

I N D E X

NAME: Rohit Rajput STD.: QSE SEC.: D ROLL NO.: _____ SUB.: OJ-LPB observation

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	26 Sept 25	Prints of Q.E	1,2	X 10/10
2.	10 Oct 25	To create a class Student & using method accept, display & calculate the SGPA of a student	3,4	X 10/10
3	24 Oct 25	To create a class BookDemo & using ToString() display the complete details	5 to 7	X 24/10
4.	7 Nov 25	To create an abstract class Shape & make 3 classes Rectangle, Triangle, Circle	8,9	X 7/11/25
5.	11 Nov 25	To create a class Bank & display details	10-14	S. Printed 7/11/25
6.	11 Nov 25	To create a package Percentage & display marks of student	15-18	S. Printed 14/11/25
7.	18 Nov 25	To demonstrate exception handling	19,20	S. Printed 27/11/25
8.	12 Dec 2025	To demonstrate working of Threads using display messages in two threads with time delay	21,22	S. Printed 12/12/25

①

Prog 1: Roots of QE

```

import java.util.Scanner;
public class Quadratic
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.println ("Enter the coefficient of a,b,c");
        double a = input.nextDouble();
        double b = input.nextDouble();
        double c = input.nextDouble();
        System.out.println ("Values Entered");
        if (d>0)
        {
            double r1 = (-b + Math.sqrt(d)) / (2*a);
            double r2 = (-b - Math.sqrt(d)) / (2*a);
            System.out.println ("The roots are real & distinct");
            System.out.println ("Root1 = " + r1);
            System.out.println ("Root2 = " + r2);
        }
        else if (d==0)
        {
            double r1 = -b/(2*a);
            double r2 = r1;
            System.out.println ("Root1 = " + r1);
            System.out.println ("Root2 = " + r2);
        }
        else
        {
            double realPart = -b/(2*a);
            double imgPart = Math.sqrt(-d)/(2*a);
            System.out.println ("The roots are complex");
            System.out.println ("Root1 = " + realPart + "+" + imgPart + "i");
            System.out.println ("Root2 = " + realPart + "-" + imgPart + "i");
        }
        input.close();
    }
}

```

OUTPUT

① ⇒ Enter the coefficient of a

1

Enter the coefficient of b

-5

Enter the coefficient of c

6

values Entered

The Roots of the equation are real & distinct

Root 1 : 3.0

Root 2 : 2.0

② ⇒ Enter the coefficient of a

1

Enter the coefficient of b

-4

Enter the coefficient of c

4

values Entered

The Roots of the equation are real & equal

Root 1 = 2

Root 2 : 2

③ ⇒ Enter the coefficient of a

1

Enter the coefficient of b

4

Enter the coefficient of c

5

values Entered

The Roots of the equation are imaginary

Root 1 : -2 + i

Root 2 : -2 - i

(3)

Prog 2: To create a class Student & using method accept, display & calculate SGPA of a student.

```

import java.util.*;
class Student {
    public static int credit(int score) {
        int points;
        if (score > 90)
            points = 10;
        else if (score >= 80)
            points = 9;
        else if (score >= 70)
            points = 8;
        else if (score >= 60)
            points = 7;
        else if (score >= 50)
            points = 6;
        else
            points = 0; // or assign a failing grade
        return points;
    }

    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int credsum=0;
        int Pointsum=0;
        float SGPA;
        System.out.println ("Enter name and usn");
        String name = sc.nextLine();
        String USN = sc.nextLine();
        System.out.println ("Enter the number of subjects : ");
        int a = sc.nextInt();
        int cred [] = new int [a];
        int marks [] = new int [a];
        for (int i=0 ; i<a ; i++) {
            System.out.println ("Enter sub " + i + " credits");
            cred [i] = sc.nextInt();
            credsum += cred [i];
            System.out.println ("Enter marks obtained");
            marks [i] = sc.nextInt();
            Pointsum += credits (marks [i]) * cred [i];
        }
    }
}

```

SGPFI = PointSum / credSum;
System.out.println ("SGPFI is "+SGPFI);

3
3

OUTPUT

Enter name and USN :

Rohit
1BF24EC540

Enter number of subjects :

5

Enter subject 1 credits :

4

Enter marks obtained :

98

Enter marks subject 2 credits :

3

Enter marks obtained :

97

Enter subject 3 credits :

3

Enter marks obtained :

87

Enter subject 4 credits :

2

Enter marks obtained :

89

Enter subject 5 credits :

4

Enter marks obtained :

91

SGPFI = 9.69

Rohit
16/10/25

⑤

Program 38 Create a class Book which contains four members : name, author, price, num-pages. Include a constructor to set the values of the members. Include methods to set & get the details of the objects. Include methods to set & get the details of. Include a toString() method that could display the complete details of the book. Develop a Java program to create n books objects.

```

import java.util.Scanner;
class Book {
    private String name;
    private String author;
    private double price;
    private int numPages;

    public Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

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

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

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

    public void setNumPages(int numPages) {
        this.numPages = numPages;
    }

    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public int getNumPages() {
        return numPages;
    }

    public String toString() {
        return "Book name: " + name + " Author: " + author +
            " Price: " + price + " Number of Pages: " + numPages +
            "\n";
    }
}

```

3

3

```
public class BookDemo {  
    public static void main (String [] args) {  
        Scanner sc = new Scanner (System.in);  
        System.out.print ("Enter no. of books : ");  
        int n = sc.nextInt ();  
        sc.nextLine ();  
        Book [] books = new Book [n];  
        for (int i = 0; i < n; i++) {  
            System.out.println ("Enter details of Book " + (i+1) + ":");  
            System.out.print ("Name: ");  
            String name = sc.nextLine ();  
            System.out.print ("Author: ");  
            String author = sc.nextLine ();  
            System.out.print ("Price: ");  
            String price = sc.nextLine ();  
            System.out.print ("Number of pages: ");  
            int pages = sc.nextInt ();  
            sc.nextLine ();  
            books[i] = new Book (name, author, price, pages);  
        }  
        System.out.println ("\n-- Book Details --");  
        for (Book b : books) {  
            System.out.println (b.toString());  
        }  
        sc.close();  
    }  
}
```

7

OUTPUT

Enter number of books : 2

Enter details of Book 1 :

Name : HC VERMA

Author : HC VERMA

Price : 1500

Number of Pages : 700

Enter details of Book 2 :

Name : S CHAND

Author : S CHAND

Price : 700

Number of Pages : 400

-- Book Details --

Book Name : HC VERMA

Author : HC VERMA

Price : 1500.0

Number of Pages : 700

Book Name : S CHAND

Author : S CHAND

Price : 900.0

Number of Pages : 400

R
24/10/25

Program 4 Develop a Java Program to create an abstract class named Shape that contains two integers and empty method named pointArea(). Provide three classes named Rectangle, Triangle & Circle such that each one of the class extends the class Shape. Each one of the classes contain only the method pointArea() that prints the area of the given shape.

Code 8

```
import java.util.Scanner;  
abstract class Shape {  
    int a, b;  
    abstract void pointArea();  
}  
  
class Rectangle extends Shape {  
    Rectangle (int x, int y) {  
        a=x;  
        b=y;  
    }  
    void pointArea () {  
        System.out.println ("Area of Rectangle : " + (a*b));  
    }  
}  
  
class Triangle extends Shape {  
    Triangle (int x, int y) {  
        a=x;  
        b=y;  
    }  
    void pointArea () {  
        System.out.println ("Area of triangle : " + (0.5*a*  
    }  
}  
  
class Circle extends Shape {  
    Circle (int r) {  
        a = r;  
    }  
    void pointArea () {  
        System.out.println ("Area of circle : " + (math.pi*  
    }  
}
```

(9)

public class Main {

public static void main (String [] args) {

Scanner in = new Scanner (System.in);

System.out.println ("Enter length & breadth of rectangle:");

int l = in.nextInt();

int b = in.nextInt();

Rectangle rect = new Rectangle (l, b);

System.out.println ("Enter base & height of triangle: ");

int base = in.nextInt();

int height = in.nextInt();

Triangle tri = new Triangle (base, height);

System.out.println ("Enter the radius of circle: ");

int r = in.nextInt();

Circle circ = new Circle (r);

rect.printArea();

tri.printArea();

cir.printArea();

sc.close();

}

}

OUTPUT:Enter length and breadth of rectangle: 20
30Enter the base & height of triangle: 40
30

Enter the radius of circle: 50

Area of Rectangle: 600

Area of Triangle: 600.0

Area of Circle: 7853.981633974483

7/11/25

10) Program 5: Develop a Java Program to create a class Bank that maintains 2 kinds of account for its customers, one called Savings Account & the other Current Account. The savings account provides compound interest & withdrawal facility but no cheque book facility. The current Account provides check cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Current and Sav-Acc to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customer & update the balance
- b) Display the balance.
- c) Compute and deposit interest
- d) Permit withdrawal and update the Balance.

Check for the minimum balance, impose penalty if necessary and update the balance.

Code:

```
import java.util.Scanner;  
class Account {  
    protected String customerName;  
    protected int accountNumber;  
    protected String accountType;  
    protected double balance;  
  
    public Account (String name, int accNo, String accType, double initialBalance){  
        this.customerName = name;  
        this.accountNumber = accNo;  
        this.accountType = accType;  
        this.balance = initialBalance;  
    }  
  
    public void deposit (double amount){  
        if (amount > 0){  
            balance += amount;  
            System.out.println ("Amount deposited successfully.");  
        } else {  
            System.out.println ("Invalid deposit amount");  
        }  
    }  
}
```

```

public void displayBalance() {
    System.out.println("Account Holder: " + customerName);
    System.out.println("Account Number: " + accountNumber);
    System.out.println("Account Type: " + accountType);
    System.out.println("Account Balance: $ " + balance);
}

```

```

public void withdraw(double amount) {
}

```

```

class SavAcct extends Account {
}

```

```

    private static final double interestRate = 0.05;

```

```

    public SavAcct(String name, int accNo, double initialBalance) {
    }

```

```

        super(name, accNo, accountType == "Savings", initialBalance);
    }

```

```

    public void computeAndDepositInterest(int years) {
    }

```

```

        double interest = balance * Math.pow(1 + interestRate, years)
                        - balance;
    }

```

```

        balance = balance + interest;
    }

```

```

        System.out.println("Interest of $" + String.format("%.2f", interest)
                            + " added for " + years + " years" + " year(s).");
    }

```

@Override

```

    public void withdraw(double amount) {
    }

```

```

        if (amount <= 0) {
    }

```

```

            System.out.println("Invalid withdrawal amount!");
            return;
        }
    }

```

```

        if (amount > balance) {
    }

```

```

            System.out.println("Insufficient Funds!");
        }
    }

```

```

    } else {
    }

```

```

        balance -= balance - amount;
    }

```

```

        System.out.println("Withdrawal of $" + amount +
                            " Successful.");
    }
}

```

}

```

class CustAcct extends Accounts {
}

```

```

    private static final double minBalance = 1000.0;

```

```

    private static final double serviceCharge = 100.0;

```

```

    public CustAcct(String name, int accNo, double initialBalance) {
    }

```

```

        super(name, accNo, "Customer", initialBalance);
    }
}

```

}

```

⑫ private void checkMinimumBalance() {
    if (balance < minBalance) {
        balance -= service_charge;
        System.out.println ("Balance below minimum! & " + service
                           + " service charge imposed.");
    }
}

@Override
public void withdraw (double amount) {
    if (amount <= 0)
        System.out.println ("Invalid withdrawal amount!");
    else
        return;
}

if (amount > balance)
    System.out.println ("Insufficient funds");
else
    balance -= amount;
    System.out.println ("Withdrawal of " + amount + " success
checkMinimumBalance());
}

public class Bank {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("--- Welcome to Bank System ---");
        System.out.print ("Enter customer Name : ");
        String name = sc.nextLine();
        System.out.print ("Enter Account Number : ");
        int accNo = sc.nextInt();
        System.out.print ("Enter Account Type (1 for Savings
                           & for Current) : ");
        int type = sc.nextInt();
        System.out.print ("Enter Initial Deposit : ");
        double balance = sc.nextDouble();
        Account acc;
        if (type == 1)
            acc = new SavAccount (name, accNo, balance);
        else
            acc = new CurrAcct (name, accNo, balance);
        int choice;
        do {
            System.out.println ("In-- menu--");
            System.out.print ("1. Deposit");
        }
    }
}

```

(13) 5

```

System.out.println ("2. withdraw");
System.out.println ("3. Display Balance");
if (acc instanceof SavAcc)
    System.out.println ("4. Compute and deposit interest");
else
    System.out.println ("5. Exit");
System.out.println ("Enter your choice");
choice = sc.nextInt();

switch (choice) {
    case 1:
        System.out.println ("Enter amount to deposit: £");
        acc.deposit (sc.nextDouble());
        break;
    case 2:
        System.out.println ("Enter amount to withdraw: £");
        acc.withdraw (sc.nextDouble());
        break;
    case 3:
        acc.displayBalance ();
        break;
    case 4:
        if (acc instanceof SavAcc) {
            System.out.println ("Enter no. of years for interest
                calculation:");
            int years = sc.nextInt();
            ((SavAcc) acc).computeAndDepositInterest (years);
        }
        else
            System.out.println ("Interest calculation not applicable
                for current Account.");
        break;
    case 5:
        System.out.println ("Thankyou for using the Bank
            System");
        break;
    default:
        System.out.println ("Invalid choice! Try Again.");
}
}

while (choice != 5);
sc.close();
}

```

(14)

OUTPUT

-- Welcome to the Bank System --

Enter customer Name : Raj

Enter Account Number : 1234

Enter Account Type (1 for Savings / 2 for Current) : 1

Enter Initial deposit : ₹1000

-- Menu --

1. Deposit
2. Withdraw
3. Display Balance
4. Compute & deposit Interest
5. EXIT

Enter your choice : 2

Enter the amount to withdraw : ₹1000

Withdrawal of ₹1000.0 successful.

-- Menu --

1. Deposit
2. Withdraw
3. Display Balance
4. Compute & deposit Interest
5. Exit

Enter your choice : 5

Thank you for using the Bank System.)

S. Puri

Program 6: Create a package CIF which has two classes - Personal and Internal. The class Personal has members like usn, name, sem. The class Internal 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 derived class of Personal. 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.

sol \Rightarrow Ciepackage/

CIF /

Personal.java

Internal.java

SEE /

External.java

Code ?

CIF:

Ciepackage/CIF/Personal.java

Package CIF;

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;

}

}

Ciepackage/CIF/External.java

Package CIF;

Public class Personal {

 public String usn;

 public String name;

 public int sem;

 public P

⑯ CIEpackage / CIE / Internals.java

```
package CIE;
public class Internals {
    public int internalMarks [] = new int [5];
    public Internals (int marks []) {
        for (int i=0; i<5; i++) {
            internalMarks [i] = marks [i];
        }
    }
}
```

SEE:

CIEpackage / SEE / External.java

```
package SEE;
import CIE.Personal;
public class External extends Personal {
    public int seeMarks [] = new int [5];
    public External (String usn, String name, int sem, int sec) {
        super (usn, name, sem);
        for (int i=0; i<5; i++) {
            this.seeMarks [i] = sep [i];
        }
    }
}
```

main:

CIEpackage / FinalMarks.java

```
import CIE.*;
import SEE.*;
import java.util.Scanner;
public class FinalMarks {
    Scanner sc = new Scanner (System.in);
    System.out.print ("Enter number of students : ");
    int n = sc.nextInt();
    Internals internal [] = new Internals [n];
    External external [] = new External [n];
    for (int i=0; i<n; i++) {
        System.out.println ("Enter details " + (i+1));
        System.out.print ("USN : ");
        String usn = sc.next();
        System.out.print ("Name : ");
        String name = sc.next();
        internal [i] = new Internals (internalMarks);
        external [i] = new External (usn, name, sem, sec);
    }
}
```

```

systems.out.println ("Semester :");
int sem = sc.nextInt();

int internalmarks []= new int [5];
System.out.println ("Enter 5 Internal marks");
for (int j=0; j<5; j++) {
    internalmarks [j] = sc.nextInt();
}

int semmarks []= new int [5];
System.out.println ("Enter 5 Sem marks");
for (int i=0; i<5; i++) {
    System.out.println ("Student :" + external[i].name);
    System.out.println ("USN :" + external[i].usn);
    System.out.println ("Semester :" + external[i].sem);

    System.out.println ("Course 1st CIE 1st Sem 1st Final");
    for (int j=0; j<5; j++) {
        int finalmark = internal[i].internalmarks[j] +
            (external[i].semarks[j]/2);
        System.out.println ((j+1) + "st " +
            internal[i].internalmarks[j] + "st " +
            external[i].semarks[j] + "st " +
            finalmark);
    }
}
sc.close();

```

OUTPUTS

Enter number of students : 2

Enter detail of student 1

USN : 1BF2UUCS 289

Name : Harey

Semester : 04

Enter 5 Internal marks :

48
49
50
46
47

(18)

Enter 5 SEE marks:

99

100

94

96

98

Enter details for Student 2

USN: 1BF24CS098

Name: Ram

Semester: 06

Enter 5 Internal marks:

40

48

50

50

47

Enter 5 SEE marks:

98

99

100

96

95

-- Final marks --

Student: Harry

USN: 1BF24CS289

Semester: 4

Course	CIE	SEE	Final
1	48	99	97
2	49	100	99
3	50	94	97
4	46	96	94
5	47	98	96

S. P. J.
14/1/15

Student: Ram

USN: 1BF24CS098

Semester: 5

Course	CIE	SEE	Final
1	40	98	89
2	48	99	97
3	50	100	100
4	50	96	98
5	47	95	94

Program 7: Write a program that demonstrate handling of exception in inheritance tree. Create a basic class called "Father" and 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 uses both father's and son's age and throws an exception if son's age is \geq father's age.

Code:

```

class WrongAgeException extends Exception {
    public WrongAgeException (String msg) {
        super(msg);
    }
}

class Father {
    int fatherAge;
    Father (int age) throws WrongAgeException {
        if (age<0) {
            throw new WrongAgeException ("Father's age cannot be negative");
        }
        this.fatherAge = age;
        System.out.println ("Father's age : " + fatherAge);
    }
}

class Son extends Father {
    int sonAge;
    Son (int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAgeException ("Son's age cannot be >= Father's age");
        }
        if (sonAge<0) {
            throw new WrongAgeException ("Son's age cannot be negative");
        }
        this.sonAge = sonAge;
        System.out.println ("Son's age : " + sonAge);
    }
}

public class Main {
    public static void main (String [] args) {
        try {
            Son s = new Son (40, 20);
        } catch (WrongAgeException e) {
            System.out.println ("Exception caught : " + e.getMessage());
        }
    }
}

```

(2)

Output

→ Father's age : 40
Son's age : 20

→ Exception caught : Father's age cannot be negative

→ Father's age : 40

Exception caught : Son's age cannot be negative.

→ Father's age : 40

Exception caught : Son's age cannot be \geq Father's age.

S. Pratul

Program: 8

Write a programs which creates two threads, one thread displaying "BMS College of Engineering" once every 10 seconds and another displaying "CSE" once every two seconds

Code:

```

class CollegeThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS college of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread Interrupted");
        }
    }
}

class CSEThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread Interrupted");
        }
    }
}

public class Main {
    public static void main (String [] args) {
        CollegeThread t1 = new CollegeThread();
        CSEThread t2 = new CSEThread();
        t1.start();
        t2.start();
    }
}

```

(22)

OUTPUT

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

Solved
12/12/25