

## LAB - 01

Write a program to overload the method print 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.

Code:

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 numbers is " + sum);

Void Print (int m, int n) {

System.out.println ("Prime numbers 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);
```

3

3

## Program 2

Write a Java program to create a class Grocery that has the variables c-name and c-ph. Create a method to accept 3 parameters to specify quantity of pulses and quantity of sugar. The method to return the total price. Display the name, ph-no and total bill of 3 customers.

Code:

```
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;
```

3

```
void calc (double q-dal, double q-pulses, double q-sugar) {
    total = q-dal * 100 + q-pulses * 80 + q-sugar * 50;
```

3

Void display()

```
System.out.println("Name " + " " + phone number
                    " " " " + " Total ");
```

```
System.out.print(c-name + " " + c-ph + " "
                  + total);
```

```
System.out.println();
```

}

}

class G Demo {

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

```
Grocery g1 = new Grocery("Rama", "8060302010");
```

```
Grocery g2 = new Grocery("Shana", "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

Rama

Shana

Bhama

Phone number : Total

8060302010 410.0

7689632510 800.0

9632587412 205.0

## 1) Output:

Sum of 5 natural numbers is 15  
Prime numbers in the range are

7

11

13

```
3) import java.util.Scanner;  
class Quad {  
    int a,b,c;  
    double root1, root2, d;  
    Scanner s = new Scanner (System.in);  
    void input () {  
        System.out.print ("Quadratic equation in the form:  
                        ax^2 + bx + c");  
        System.out.print ("Enter a:");  
        a = s.nextInt();  
        if (a == 0) {  
            System.out.println ("Not Quadratic!");  
        }  
        else {  
            System.out.print ("Enter b:");  
            b = s.nextInt();  
            System.out.print ("Enter c:");  
            c = s.nextInt();  
        }  
        }  
        void discriminant () {  
            int d = (b*b) - (4*a*c);  
        }  
        void calculate Roots () {  
            if (d > 0) {  
                System.out.println ("Roots are real and unequal");  
            }
```

```
xroot 1 = (-b + Math.sqrt(a)) / (2*a);  
xroot 2 = (-b - Math.sqrt(a)) / (2*a);  
System.out.println("First Root: " + xroot 1);  
System.out.println("Second Root: " + xroot 2);  
}
```

else if ( $d == 0$ ) {

```
System.out.println("Roots are real and equal");  
xroot 1 = (-b + Math.sqrt(d)) / (2*a);  
System.out.println("Root: " + xroot 1);  
}
```

else {

```
System.out.println("No real solution");
```

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

```
double imaginary = Math.sqrt(-d) / (2*a);
```

```
System.out.println("The equation has 2  
complex roots: " + real +  
" + " + 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 equation in the form:  $ax^2 + bx + c$   
Enter a: 2  
Enter b: 2  
Enter c: 2

No real solutions, Roots are Imaginary

The equation has two complex roots:  $0.0 + 0.866i$ ;  
and  $0.0 - 0.866i$ ;

~~QA~~

~~Q2 Q3~~

~~25~~

## LAB-02

Program 1: WAP to accept student details and marks also include a method to calculate and display appropriate details

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

```
class Student {
```

```
    String USN;
```

```
    String name;
```

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

```
    void Details()
```

```
{
```

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

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

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

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

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

```
    System.out.print("Enter Marks for 6  
subjects: ");
```

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

```
{
```

```
    System.out.print("Subject " + (i + 1) + ": ");
```

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

```
}
```

```
} double percentage()
```

```
{
```

```
    int total = 0;
```

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

```
{
```

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

```
}
```

```
double P = total / 6;  
return P;  
}
```

```
void display()  
{
```

```
System.out.println("In Student details:");
```

```
System.out.print(" USN: " + usn);
```

```
System.out.print(" Name: " + name);
```

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

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

```
{
```

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

```
}
```

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

```
}
```

```
class Lab1Student
```

```
{
```

~~Public static void main (String args [] ) {~~~~Scanner s = new Scanner (System.in);~~~~System.out.print("Enter the number of  
students: ");~~~~int n = s.nextInt();~~~~Student[] students = new Student[n];~~~~for (int i = 0; i < n; i++)~~

```
{
```

~~students[i] = new Student();~~~~System.out.println("In Enter details for  
student " + (i + 1) + ":");~~~~students[i].Details();~~

```
}
```

for (Student student : students)

{

student . display ();

}

}

input (" Enter id " ) altreq . tuo . matu2

(num + " name " ) altreq . tuo . matu2

: Output : - num + " name " altreq . tuo . matu2

( " name " ) altreq . tuo . matu2

Enter the USN of student

IBM 22 CS 310

Enter the name of student

Tulasi Krishna Tamina

Enter the marks of student in all 6 subjects

78

56

90

99

79

89

Name : Tulasi Krishna Tamina

USN : IBM 22 CS 310

Percentage : ~~81.8333333%~~

1) Repeat n more times.

for ( Student student : students )

{

2) Student student = ( Student )

( " " + ( i + 1 ) + " " ) altreq . tuo . matu2

( " " + ( i + 1 ) + " " ) altreq . tuo . matu2

⇒ Create a class Book that contains four members: name, author, price and num-pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a java program to create n book objects.

```
import java.util * ;
```

```
class Books
```

```
{
```

```
String name;
```

```
String author;
```

```
int price;
```

```
int numPages;
```

```
Books (String n, String a, int p, int n)
```

```
{
```

```
name = n;
```

```
author = a;
```

```
price = p;
```

```
numPages = n;
```

```
}
```

## String to String ()

```

String name, author, price, numpages;
name = "Book name :" + this.name + "\n";
author = "Author :" + this.author + "\n";
price = "Price : Rs ." + this.price + "\n";
numpages = "No. of pages :" + "this.numpages" + "\n";
return name + author + price + numpages;
}

```

## 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 number of books");
```

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

```
Books b[] = new Books [n];
```

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

```
System.out.print ("Name : ");
```

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

```
System.out.print ("Author : ");
```

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

```
System.out.print ("Price : ");
```

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

```
System.out.println ("No of pages:");
```

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

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

Output:

Enter number of books

1

Name :

Xyz

Author :

ABC

Price :

100

No of pages :

90

Book name : XYZ

Author : ABC

Price : Rs 100

No. of pages : 90

## LAB-103

Q1) Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain the method printArea() that prints the area of the given shape.

abstract class Shape {

int length;

int width;

public Shape (int length, int width) {

this.length = length;

this.width = width;

}

public abstract void printArea();

3  
class Rect extends Shape {

public Rectangle (, ) {

super (, );

}

public void printArea () {

float area = length \* width;

System.out.println ("Area of Rectangle " + area);

3

class Tri extends Shape {

public Triangle (, ) {

super (, );

}

```

public void printArea() {
    float area = 0.5 * length * width;
    System.out.println("Area of triangle " + area);
}

```

```

class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);
    }
}

```

```

public void printArea() {
    float area = 3.14 * length ** 2;
    System.out.println("Area of circle = " + area);
}

```

Program - (1). java output

```

public class Main {
    public static void main(String args[]) {
        Rect rect = new Rect(13, 2);
        Tri tri = new Tri(13, 2);
        Circle cl = new Circle(13);
        rect.printArea();
        tri.printArea();
        cl.printArea();
    }
}

```

Output :-

$\text{Area of rectangle} = 112.0$   
 $\text{Area of triangle} = 100.0$   
 $\text{Area of circle} = 78.5$

Q) Bank Question

Import java.util.Scanner

class Account {

String c.name;

long acc no;

String acctype;

double bal;

// Constructors

Public void deposit (double amt) {

bal += amt;

SOP ("Deposited ! Balance = " + balance);

}

Public void display bal () {

SOP ("Balance = " + bal);

}

}

```
class Current extends Account {  
    double service_charge; minbal;  
    // constructor with super (cname, accno, "current");  
    public void deposit (double amt) {  
        super.deposit (amt);  
    }  
}
```

```
public void display_bal () {
```

```
    super.display_bal ();
```

```
    public void minbal () {
```

```
        if (bal < minbal) {
```

```
            bal -= service_charge;
```

```
            System.out.println ("balance");
```

```
}
```

```
class Sarcet extends Account {
```

```
    double interest_rate;
```

```
    // constructor with super
```

```
    public void check_interest () {
```

```
        double interest = bal * interest_rate;
```

```
        bal += interest;
```

```
        System.out.println ("balance")
```

```
    public void display_bal () {
```

```
        check_interest ();
```

```
        super.display_bal ();
```

```
    public void withdraw (double amt) {
```

```
        if (amt <= bal) & bal -= amt;
```

```
        System.out.println ("balance");
```

```
}
```

```
else {  
    System.out.println("Insufficient balance");  
}  
}
```

```
public class main {  
    public static void main (String args []) {  
        Scanner S = new Scanner (System.in);  
        CurrentAcct c1 = new CurrentAcct (1, , );  
        c1.deposit (20000);  
        c1.displayBal ();  
        SavingsAcct s1 = new SavingsAcct (1, , );  
        c1.deposit (20000);  
        c1.displayBal ();  
        System.out.println ("Enter withdrawal amount: ");  
        double withdrawl amt = Scanner.nextDouble();  
        s1.withdrawl ();  
        s1.displayBal ();  
        Scanner.close ();
```

3

~~8~~

Week - 04

## CIE Package

Package CIE;

public class Personal {

    public String usn;

    public String name;

    public int sem;

    public Personal (String usn, String name, int sem) {

        this.usn = usn;

        this.name = name;

        this.sem = sem;

}

}

import java.util.Arrays;

public class Internals {

    public int [] internalMarks;

    public Internals (int [] internalMarks) {

        this.internalMarks = internalMarks;

}

}

Package SEE;

import CIE.Personal;

public class ~~External~~ External Extends Personal {

    public int [] seeMarks;

    public External (String usn, String name,

        int sem, int [] seeMarks) {

        super (usn, name, sem);

        this.seeMarks = seeMarks;

}

}

```
package Fox.main;
import java.util.Arrays;
import CIE.internals;
import CIE.personal;
import SEE.External;
public class Main {
    public static void main (String [] args) {
        int n = 3;
        Student [] students = new Student [n];
        for (int i = 0; i < n; i++) {
            int [] internalMarks = {80, 75, 90, 85, 95};
            int [] seeMarks = {70, 80, 75, 90, 85};
            students [i] = new Student (new Personal ("USN"
                + i, "Student" + i, 1),
                new Internals (internalMarks));
            Student [i].see = new External ("USN" + i,
                "Student" + i, 1, SeeMarks);
        }
        for (int i = 0; i < students.length; i++) {
            Student student = students [i];
            System.out.println ("Student : " + student.personal.name);
            System.out.println ("Internal Marks : " + Arrays.toString
                (student.internals.internalMarks));
            System.out.println ("SeeMarks : " + Arrays.toString
                (student.see.seeMarks));
            System.out.println ();
        }
    }
}
```

```
Static class Student {  
    public personal personal;  
    public Internals Internals;  
    public External SEE;  
    public Student (personal personal, Internals  
        Internals) {
```

this. personal = personal;

this. Internals = Internals;

}

E

Output:-

Student : Student 0

Internal marks : [80, 75, 90, 85, 95]

SEE marks : [70, 80, 75, 90, 85]

Student : Student 1

Internal marks : [80, 75, 90, 85, 95]

SEE marks : [70, 80, 75, 90, 85]

Student : Student 2

Internal marks : [80, 75, 90, 85, 95]

SEE marks : [70, 80, 75, 90, 85]

Week - 06

## Father - son Age Relationship

class Father {

    public int age;

    Father (int age) {

        if (age <= 0) {

            throw new Illegal Argument Exception (

            "Age cannot be -ve");

    }

    this. age = age;

}

    public class son extends Father {

        public int sonAge;

        public son (int fatherAge, int sonAge) {

            super (fatherAge);

            if (sonAge >= fatherAge) {

                throw new Illegal Argument Exception (

                "son's age cannot be >= father's age");

}

        this. sonAge = sonAge;

}

import java.util.Scanner;

public class main {

    public static void main (String [] args) {

        Scanner s = new Scanner (System.in);

    try {

        S.o.p ("Enter father's age : ");

        int fatherAge = Scanner. NextInt();

```
S.o.p (" Enter son's age : ");  
int sonage = Scanner . nextInt ();  
Son son = new Son (father Age , son age );  
S.o.p (" Father's age : " + son . age );  
S.o.p (" son's age : " + son . sonage );  
}
```

```
catch (Illegal argument Exception e) {
```

```
    S.o.p ("Exception : " + e . getMessage ());  
}
```

```
    Scanner . close ();  
}
```

```
}
```

Output:

Enter father's age : 20

Enter son's age : 40

Exception : son's age cannot be >= father's age.

✓

## BMSCE - CSE thread :

```
public class Main {  
    public static void main (String [] args) {  
        Thread thread1 = new Thread (new Runnable () {  
            public void run () {  
                while (true) {  
                    System.out.println ("BMSCE");  
                    try {  
                        Thread.sleep (1000);  
                    } catch (InterruptedException e) {  
                        e.printStackTrace ();  
                    }  
                }  
            }  
        });  
        Thread thread2 = new Thread (new Runnable () {  
            public void run () {  
                while (true) {  
                    System.out.println ("CSE");  
                    try {  
                        Thread.sleep (2000);  
                    } catch (InterruptedException e) {  
                        e.printStackTrace ();  
                    }  
                }  
            }  
        });  
        thread1.start ();  
        thread2.start ();  
    }  
}
```

# ① Creating label, button and Textfield in a Frame Using AWT

```

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

public class AWTExample extends WindowAdapter {
    Frame f;
    AWTExample() {
        f = new Frame();
        f.addWindowListener(this);
        Label l = new Label("Employee id:");
        Button b = new Button("Submit");
        TextField t = new TextField();
        l.setBounds(20, 80, 80, 30);
        b.setBounds(20, 100, 80, 30);
        t.setBounds(100, 100, 80, 30);
        f.add(b);
        f.add(l);
        f.add(t);
        f.setSize(400, 300);
        f.setTitle("Employee info");
        f.setLayout(null);
        f.setVisible(true);
    }

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

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

```

② Create a button and add a action listener for Mouse click.

```
Import java.awt.*;  
Import java.awt.event.*;  
public class EventHandling extends WindowAdapter  
implements ActionListener {
```

Frame f;

TextField tf;

```
EventHandling() {
```

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

```
f.addWindowListener(this);
```

```
tf = new TextField();
```

```
tf.setBounds(60, 50, 170, 20);
```

```
Button b = new Button("click me");
```

```
b.setBounds(100, 120, 80, 30);
```

```
b.addActionListener(this);
```

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

```
f.setSize(300, 300);
```

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

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

```
public void actionPerformed(ActionEvent) {  
    tf.setText("Welcome"); } }
```

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

```
public static void main(String args[]) {  
    new EventHandling(); }
```

Output

①

I am showing output :-

"Employee Info"

Employee ID:

SUBMIT

Employee ID

fixed fee 30.00/-

Employee info will be displayed

Output ② :-

Welcome

click me

## Example programs on IO

```
① import java.io.*;
public class Byte Array Input {
    public static void main (String [] args) throws
        IOException {
        byte [] buf = { 35, 36, 37, 38 };
        ByteArrayInputStream byt = new ByteArrayInputStream
            (buf);
        int k = 0;
        while ((k = byt.read ()) != -1) {
            char ch = (char) k;
            System.out.println ("ASCII Value of character is :
                " + k + "; Special character
                is : " + ch);
    }
}
```

Output:

ASCII value of character is : 35;  
Special character is : #

ASCII value of character is : 36;  
Special character is : \$

ASCII value of character is : 37;  
Special character is : %

ASCII value of character is : 38;  
Special character is : &

② import java.io.\*;

```
public class ByteArrayEx {  
    public static void main (String args[]) throws  
        Exception {  
        FileOutputStream bout1 = new FileOutputStream  
            ("Example.txt");  
        FileOutputStream bout2 = new FileOutputStream  
            ("Example2.txt");
```

Byte Array Output Stream bout = new  
 Byte Array output stream;

```
    bout.write (65);  
    bout.writeTo (fout1);  
    bout.writeTo (fout2);
```

bout.flush();

bout.close();

System.out.println ("Success...");

}

}

Output :-

→ java ByteArrayInput  
Success...

③ public class FileEx {  
    public static void main (String a[]) throws  
                                IOException {  
        FileInputStream fin = new FileInputStream  
                                ("Example.txt");  
        int content;  
        System.out.println ("Remaining bytes that can be  
                                read: " + fin.available());  
  
        content = fin.read();  
        System.out.println ((char) content + " ");  
        System.out.print (content + " ");  
        System.out.println ("Remaining bytes that can  
                                be read: " + fin.available());  
  
        System.out.println ("Remaining bytes that can  
                                be read: " + fin.available());  
    }  
}

Output :-

Remaining bytes that can be read: 1  
A 65 Remaining bytes that can be read: 0  
Remaining bytes that can be read: 0

④ Import java.io.FileInputStream;  
Import java.io.IOException;

```
public class FileEx2 {
    public static void main (String ar[]) throws
        IOException {
        FileInputStream fin = new FileInputStream
            ("Example.txt");
        byte [] bytes = new byte [20];
        int i;
        char c;
        i = fin.read (bytes);
        System.out.println ("Number of bytes read: " + i);
        System.out.println ("Bytes read: ");
        for (byte b: bytes) {
            c = (char) b;
            System.out.print (c);
        }
    }
}
```

Output:

Number of bytes read: 1  
Bytes read: A