

}

System.out.println("Sum of " + n + " natural numbers is " + sum);

void point($\text{int } m, \text{int } n$)

{

System.out.println("Prime Numbers in the range are : ");

for ($\text{int } i=m; i \leq n; i++$)

{

int flag = 0;

for ($\text{int } j=2; j \leq i; j++$)

{

if ($i \% j == 0$)

{

flag = 1;

break;

}

}

class overload@mo

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

```
Overload o = new Overload();
```

```
o.point(5);
```

```
o.point(7, 13);
```

Output:-

```
Enter n: 5
```

```
Enter orange 7 13
```

```
Sum: 15
```

```
Prime Numbers are - 7
```

2) Write a java program to create a class grocery

that has the variables - c-name and c-phone.

Create a method to accept 3 parameters to specify
quantity Dal, quantity of pulses and quantity of sugar

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-pulse, double q-sugar)
```

```
}
```

```
total = q-calc * 100 + q-pulses * 80 + q-sugar * 50;
```

```
y
```

```
void display()
```

```
}
```

```
System.out.println("Name " + " " + "phone no " + " " +  
"total");
```

```
System.out.println("Name " + " " + C-ph" + " " + total)
```

```
System.out.print();
```

```
}
```

```
class Demo
```

```
{
```

~~Enthält [no args] entf. und void Funktion~~

```
public static void main (String [Jargs])
```

```
{
```

```
grocery g1 = new grocery ("Rama", " 806030210")
```

```
grocery g2 = new grocery ("store " "Bhanu", " 786789")
```

```
g1.calc(2,2,1)
```

```
g1.display();
```

```
g2.calc(3,5,2)
```

```
g2.display();
```

```
g3.calc(1,1,0.5);
```

```
g3.display();
```

3

3

Output:-

Name	phonenumbers	Total
Rama	123456789	4100

Name	phonenumbers	Total
Shama	123456789	800.0

Name	phonenumbers	Total
Bhanu	123456789	265.5

3) Quadratic Equations [with all conditions]

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

```
class Quad {
```

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

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

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

```
void input {
```

System.out.println("Quadratic Eqn form
a) $a x^2 + b x + c = 0$ ");

$a \neq 0$

```

System.out.println ("Enter a:");
a = sc.nextInt();
System.out.println ("Enter b and c");
b = sc.nextInt();
c = sc.nextInt();
}

void discriminant() {
    d = (b*b) - (4*a*c);
}

void calculateRoots() {
    if (d > 0) {
        System.out.println ("Roots are real and unreal");
        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 unreal");
        root1 = (-b + math.sqrt(d)) / (2*a);
        System.out.println ("Roots " + root1);
    }
}

```

else

}

System.out.println("No real solution, Roots
are imaginary ");

double real = -b/(2*a);

double imaginary = math.sqrt(c-d)/(2*a);

System.out.println("The equation has 2

complex roots : "+real+"+"i"+imaginary+"

" and "+real+"-"+i"+imaginary+"i");

}

class main {

public static void main(String[] args)

{

quad q = new quad();

q. input();

q. discriminant

q. calculateRoots();

{

}

quadratic equations is in form of $ax^2 + bx + c$

Enter a: 5

Enter b: 10

Enter c: 30

No real solutions, Roots are imaginary

The equation has 2 complex roots: -1.0 + 2.23606797
and -1.0 - 2.236.

LAB-3

1) write a java program to create a class student with members USN, name, marks (6 subjects), include methods that take in students information and calculate percentage

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

```
class Student {
```

```
    String USN;
```

```
    String name;
```

```
    int marks[] = new int[6];
```

```
    void details
```

```
}
```

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

```
    System.out.println("Enter USN");
```

```
    USN = sc.nextLine();
```

```
System.out.println("Enter name:");
```

```
name = sc.nextLine();
```

```
System.out.println("Enter marks for 6 subjects");
```

```
for(int i=0; i<6; i++)
```

```
{
```

```
System.out.println("Subjects," + (i+1) + " ");
```

```
marks[i] = sc.nextInt();
```

```
{
```

```
}
```

```
Double percentage()
```

```
{
```

```
int total=0;
```

```
for(int i=0; i<6; i++)
```

```
{
```

```
total += marks[i];
```

```
{
```

```
double p = total/6;
```

```
return p;
```

```
y
```

```
void display()
```

```
{
```

```
System.out.println("In Student Details");
```

```
System.out.println ("USN" + usn);
```

```
System.out.println ("Name" + name);
```

```
System.out.println ("Marks:");
```

```
for (int i=0; i<6; i++)
```

```
{
```

```
    System.out.println ("Percentage:" + percentage() + ".");
```

```
}
```

```
System.out.println ("Subject" + (i+1) + " " + marks[i]);
```

```
}
```

```
class Lab1Student
```

```
{
```

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

```
{
```

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

```
    System.out.println ("Enter student Details Number");
```

```
    int n = sc.nextInt();
```

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

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

```
{
```

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

```
        System.out.println ("Enter details for student
```

```
        " + (i+1) + ".");
```

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

```
}
```

for (Students student : students)

3

student.display());

5

5

3

Output :-

Enter no of student :- 1

Enter details for student :-

Enter the name : tanush

Enter marks for 6 subjects;

Enter subject 1 : 90 subject 1 : 90

Enter subject 2 : 86 subject 2 : 86

Enter subject 3 : 85 subject 3 : 85

Enter subject 4 : 87 subject 4 : 87

Enter subject 5 : 89 subject 5 : 89

Enter subject 6 : 72 subject 6 : 72

Average = 89.6 %

✓ 11/24

2) create a book class Book that contains 4 members name, author, price, num-pages, include a constructor to set values for the members

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

```
class Books
```

```
String name;
```

```
String Author;
```

```
int price;
```

```
int num-pages;
```

```
Books (String Name, String author, int price, int numPages)
```

```
}
```

```
this.name = Name;
```

```
this.Author = Author;
```

```
this.numPage = numPages;
```

~~```
this.price = price;
```~~

```
}
```

```
public String toString()
```

```
{
```

```
String name; author; price; numPages;
```

```
name = "Book name : " + this.name + "\n";
```

```
numPages = "Number of pages
```

price = "price" + this.price + "n";

return name + author + numPages + price;

}

}

class main

{

public static void main (String [] args)

{

Scanner sc = new Scanner (System.in);

int n;

String name;

String author;

int price;

int numPages;

System.out.println ("Enter no. of Books");

n = sc.nextInt();

books b[] = new Books[n];

b = new Books[n];

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

{

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

```
System.out.println("Enter the name of book:");
```

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

```
System.out.println("Enter the author:");
```

```
author = s.next();
```

```
System.out.println("Enter the price");
```

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

```
System.out.println("Enter the number of pages");
```

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

```
numPages
```

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

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

3

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

y

y

```
Output:
```

Enter the number of books : 1

Book 1 :

Enter the name of the book : Atomic habit<sup>®</sup>

Enter the author of the book : James clear

Enter the price of the book : 950

Enter the number of pages : 320

Book 1 :

Book name : Atomic habit<sup>®</sup>

ବୁଦ୍ଧିମତ୍ତା କାହାର କାହାର

ପାଦିବା କାହାର

କାହାର କାହାର କାହାର

LAB- Program 5 19/1/24

Develop a Java program to create an abstract class named shape that contains two integers and an empty method named pointarea(). Provide three classes named Rectangle, Triangle, Circle such that each one of the classes extends the class shape. Each of the classes contain the method pointarea() that prints the area of the given shape.

import java.util.Scanner;

abstract class shape {

    double a, double b;

~~public~~ shape (double a, double b)

{

    this.a = a;

    this.b = b;

}

abstract void pointarea();

}

class rectangle extends shape {

    rectangle (double len, double wid)

{

        super(len, wid);

}

```
void pointarea()
```

```
}
```

```
double area = a * b;
```

```
System.out.println("Rectangle area is :" + area);
```

```
}
```

```
}
```

```
class triangle extends shape {
```

```
triangle(double base, double height)
```

```
{
```

```
super(base, height);
```

```
}
```

```
void pointarea() {
```

```
double area = 0.5 * a * b;
```

```
System.out.println("Triangle area : " + area);
```

```
}
```

```
}
```

```
class circle extends shape {
```

```
circle()
```

```
circle(double radius) {
```

```
super(radius, 0);
```

```
}
```

```
void pointarea() {
```

```
double area = 3.14 * r * r;
```

```
void pointArea()
```

```
}
```

```
double area = a * b;
```

```
System.out.println ("Rectangle area is :" + area);
```

```
}
```

```
}
```

```
class triangle extends shape {
```

```
triangle (double base, double height)
```

```
{
```

```
super (base, height);
```

```
}
```

```
void pointArea () {
```

```
double area = 0.5 * a * b;
```

```
System.out.println ("Triangle area : " + area);
```

```
}
```

```
}
```

```
class circle extends shape {
```

```
circle ()
```

```
circle (double radius) {
```

```
super (radius, 0);
```

```
}
```

```
void pointArea () {
```

```
double area = 3.14 * a * b
```

```
system.out.println("Circle area "+area);
}
```

```
{
 public class Rectangle {
```

```
 public static void main() {
```

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

```
 system.out.println("Enter length of rectangle");
```

~~length~~ ~~width~~

double

~~length~~ = sc.nextDouble();

```
 system.out.println("Enter width of rectangle");
```

double width = sc.nextDouble();

~~system.out.println("Enter the base of triangle");~~

double base = sc.nextDouble();

```
 system.out.println("Enter the height");
```

double height = sc.nextDouble();

```
 system.out.println("Enter the radius");
```

double radius = sc.nextDouble();

Rectangle rectangle = new Rectangle(length, width);

Triangle triangle = new Triangle(base, height);

~~Circle~~ circle = new Circle(radius);

rectangle . pointArea();  
triangle . pointArea();  
circle . pointArea();

3

3

3

Output :-

Enter length of Rectangle : 100

Enter width of Rectangle : 250

Enter base of triangle : 150

Enter height of triangle : 200

Enter radius of circle : 10

Rectangle Area : 25000.0

Triangle Area : 15000.0

Circle Area : 11304.0

19/1/24

## Program - 6

Write a program that demonstrates handling exception  
in inheritance ~~case~~ create a base class called father  
and derived class son which extends the base class  
inside the father class implement a constructor which takes  
the age and throws an exception WrongAge() when input  
< 0, in son class, implement a constructor that throws  
exception if son age > father's age

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

```
class wrongAge extends exception
```

```
{
 wrongAge(String message);
 super(message);
}
```

```
class Father {
```

```
 int age;
```

```
 Father(int age) throws wrongAge
```

```
 {
 this.age = age;
 }
```

```
 if (age > 0)
```

```
 {
 //
 }
```

```
 throw new wrongAge("Age cannot be negative");
}
```

```
}
```

```
}
```

class son extends father {

int sonage;

son ( int fatherage , int sonage ) throws wrongage

} this.sonage = sonage;

super(fatherage);

if ( sonage >= fatherage )

}

throw new WrongAge ("Son's age cannot be  
greater than or equal to Father's age");

g

g

public class Main {

public static void main ( String [] args )

Scanner sc = new Scanner ( System . in );

System.out.println ("enter father's age");

int fatherage = sc.nextInt();

System.out.println ("enter the son's age");

int sonage = sc.nextInt();

System.out.println ("Father's Age "+ fatherage + " , Son's  
age is "+ sonage );

3

g

Output :-

enter father's age :- 42

enter son's age :- 55

→ Exception in thread "main" wrongAge

addressed →

→ enter father's age - 42

enter son's age - 20

→ Program - 7 - create a package CIE which has two classes student and internal, the class student has members like USN, sem, name, create a folder CIE and save the programs student.java and internal.java, create a folder SEE and save program external.java

package CIE;

public class student {

String USN;

String name;

int sem;

}

```
package CIE;
public class Internals extends Student {
 int[] internal_marks = new int[5];
}
```

```
package SEE;
import CIE.Student;
public class External extends Student {
 int[] external_marks = new int[5];
}
```

```
import CIE.Internals;
import SEE.External;
public class Main {
 public static void main (String[] args) {
 Internals student1 = new Internals();
 student1.usn = "1BM22CS304";
 student1.name = "Tanush";
 student1.sem = 3;
 student1.intern_marks = new int[5] {80, 90, 85, 92, 99};
 External student2 = new External();
```

Internals student1 : new Internals()

student1.usn : "1BM22CS304";

student1.name : "Tanush";

student1.sem : 3;

student1.intern\_marks :

new int[5] {80, 90, 85, 92, 99};

External student2 : new External();

→ WAP which creates 2 threads  
1) BMS college of engineering every 10 seconds  
2) CSE once every 2 seconds

Thread 1 extends Thread

```
? public void run ()
```

```
? for ? i=0; i<10; i++
```

```
? System.out.println ("BMS college");
```

```
? Thread.sleep (10000);
```

```
}
```

```
? catch (InterruptedException e)
```

```
}
```

```
? System.out.println ("exception occurred");
```

class Thread2 extends Thread {

```
public void run ()
```

```
? for ? i=0; i<10; i++
```

```
? System.out.println ("CSE");
```

```
? Thread.sleep (2000);
```

```
catch (InterruptedException)
```

```
{
 System.out.println ("exception 2 occurred");
```

```
}
```

```
y
```

```
class main {
```

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

```
{
```

```
 Thread T1 = new Thread();
```

```
 Thread T2 = new Thread();
```

```
 t1.start();
```

```
 t2.start();
```

```
}
```

```
}
```

```
output />
```

```
BMS college of engineering.
```

```
CSE
```

```
BMS college of engineering
```

Program :- create label , button and textfield  
and frame using AWT

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

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

```
public class AWT example extends windowAdapter
```

```
{
```

```
frame f;
```

```
Awt example ()
```

```
f = new Frame();
```

```
f = addWindowListener (this);
```

```
label l = new Label ("Employee Id");
```

```
button b = new Button ("Submit");
```

```
textfield t = new textfield();
```

```
t.setbounds (20, 80, 80, 30);
```

```
f.setbound (20, 100, 80, 30);
```

```
b.setbound (100, 100, 80, 30);
```

```
f.add (b);
```

```
f.add (l);
```

```
f.add (t);
```

```
f.setsize (400, 300)
```

```
f.setLayout (null);
```

```
f.setVisible (true);
```

```
y
```

```
public void windowClosing(WindowEvent e)
{
 System.exit(0);
}

public static void main(String[] args)
{
 AWTExample obj = new AWTExample();
}
```

O/P : employeeID



### Example-2

```
public class FileEx
{
 public static void main(String a[])
 throws IOException
 {
 FileInputStream fin = new FileInputStream
 ("example.txt");
 int content;
 sop("Remaining bytes that can be
 read: " + fin.available());
 }
}
```

sop(content + " ");  
sop("Remaining bytes that can be read": + fin.  
available());

sop("Remaining bytes that can be read": + fin.  
available());

O/P :- Remaining Bytes that can be read : 1 A  
Remaining bytes that can be read: 0 65