

22/12/2023 -:

Date \_\_\_\_\_  
Time \_\_\_\_\_

(Q.1.) 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.

Ans → 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 OverloadDemo {

```

public static void main (String [] args) {
    Overload o = new Overload();
    o.print(5);
    o.print(7, 13);
}

```

## Output :-

Sum of 5 natural numbers is 15

Prime numbers in the range are

7

11

13

Q. 2.) Write a Java program to create a class Grocery that has the variables c\_name and c\_phne. 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\_ne 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) {
}

```

total = q\_dal \* 100 + q\_pulses \* 80 + q\_sugar \* 50;

void display()

{

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

System.out.println(c\_name + " " + c\_ph + " " +  
total);

System.out.println();

}

### Class CDemo

public static void main(String[] args) {  
Grocery g1 = new Grocery("Rama", "8060302010")

Grocery g2 = new Grocery("Sham", "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 | 8060302010   | 410.00 |

| Name   | Phone number | Total  |
|--------|--------------|--------|
| Shama  | 7689632510   | 800.00 |
| Bhamra | 9632587412   | 25.00  |

Q. 30)

Write a Java program to calculate roots of a quadratic equation. Use appropriate methods to take inputs, 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 (d == 0)
```

```
        System.out.println ("Roots are real and equal");  
        root1 = (-b + Math.sqrt(d)) / (2 * a);
```

```
        System.out.println ("Root : " + 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 + " + " + 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 is in the form:  $ax^2 + bx + c = 0$

Enter a: 5

Enter b: 2

Enter c: 3

No real solutions. Roots are imaginary

The equation has two complex roots:

$0.0 + 0.7483314773547882i$  and

$0.0 - 0.7483314773547882i$

~~$\sqrt{3}$~~   
 ~~$\sqrt{3}$~~   
 ~~$\pm 1.2$~~

Q.1) Write a Java program to create a class Student with members USN, name, marks (6 subjects). Include methods to accept student details and marks, also include a method to calculate the percentage and display appropriate details.

import java.util.Scanner;

class Student {

String ~~usn~~ usn;

String name;

double [] marks = new double [6];

void acceptDetails () {

Scanner sc = new Scanner (System.in);

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

usn = sc.nextLine();

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

name = sc.nextLine();

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

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

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

marks [i] = sc.nextDouble();

}

y

double totalMarks = 0;

for (double mark : marks) {

totalMarks += mark;

}

return (totalMarks / 6);

```
    void displayDetails() {  
        System.out.println("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.print(subjects[i] + ":" + marks[i]);  
        }  
    }
```

```
public class Student {  
    public static void main (String [] args) {  
        Scanner sc = new Scanner (System.in);  
        System.out.print ("Enter number of students : ");  
    }  
}
```

```
    int numStudents = sc.nextInt();  
    Student [] students = new Student [numStudents];  
    for (int i = 0; i < numStudents; i++) {  
        System.out.println ("Enter details for student " + (i + 1));  
        students [i] = new Student ();  
        students [i].acceptDetails ();  
    }  
    for (int i = 0; i < numStudents; i++) {  
        students [i].displayDetails ();  
    }  
}
```

Output -

Enter number of student : 2

Enter details for student 1 :

Enter USN : 16m22 CS 111

Enter name : aditya

Enter marks for 6 subjects :

Subject 1 : 2.0

Subject 2 : 3

Subject 3 : 4

Subject 4 : 5

Subject 5 : 6

Subject 6 : 6

Student Details :

USN : Ibm 22CS11

Name : aditya

Marks :

Subject 1 : 2.0

Subject 2 : 3.0

Subject 3 : 4.0

Subject 4 : 5.0

Subject 5 : 6.0

Subject 6 : 6.0

Percentage : 4.3333 %

→ book class.java

Q.2.) 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 book. Develop a Java program to create 'n' book objects.

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

```
class Books {
```

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

```
    String name, author;
```

```
    int price, 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() {
```

```
    String bookName = "Book Name : "
```

```
        + this.name + "\n";
```

```
    String authorName = "Author Name : " +  
        this.author;
```

~~```
    String bookPrice = "Price : " + this.price;
```~~~~```
    return bookName + authorName  
        + bookPrice + pages;
```~~

```
class BookProgram {
```

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

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

```
    int n;
```

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

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

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

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

```
System.out.println("Details of book " +  
+ (i + 1) + ": ");
```

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

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

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

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

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

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

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

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

```
for (Books book : b) {
```

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

## Output :-

Enter number of book : 2

Book 2 :

Enter name of book : jungle-book

Enter author of book : Rudyard-Kipling

Enter price of book : 1000

Enter pages : 500

Book 2 :

Book name : jungle-book

Author : Rudyard - Kipling

Price : 1000

Pages : 500

✓  
Jungle Book  
Rudyard Kipling  
1000  
500

Lab → 19/01/2024

(1) Topic  
Page

(1)

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 contains the method printArea() that prints the area of the given shape.

abstract class Shape {

protected int side1;

protected int side2;

public Shape (int side1, int side2) {

this.side1 = side1;

this.side2 = side2;

}

public abstract void printArea();

class Rectangle extends Shape {

public Rectangle (int length, int width) {

super(length, width);

public void printArea () {

int area = side1 \* side2;

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

}

class Triangle extends Shape {

public Triangle (int base, int height) {

super(base, height);

y

Type

```
public void printArea() {  
    double area = 0.5 * side1 * side2;  
    System.out.println("Area of  
    Triangle: " + area);
```

class Circle extends Shape {

```
public Circle(int radius) {
```

```
    super(radius);
```

```
public void printArea() {
```

double area = Math.PI \* side1 \* side2;

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

```
public class program1 {
```

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

```
    Rectangle rectangle = new Rectangle
```

```
(4, 6);
```

```
    Triangle triangle = new Triangle (4, 6);
```

```
    Circle circle = new Circle (4);
```

rectangle.printArea();

triangle.printArea();

circle.printArea();

Output -

Area of Rectangle : 24

Area of Triangle : 12.0

Area of Circle : 50.265482457431

(Q.20)

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name and type of account. From this derive the classes Curr-Acc and Sav-Acc to make them more specific to their requirements:

(a) Accept deposit from customer and update the balance.

(b) Display the balance

(c) Compute and deposit interest

(d) Permit withdrawal and update the balance. Check for minimum balance, impose penalty if necessary and update the balance.

~~Ans)~~ import java.util.Scanner;

```
class Account {
```

```
protected String customerName;
```

```
protected long accountNumber;
```

```
protected String accountType;
```

```
protected double balance;
```

```
public Account (String customerName, long accountNumber, String accountType,
```

```
double balance) {
```

```
this. customerName = customerName;
this. accountNumber = accountNumber;
this. accountType = accountType;
this. balance = balance;
```

```
public Account (String customerName,
                String accountNumber, String accountType,
                double balance) {
```

```
this. customerName =
public void displayBalance () {
    System.out.println ("Account
        Number:" + accountNumber);
```

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

```
System.out.println ("Account Type:" + accountType);
System.out.println ("Balance : $" + balance);
```

```
public void deposit (double amount) {
    balance += amount;
    System.out.println ("Deposit of $" +
        amount +
        " successful");
```

```
displayBalance ();
```

```
public void withdraw (double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println ("Withdrawal of $" +
            amount +
            " successful");
```

else {

    System.out.println("Insufficient funds");  
    displayBalance();

}

class CurrentAccount extends Account {  
    private double minimumBalance = 1000;  
    private double serviceCharge = 50;  
    public CurrentAccount (String customerName,  
                              long accountNumber,  
                              double balance) {

    super (customerName, accountNumber,  
                              "Current", balance);

    public void withdraw (double amount) {  
        if (amount <= balance - minimumBalance)  
            balance -= amount;

        System.out.println("Withdrawal of  
                              \$ " + amount  
                              + " successful");

    else {

        System.out.println("Insufficient funds  
                              Service charge of  
                              \$ " + serviceCharge +  
                              " imposed");

    displayBalance();

class SavAcct extends Account {  
private double interestRate = 0.02;  
public SavAcct (String customerName,  
long accountNumber,  
double balance) {  
super (customerName, accountNumber,  
"Savings", balance);

}  
void computeInterest () {  
double interest = balance \* interestRate;  
balance += interest;  
displayBalance ();

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

SavAcct savingAct = new SavAcct ("John Doe",  
1234567890,  
5000);

SavingAct . displayBalance ();  
SavingAct . deposit (1000);  
SavingAct . computeInterest ();  
savingAct . withdraw (2000);

CurrAcct currentAct = new CurrAcct ("Jane Doe",  
9876543210,

```
currentAct.displayBalance();  
currentAct.deposit(500);  
currentAct.withdraw(2000);  
currentAct.close();
```

### Output :-

Account Number : 12345

Customer Name : John Doe

Account Type : Savings

Balance : \$ 6000.0

Deposit ~~7000~~ of \$ 1000.0 credited

Account number : 123456

Customer Name : John Doe

Interest of \$ 3000.0 credited

Account number : 987654

Customer Name : Jane Doe

Account Type : Current

Balance : \$ 2000.0

Inufficient funds.

Account Number : 987654

Customer Name : Jane Doe

Balance : \$ 1950.0

1950.0  
1950.0  
1950.0

Q. 1) 16/02/2024

Q. 1) Create a package CIE which has two classes - Student and Internals. Student has members like usn, name. The class Internals derived from student has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of 'n' students in all five courses.

Ans =

```
package CIE;
import java.util.Scanner;
public class Student {
    protected String usn = new String();
    protected String name = new String();
    protected int mark;
```

```
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("Line " + ~~num~~);  
item ~~no~~ = item.nextInt();

Date \_\_\_\_\_  
Page \_\_\_\_\_

}  
public void displayStudentDetails() {

System.out.println("The usn is: " + usn);

System.out.println("The name is: " + name);

System.out.println("The sem is: " + sem);

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.println("Enter marks for  
course " + (i+1) + ": ");

marks[i] = sc.nextInt();

}

~~package~~

package SSE;

import CIE.intervals;

import java.util.Scanner;

public class External extends Intervals {

protected int marks[];

protected int finalMarks[];

public ~~External~~ () ;  
marks = new ~~int~~ int [5];  
finalMarks = new int [5];

public void input SEE marks () ;

Scanner sc = new Scanner (System.in);  
for (int i=0; i<5; i++) {

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

marks [i] = sc.nextInt();

public void calcFinalMarks () {

for (int i=0; i<5; i++) {

finalMarks [i] = ~~marks[i]/2~~;

+ super.marks [i];

Package SEE;

Public class External Extends Internal

public External ;

Public class Main {

Public static void main (String args) {

```
for (int i=0; i<5; i++)
```

```
    finalMarks[i] = new English();
```

```
    finalMarks[i] = new English();
```

```
    finalMarks[i].Input();
```

```
System.out.println("Display marks")  
for (int i=0; i<5; i++) {
```

```
    finalMarks[i].calFinalMarks();
```

```
}
```

```
}
```

Q.) Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and the derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age < 0. In son class, implement a constructor that takes both father and son's age and throws an exception if son's age is  $\geq$  father's age.

Ans) import java.util.Scanner;  
class WrongAge extends Exception {  
 public WrongAge() {

public WrongAge (String message),  
super (message);  
}  
}

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

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's age : " + fatherAge);

class Son extends Father {  
protected int sonAge;  
public Son() throws WrongAgeException {  
sonAge = readInt();  
super();  
}

if (sonAge >= super.FatherAge) {

throw new WrongAge("Son's age  
cannot be greater than father's  
age");

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

try {

Son son = new Son();  
son.display(); } }

Catch (Wrong Age e) of

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

Q.) Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every second and another displaying once every two seconds.

Ans)

```
public class Main {
    public static void main(String[] args) {
        Thread thread1 = new Thread() {
            while (true) {
                System.out.println("BMS College of Engineering");
                try {
                    Thread.sleep(10000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        };
        Thread thread2 = new Thread() {
            while (true) {
                System.out.println("CSB");
                try {

```

```
Thread thread2 = new Thread() {
            while (true) {

```

```
System.out.println("CSB");
            try {

```

Thread.sleep(2000);

} catch (InterruptedException e) {  
e.printStackTrace();

}

thread1.start();

thread2.start();

Output :-

BMS College of Engineering

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS College of Engineering

✓ ✓ ✓  
✓ ✓ ✓  
16.0%

Q.1.) Creating Label, button and Textfield  
in a Frame using AWT.

Ans:-

```
import java.awt.*;
import java.awt.event.*;
public class AWTExample extends Frame {
    Label l;
    Button b;
    Textfield t;
    AWTExample() {
        b = new Button("submit");
        l = new Label("Employee info");
        t = new Textfield("Employee ID");
        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 awt_obj = new AWTExample();
}
```

Create a button and add a action listener for mouse click.

Ans →

```
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 e) {  
tf. setText ("Welcome");  
}  
public void windowClosing (WindowEvent e) {  
System.exit (0);  
}  
public static void main (String args []) {  
new EventHandling ();  
}
```

J.S.B  
23.02.24

Examples on I/O :-

Example → (1) :

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

Special character is : " + ch );

Example → (2) :

```
import java.io.*;  
public class ByteArrayInput {  
    public static void main (String args) throws IOException {  
        byte [] buf = {35, 36, 37, 38};  
        ByteArrayInputStream byt = new ByteArrayInputStream  
        InputSt
```

```
public class FileTy {  
    public static void main (String args) throws  
    IOException {  
        FileInputStream fin = new
```

```

FileInputStream fin = new FileInputStream("Example.txt");
int content;
System.out.println("Remaining bytes that can
be read: " + fin.available());
content = fin.read();
System.out.print((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());
}

```

Example → ③ :-

```

import java.io.FileInputStream;
import java.io.IOException;
public class FileEx2 {
    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.print("Bytes read: ");~~

```

for (byte b : bytes) {
    c = (char) b;
}

```

System.out.print(c);  
    }  
}

Example → (4) :-

```
import java.io.*;  
public class ByteArray_ex1  
{  
    public static void main (String [] args)  
throws Exception {  
    FileOutputStream fout1 = new  
        FileOutputStream ("Example1.txt");  
    FileOutputStream fout2 = new  
        FileOutputStream ("Example2.txt");  
  
    ByteArrayOutputStream bout = new  
        ByteArrayOutputStream ();  
  
    bout.write (65);  
    bout.writeTo (fout1);  
    bout.writeTo (fout2);  
    bout.flush ();  
    bout.close ();  
    System.out.println ("Success...");  
}
```

Output :-

Eg (1) :

~~2 bytes~~

ASCII of char: 35. special char #  
ASCII of char: 36 special char \$  
ASCII of char: 37 special char %  
ASCII of char: 38 special char &

Eg: ~~(2)~~ :-

Remaining bytes that can be read: 2

~~Remaining bytes that can be read: 2~~

104

105

Remaining bytes that can be read: 0

Eg: ~~(3)~~ ~~(4)~~ :-

~~Number of bytes read: 2~~

Number of bytes read: 2  
Bytes read: hi

hi  
world