

Lab Exercise - 22/12/2023

Q1) Write a program to overload the method print that prints sum of n natural numbers when one variable is passed, and prints the prime number in a given range when 2 parameters are passed.

Solⁿ 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);

3

```
void print (int m, int n) {
```

```
    System.out.print (" Prime numbers  
in the range are ");
```

```
    for (int j = 2; j <= i / 2; j++) {
```

```
        if (i % j == 0) {
```

```
            flag = 1;
```

```
            break;
```

```
}
```

```
}
```

```
        if (flag == 0)
```

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

```
}
```

```
}
```

```
}
```

```
class OverloadDemo {
```

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

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

```
    o.print (5);
```

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

```
}
```

Q12) 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 of dal, 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.

Ans) 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) {
```

$$\text{total} = q_{\text{dal}} * 100 + q_{\text{pulses}} * 80
+ q_{\text{sugar}} * 50;$$

{

```
void display()
```

{

```
System.out.printIn ("Name " + "  
" Phone number " " + " " + "Total");
```

```
System.out.printIn (c_name + " "  
c_ph + " " + total);
```

```
System.out.printIn ();
```

{

{

Class GDem0 {

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

Grocery g2 = new Grocery ("Rama", "8060302010")

Grocery g2 = new Grocery ("Sharma", "7689632510")

Grocery g3 = new Grocery ("Bhama", "9632587412")

g2.calc(2, 2, 1);

g2.display();

g2.calc(3, 5, 2);

g2.display();

g3.calc(1, 1, 0, 5);

g3.display

3

3

OUTPUT: Name : Rama

Phone - number = 8561011554

Total - amount = 410.

Q13) Write a Java program to calculate roots of a quadratic equation. Use appropriate methods to take input and calculate the roots.

Ans) 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
equation 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 and 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 ($a == 0$)

{

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

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

System.out.println ("Root: " + root1);

3

else

{

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

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

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

3

3

3

class main {

public static void main (String[] args) {

Quad q = new Quad();

or: insurce();

or: discriminant();

or: calculate Roots();

3

3

~~Print~~
-22/12/23

12-01-24

1) Program-1:

import java.util.Scanner;

class student {

String USN;

String name;

double [] marks = new double [6];

void acceptDetails() {

Scanner scanner = new Scanner (System.in);

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

USN = scanner.nextLine();

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

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

marks[i] = scanner.nextInt();

}

}

double calculatePercentage ↳ ?

double totalMarks = 0;

for (double mark : marks) {

 totalMarks += mark;

}

return (totalMarks / 6);

3

void displayDetails() {

 System.out.println("USN: " + USN);

 System.out.println("Name: " + Name);

 System.out.println("Percentage: " +

 calculatePercentage() + "%");

3

3

public class Main {

 public static void main (String [] args)

 Scanner Scanner = new Scanner (System.in);

 System.out.print ("Enter the number

 of Students: ");

 int numStudents = Scanner.nextInt();

 Student [] students = new Student [numStudents];

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

{

 Students [i] = new Student ();

 System.out.print ("Enter details

 for student " + (i + 1) + ":");

 Students [i].acceptDetails();

}

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

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

```
{  
    System.out.println("Details for  
Student" + (i + 1) + ":");
```

```
    student[i].displayDetails();
```

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

3
3
3

27 Program-2

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

```
class Book {
```

```
    private String name;  
    private String author;  
    private double price;  
    private int numPages;
```

~~11 constructor to set the values for the members~~

```
public Book (String name, String author,  
double price, int num Pages) {
```

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

```
    this.author = author;
```

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

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

3.

11 Getter Methods

public String getName() {
 return name;

3

public String getAuthor() {
 return author;

3

public double getPrice() {
 return price;

3

public int getNumPages() {
 return numPages;

3

11 Setter Methods

public void setName(String name) {
 this.name = name;

3

public void setAuthor (String author) {
 this . author = author;

3

public void setPrice (double price) {
 this . price = price;

3

public void setNumPage (int numPage) {
 this . numPage = numPage;

3

@ Override

public String toString () { return "Book
details " + author + "\n Price: \$" + num
Page; }

3

3

public class Main {

 public static void main (String args[]){
 Scanner scanner = new Scanner (System. in);
 System.out.print ("Enter the number
 of books: ");
 }

 int numBooks = scanner.

 3

 3

 3

O/P:

Enter no. of students: 1

Name: Varsh

USN: IBM22S318

Enter marks: 100 100 100 100 100 100

The percentage of Varsh with IBM22S318

is: 100.0

19/01/24

Program - 1: Area

abstract class shape {

 double a, double b;

 shape (double l, double r)

}

 a = l;

 b = r;

}

 shape (double r)

}

 a = r;

}

abstract void printArea();

3

Class Rectangle extends shape

{

 Rectangle (double l, double b)

{

super (l, k);

{

void PrintArea()

{

double area;

area = a * b

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

{

{

Class Triangle extends shape

{

Triangle (double b, double h)

{

Super (b, h);

{

void PrintArea()

{

double area;

$$\text{area} = (l * b) / 2$$

System.out.println ("Area of triangle
+ area")

{

{

class Circle extends Shape

{

Circle (double r)

{

super();

{

void PrintArea()

{

double area;

$$\text{area} = 3.14 * r * r$$

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

{

{

class Run {

 public static void main (String [] args)
 {

 Rectangle r = new Rectangle (2,4);

 Triangle t = new Triangle (2,6);

 Circle c = new Circle (3);

 Shape s

 s = r

 s. printArea();

 s = t;

 s = printArea()

 s = c;

 s. printArea();

 }

}

O/P: Area of rectangle = 16

 Area of triangle = 6

 Area of circle = 28.26.

Program-2: Bank

```
import java.util.Scanner;
class Account {
    String customerName;
    long accno;
    String accountType;
    double balance;
    public Account (String customerName,
                    long accno, String accountType)
    {
        this.customerName = customerName;
        this.accno = accno;
        this.accountType = accountType;
        this.balance = 0.0;
    }
    public void displayBalance()
    {
        System.out.println ("Account Number : "
                            + accno);
    }
}
```

System.out.println ("Customer Name: "+
customerName);

System.out.println ("Account Type: "
+ accountType);

System.out.println ("Balance: \$ " + balance
3

3

class CurAcct extends Account {

double minBalance;

double serviceCharge;

public CurAcct (String customerName,
long accno) {

super (customerName, accno, "current");

this.minBalance = 500.0

this.serviceCharge = 50.0

3.

public void withdraw (double amount)
if (balance - amount >= minBalance)
balance -= amount;

System.out.println ("Withdrawal
successful. Current Balance : \$ " + balance)
}

else {

System.out.println ("Insufficient
funds withdrawal not allowed")

}

}

public void impose Service Charge {
if (balance < minBalance) {
balance = serviceCharge;

System.out.println ("Service charge
imposed current Balance : Rs " + balance)

}

3

3

```
class SavAcct extends Amount {  
    double interestRate = 0.05;  
    SavAcct (String customerName, long  
    accountNumber, double balance) {  
        super (customerName, accountNumber,  
        "Savings", balance);  
    }  
}
```

```
public void computeInterest() {  
    double interest = balance * interestRate;  
    balance = balance + interest;  
    System.out.println ("Interest of " + interest);  
    displayBalance();  
}
```

```
3  
public class Test {  
    public static void main (String args[]) {  
        Scanner sc = new Scanner (System.in);  
    }  
}
```

S1. displayBalance();

S1. deposit(1000)

S1. computerInterest()

S1. withdrawal(2000)

Customer C1 = new Customer ("Aditya", 9876543210,
1500);

C1. displayBalance();

C1. deposit(500);

C1. withdrawal(2000);

3

3

O/P:

Choose Account type:

1. Current

2. Savings

Enter choice (1 or 2): 1

Enter customer name: Vaish

Enter account number: 0

DOMS Page No.
Date / /

Enter initial balance: \$1000

Enter withdrawal balance: \$100

withdrawal successful.

16/21/24.

Q1) Program - 1

```
package CIE;
import java.util.Scanner;
public class student {
    protected String USN = new String();
    protected String USN = new String();
    protected int sem;

    public void inputStudentDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.println("give USN:");
        USN = sc.nextLine();
        System.out.println("give name:");
        name = sc.nextLine();
        System.out.println("give sem:");
        sem = sc.nextInt();
    }
}
```

```
package CIE;  
import java.util. scanner;  
protected int marks [] = new int [5];  
public void input () marks () {  
Scanner sc = new scanner (System.in);  
for (int i = 0; i < 5; i++) {  
System.out.print ("Enter marks for  
course: " + (i+1));  
marks [i] = sc.nextInt();  
}  
}
```

3 package SEE;

```
import CIE.internals;  
import java.util. scanner;  
public class External extends Internals {  
protected int marks [];  
protected int final marks [];
```

3

class External () {

```
marks = new int [5];  
final marks = new int [5];
```

```
public void inputSEE marks () {
    Scanner sc = new Scanner (System.in);
    for (int i=0; i<5; i++) {
        System.out.println ("Subject" + (i+1),
                            "marks:");
        marks [i] = sc.nextInt ();
    }
}

public void calc final marks () {
    for (int i=0; i<5)
        finalmarks [i] = marks [i] + supermarks [i] / 2;
}

package SEE;

public class Finalmark extends Interface {
    public Extends;
    public class Main {
        public static void main (String args) {
            for (int i=0; i<num of Students; i++) {
                finalmarks [i] = new Extends ();
                finalmarks [i] = new Extends ();
            }
            finalmarks [i] = out (finalmarks);
        }
    }
}
```

system.out.println ("display data:");
for (int i = 0; i < no.of Students; i++) {
 marks[i].calcFinalMarks();

3

3

?

Program - 2 :

→ import java.util.Scanner;
class wrongAge extends Exception {
 public wrongAge() {
 super ("Age can't be negative");
 }
}

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

3

class InputScanner {

public static int readInt() {
 try {
 Scanner scanner = new Scanner (System.in);
 return scanner.nextInt();
 } catch (Exception e) {}
 }
}

3

3

3

class Father extends InputScanner {
protected int fatherAge;
public Father () throws wrongAge {
System.out.println("Enter father's age: ");
fatherAge = readInt();
}}

if (fatherAge < 0) {

 throw new WrongAge ("Age cannot be
 negative");
}

}

public void display () {

 System.out.println ("Father age: " +
 fatherAge)

}

}

class son extends Father {

 protected int sonAge;

 public son () throws wrongAge {

 super ();

 }

 System.out.println ("Enter son's age: ");

```
sonAge = readInt();  
if (sonAge) = super·fatherAge;  
throw new WrongAge ("Son's age cannot  
be greater than father's");  
}
```

```
else if (sonAge < 0);
```

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

```
}
```

```
public void display();
```

```
super·display();
```

```
System.out.println ("Son's age: " +  
sonAge);
```

```
}
```

```
}
```

public class Exception {

 public static void main (String a[]) {
 try {

 son son = new Son();

 son.display();

}

 catch (WrongAge e) {

 System.out.println ("Error: " + e.getMessage());

}

}

}

Program-3:

```
→ class DisplayMessage extends Thread {  
    private String message;  
    private int interval;  
    public DisplayMessage (String message,  
    int interval) {  
        this.message = message;  
        this.interval = interval;  
    }  
    public void run () {  
        while (true) {  
            System.out.println (message);  
            try {  
                Thread.sleep (1000);  
            } catch (InterruptedException e) {  
                e.printStackTrace ();  
            }  
        }  
    }  
}
```

public class Threaded {

 public static void main (String args[]){

 DisplayMessage t1 = new DisplayMessage("

 BMS College of Engineering", 10);

 DisplayMessage t2 = new DisplayMessage("02");

 thread1.start();

 thread2.start();

3

3

QD : BMS College of Engineering

CSE

BMS college of engineering

CSE

CSE

CSE

BMS college of Engineering

23/02/24

1> → import java.awt.*;
import java.awt.event.*;
public class AWTExample 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();

l.setBounds (20, 80, 80, 30);

t.setBounds (20, 100, 80, 30);

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

f.add (b);

f.add (t);

f.add (l);

f.setSize (400, 300);

b. set Title ("Employee info");

c. setLayout (null);

d. setVisible (true);

3

public void windowClosing (WindowEvent e)

?

System.exit (0);

3

public static void main (String args [])

AWT Example awt obj = new AWT Example();

3

O/P:

Employee id

Submit

(Q2)

→ import java.awt.*;
import java.awt.event.*;
public class EventHandling extends
WindowAdapter implements ActionListener {

Frame f;

f.add window Listener (this);

H = new TextField();

tf.setBounds (60, 50, 130, 20);

Button b = new Button ("Click me");

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

b.addActionListener (this);

f.add (b);

f.add (tf);

F.setLayout (null);

f.setVisible (true);

3

public void windowClosing (WindowEvent) {
System.exit (1);}

3

DOMS

Date / /

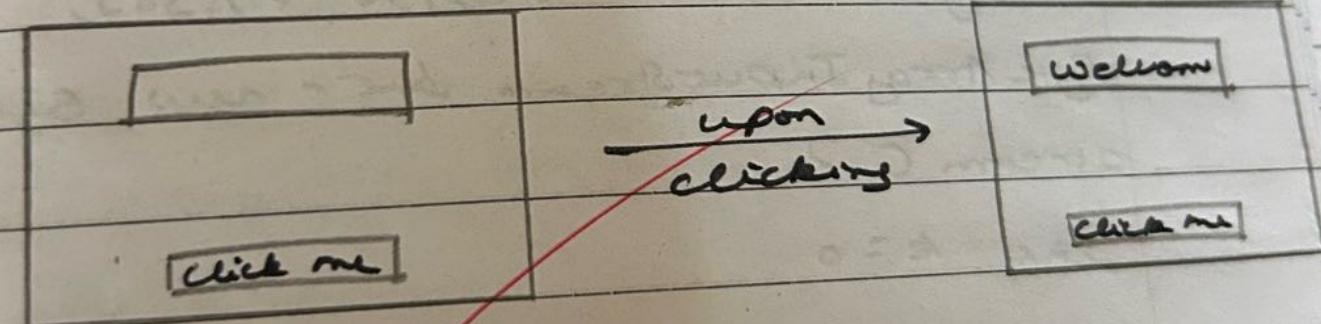
Page No.

public static void main (String args [])
new Event Handling;

3

3

O/P:



Example-1:

```
import java.io;
```

```
public class Byte Array Input {
```

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

```
        IOException {
```

```
            byte [] buf = { 35, 36, 37, 38 };
```

Byte Array InputStream buf = new ByteArrayInputStream
stream ()

```
    int k = 0
```

```
    while ((k = buf.read ()) != -1) {
```

```
        char ch = (char) k;
```

System.out.println (" ASCII value of

Character is: " + k + " ; Special character is: "

ch);

S

3

J

Example-2 :

```

import java.io.*;
public class ByteArrayInput {
    public static void main (String a[]) throws
    IOException {
        byte byt [] = {35, 36, 37, 38};
        ByteArrayInputStream bnt = new ByteArrayInputStream
        (byt);
        int k=0;
        while ((k = bnt.read ()) != -1) {
            char ch = (char)k;
            System.out.println ("ASCII value of character"
                + ch);
        }
    }
}

```

Example-3

```
public class Example2
{
    public static void main (String args[])
        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.println (content + " ");
        System.out.println ("Remaining bytes that can be read: "
            + fin.available ());
        System.out.println ("Remaining bytes that can be read: " +
            fin.available ());
    }
}
```

3.

?

Example 4

```
import java.io. FileInputStream;
import java.io. IOException;
public class FileFw {
    public static void main (String a[]) 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);
    }
}
```

Example - 5

```
import java.io.*;  
public class Byte Array - ex 2  
    public static void main (String args[]){  
        // Example -
```

```
        File output stream fout = new FileOutputStream  
        ("Binaryfile 1")
```

```
        File output stream fout2 = new FileOutputStream  
        ("Binaryfile 2")
```

```
        Byte Array Output Stream bout = new Byte Array  
        Output Stream ();
```

```
        bout. write (65);
```

~~```
bout. write to (fout1);
```~~~~```
bout. write to (fout2);
```~~~~```
bout. flush ();
```~~~~```
bout. close ();
```~~

```
3 System. out. print ("Success ... ");
```

OIP:

- 1) Ascii 8 char: 35; special char: #
Ascii 9 char: 36; special char: \$
Ascii 0 char: 37; special char: .
Ascii A char: 38; special char: ?

2) Remaining bytes that can be read: 2

h

104

i

105

Remaining bytes that can be read: 0.

4) Number of bytes read: 2

Bytes read: hi

Sneha
13/12/24