VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Abhishek patil (1BM23CS013)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019
Sep-2024 to Jan-2025

B.M.S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **Abhishek patil (1BM23CS013),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Geetha N Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Sl. No.	Date	Experiment Title	Page No.
1	9/10/24	Quadratic equation	4
2	16/10/24	SGPA calculator	9
3	23/10/24	Book program	16
4	23/10/24	Abstract class shape program	20
5	13/11/24	Bank program	24
6	13/11/24	Packages	32
7	20/11/24	Exception handling	39
8	27/11/24	Multithreading	44
9	27/11/24	Integer division with user interface	47
10	27/11/24	Inter process communication and deadlock	50

Github Link:

https://github.com/abhishekpatil-bms/java

Program 1

Implement Quadratic Equation

Algorithm

Date 9/10/24

lab program - 1

1 Q) Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a,b,c. cover all cases

→ class Quadratic

→ import java.util.scanner;

```
class Quadratic_eq {
    public static void main(string[] args)
    {
        int a,b,c;
        double r1,r2,d;

        Scanner s = new Scanner(System.in);

        System.out.println("Enter coefficient a=");
        a = s.nextInt();
        System.out.println("Enter coefficient b=");
        b = s.nextInt();
        System.out.println("Enter coefficient c=");
        c = s.nextInt();

        d = b*b-(4*a*c);

        if (d>0)
        {
            System.out.println("Roots are Real and unique");
            r1 = (-b + Math.sqrt(d))/(2*a);
            r2 = (-b - Math.sqrt(d))/(2*a);
            system.out.println("Root 1 = %.2f \n
                                Root 2 = %.2f", r1,r2);
        }
        else if (d==0)
        {
            r1 = -b/(2*a);
            system.out.println("Roots are Real and equal \n
                                Root 1 = %.2f \n Root 2 = %.2f",
                                r1,r1);
        }
        else
        {
            r1 = -b/(2*a);
            r2 = Math.sqrt(-d)/(2*a);
            system.out.println("Roots are Imaginary and
                unique \n Root 1 = %.2f + %.2f i \n
                Root 2 = %.2f - %.2f i ", r1,r2,r1,r2);
        }

        s.close
    }
}
```

Output :

Enter a = 2
Enter b = 4
Enter c = 2
Root are real and equal
Root 1 = -1.00
Root 2 = -1.00
Enter a = 2
Enter b = 8
Enter c = 2
Root are real and unique
Root 1 = -0.27
Root 2 = -3.73

Enter a = 2
" b = 3
c = 8
Root are imaginary unique
Root 1 = 0.00+1.85i
Root 2 = 0.00-1.85i

o/p seen
9/10/24

Code:

```
import java.util.Scanner;
public class Quadratic_eq{
   public static void main(String args[]){
      Scanner s = new Scanner(System.in);
      double r1,r2;
      System.out.println("NAME :
ABHISHEK PATIL\nUSN :
1BM23CS013");
      System.out.print("Enter a = ");
      int a = s.nextInt();
      System.out.print("Enter b = ");
      int b = s.nextInt();
      System.out.print("Enter c = ");
      int c = s.nextInt();
      double d = (b*b)-4*a*c;
      if( d > 0){
         r1 = (-b+Math.sqrt(d))/(2*a);
         r2 = (-b-Math.sqrt(d))/(2*a);
         System.out.print("Roots are real
and unique!");
         System.out.printf("%n r1 =
%.2f%n r2 = %.2f%n", r1, r2);



      }else if(d==0){
         r1 = -b/(2*a);
         r2 = r1;
         System.out.println("Roots are real
and equal!");
         System.out.printf("r1 = %.2f /nr2
= %.2f",r1,r2);
      }else{
         System.out.println("Roots are
imaginary1");
         r1 = -b/(2*a);
```

```
        r2 = Math.sqrt(-d)/(2*a);
        System.out.printf("r1 =
%.2f+%.2fi\n",r1,r2);
        System.out.printf("r2 = %.2f-
%.2fi",r1,r2);
      }
   s.close();
   }
}
```

Output:

```
[Abhi$javac Quadratic_eq.java
[Abhi$java Quadratic_eq
 NAME : ABHISHEK PATIL
 USN : 1BM23CS013
 Enter a = 2
 Enter b = 4
 Enter c = 2
 Roots are real and equal!
[r1 = -1.00 /nr2 = -1.00%
 Abhi$java Quadratic_eq
 NAME : ABHISHEK PATIL
 USN : 1BM23CS013
 Enter a = 2
 Enter b = 82
 Enter c = 2
 Roots are real and unique!
  r1 = -0.02
  r2 = -40.98
[Abhi$java Quadratic_eq
 NAME : ABHISHEK PATIL
 USN : 1BM23CS013
 Enter a = 2
 Enter b = 3
 Enter c = 8
 Roots are imaginary1
 r1 = 0.00+1.85i
 r2 = 0.00-1.85i%
```

Program 2

SGPA Calculator

Algorithm:

lab program - 2

Q) Develop a java program to create a class student with members usn, name, an array credits & an array marks include methods to accept & display details & a method to calculate SGPA of a student.

```
import java.util.scanner;

class Student {
        string usn,name;
        int credits[8], marks[8], total_marks,
            total_credits, grades[8];
        double sgpa;

        scanner sc= new scanner(System.in);

        void details()
        {
            scanner sc= new scanner(System.in)
            System.out.println("Enter name: ");
            name = sc.nextline();
            system.out.println("Enter USN: ");
            usn = sc.nextLine();
```

```
System.out.println("Enter credit ~~marks and~~
        marks of each subject respectively");

for(int i=0; i<=7; i++){
        system.out.println("subject"+(i+1)":\n");
credits[i] = sc.nextInt();
marks[i] = sc.nextInt();
} }
void grade(){

for(int i=0; i<=7; i++)
{
        if(marks[i]==100)
        {
                grades[i]=10;
        }
        else {
                grades[i] = ((marks[i]%10)+1);
        }
}
void sgpa_cal(){
        int sum=0

        for(int i=0; i<=7; i++){

                sum+= (grades[i] * credits[i]);
        }
        sgpa = double(sum)/20.0;
}
```

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

        void display ()
        {
            System.out.println("Name :"+name+
                "\n USN:"+usn );
            system.out.println("SGPA :"+sgpa );
        }
}

class Run{
    public static void main(String[] args){

        Student[] students = new Student[3]
        for(int i=0; i<=2; i++){
            system.out.println("Student "+(i+1)":");
            students[i].details();
            students[i].grade();
            students[i].sgpa_cal();
            students[i].display();
        }
}
```

Seen execute

Output:
Enter name : Abhishek pati
USN : 1BM23CS013
Enter credit and marks respectively
Subject 1
4

Code:

```
import java.util.Scanner;
class Students {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.printf("How many students : ");
        int n = s.nextInt();
        Student[] studs = new Student[n];
        for(int i = 0; i<n ; i++){
            studs[i] = new Student();
            System.out.println("Enter student "+(1+i)+" details:\n");
```

```
            studs[i].get_details();
            studs[i].calculate_grades();
            studs[i].calculate_sgpa();
            studs[i].display();
            System.out.println("\n ----------------------------\n");
        }
        s.close();
    }
}
public class Student {
        String name,usn;
        double sgpa;
        int[] marks = new int[8];
        int[] credits = new int[8];
        int[] grades = new int[8];

        void get_details(){
            Scanner s = new Scanner(System.in);
            System.out.printf("Enter Student name: ");
            name = s.nextLine();
            System.out.printf("Enter USN: ");
            usn = s.nextLine();
            System.out.printf("Enter marks of each subject and their credits respectively: \n\n");
            for(int i=0; i<8 ; i++){
                System.out.printf("sub "+(i+1)+" marks:");
                marks[i] = s.nextInt();
                System.out.printf("sub "+(i+1)+" credit: ");
                credits[i] = s.nextInt();
                System.out.println();
            }
            s.close();
        }
        void calculate_grades(){
            for(int i=0; i<8 ; i++){
                if(marks[i] == 100){
                    grades[i] = 10;
                }
                else{
                    grades[i] = (marks[i]/10)+1;
                }

            }
        }
        void calculate_sgpa(){
```

```
        double  a = 0,b = 0;
        for(int i=0 ; i<8  ; i++){
            a = a + (grades[i] * credits[i]);
            b = b + credits[i];
        }
        sgpa = a/b;
    }
    void display(){
        System.out.println("Name :"+name);
        System.out.println("USN :"+usn);
        System.out.printf("SGPA : %.2f",sgpa);

    }
}
```

Output:

```
Abhi$java Students
How many students : 1
Enter student 1 details:

Enter Student name: Abhishek patil
Enter USN: 1BM23CS013
Enter marks of each subject and their credits respectively:

sub 1 marks:77
sub 1 credit: 4

sub 2 marks:76
sub 2 credit: 4

sub 3 marks:72
sub 3 credit: 3

sub 4 marks:66
sub 4 credit: 4

sub 5 marks:70
sub 5 credit: 3

sub 6 marks:68
sub 6 credit: 1

sub 7 marks:89
sub 7 credit: 1

sub 8 marks:73
sub 8 credit: 1

Name :Abhishek patil
USN :1BM23CS013
SGPA :7.809523809523809
```

Program 3

Book Program

Algorithm:

Date 23/10/24

lab program-3

Q) Create a class book which contains four members: name, author, price, num_pages. Include a constructor to set the value for the member. 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.Scanner;

class Books {
        String name;
        String author;
        int price;
        int num_pages;
        Books (String n, String a, int p, int g){
                this.name = n;
                this.author = a;
                this.price = p;
                this.num_pages = g;
        }
        public String toString()
        {
                String na, au, pr, nu;
                na = "Book name: " + this.name + "\n";
                au = "Author : " + this.author + "\n";
```

```
        pr = "Price : " + this.price + "\n";
        nu = "Number of Pages : " + this.num_pages + "\n";
        return na+au+pr+nu;
    }
    void get_details()
    {
        Scanner s = new Scanner(System.in);
        System.
    }

class Books_det()
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        int n;
        String name;
        String author;
        int price;
        int num_pages;

        System.Out.println("Enter How many books : ");
        n = s.nextInt();
        Books b[] = new Books[n];
        for(int i=0; i<n; i++)
        {
            Enter
            System.out.println("Enter book "+(i+1)+
                "details \n");
            System.out.println("Name : ");
            name = s.next();
            System.out.println("Author : ");
            author = s.next();
            System.out.println("Price : ");
            price = s.nextInt();
            System.out.println("Number of Pages : ");
            num_pages = s.nextInt();
        }
        for(int i=0; i<n; i++)
        {
            System.Out.println("Book "+(i+1)+"Details \n");
            System.Out.println("Name : "+name+"\n
                "Author : "+author+"\n Price : "+price+
                "Number of Pages : "+num_pages);
        }
        s.close();
    }
```

Output :

Number of books : 1

Enter book 1 details

Name : legends
Author : Abhiram
Price : 499
Pages : 325

Seen

Code: import java.util.Scanner;

```
class Books {
    String name;
    String author;
    int price;
    int num_pages;

    Books(String n, String a, int p, int g) {
        this.name = n;
        this.author = a;
        this.price = p;
        this.num_pages = g;
```

```
    }

    public String toString() {
        String na, au, pr, nu;
        na = "Name of book: " + this.name + "\n";
        au = "Name of author: " + this.author + "\n";
        pr = "Price of the book: " + this.price + "\n";
        nu = "Number of pages: " + this.num_pages + "\n";
        return na + au + pr + nu;
    }

}

class Books_det {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        String name;
        String author;
        int price;
        int n;
        int numPages;
System.out.println("\nName: Abhishek patil\nUSN: 1BM23CS013\n\n\n");
        System.out.println("Enter number of books: ");
        n = s.nextInt();
        Books b[] = new Books[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter Book " + (i + 1) + " details");
            System.out.println("Name: ");
            name = s.next();
            System.out.println(" Author: ");
            author = s.next();
            System.out.println("Price: ");
            price = s.nextInt();
            System.out.println("Number of pages: ");
            numPages = s.nextInt();
            b[i] = new Books(name, author, price, numPages);

        }
        for (int i = 0; i < n; i++) {
            System.out.println("Details of book " + (i + 1));
            System.out.println(b[i]);
        }
        s.close();
```

```
    }
}
```

```
Name: Abhishek patil
USN: 1BM23CS013



Enter number of books:
2
Enter Book 1 details
Name:
legends
 Author:
Abhiram
Price:
250
Number of pages:
356
Enter Book 2 details
Name:
Aprentice
 Author:
warden
Price:
500
Number of pages:
289
Details of book 1
Name of book: legends
Name of author: Abhiram
Price of the book: 250
Number of pages: 356

Details of book 2
Name of book: Aprentice
Name of author: warden
Price of the book: 500
Number of pages: 289
```

Program 4

Abstract Class shape program

Algorithm

lab program:4

```
abstract class shape{
    double dim1, dim2;

    Shape (double a, double b){
        dim1 = a;
        dim2 = b;
    }
    abstract double printArea();
}

class Rect extends Shape{
    Rect (double a, double b)
    {
        super(a,b);
    }
    double printArea(){
        System.out.printf("The area of the
            rectangle is: "+(dim*dim2)+"\n");
        return 0;
    }
}

class Tri extends Shape{
    Tri(double a, double b){
        super(a,b);
    }
    double printArea(){
        System.out.printf("The area: "+(dim1*
            dim2)/2)+"\n");
        return 0;
}
class cir extends Shape{
    cir (double a, double b){
        super(a,b);
    }
    double printArea(){
        System.out.printf("The area of circle is:"
            +(3.14*dim1*dim1)+"\n");
        return 0;
    }
}

Class Area{
    public static void main(String[] args)
    {
        rect r = new Rect(7,9);
        Tri t = new Tri(5,10);
        Cir c = new (7,15);
        Shape s;
        s.printArea();
        s = t
        s.printArea();
        s=c;
        s.printArea();
    }
```

Output :-

```
The area of the Rectangle is : 63.0
The area of the Triangle is : 25.0
The area of the Circle is : 153.86
```

o/p seen
Gt
23/10/24

Code:

```
abstract class Shape {
   double dim1, dim2;

   Shape(double a, double b) {
      dim1 = a;
      dim2 = b;
   }

   abstract double printArea();
}

class Rect extends Shape {
   Rect(double a, double b) {
      super(a, b);
   }

   double printArea() {
      System.out.printf("The area of the Rectangle is: " + (dim1 * dim2) + "\n");
      return 0;
   }

}

class Tri extends Shape {
   Tri(double a, double b) {
      super(a, b);
   }

   double printArea() {
      System.out.printf("The area of the Triangle is: " + ((dim1 * dim2) / 2) + "\n");
      return 0;
   }
}

class Cir extends Shape {
   Cir (double a, double b) {
      super(a, b);
   }

   double printArea() {
      System.out.printf("The area of the Circle is: " + (3.14 * dim1 * dim1) + "\n");
```

```
        return 0;
    }
}

class Area {
    public static void main(String[] args) {
        System.out.println("\nName: Abhishek paitl\nUSN: 1BM23CS013\n\n");
        Rect r = new Rect(7, 9);
        Tri t = new Tri(5, 10);
        Cir c = new Cir(7, 15);
        Shape s;
        s = r;
        s.printArea();
        s = t;
        s.printArea();
        s = c;
        s.printArea();
    }
}
```

Output

```
Name: Abhishek paitl
USN: 1BM23CS013


The area of the Rectangle is: 63.0
The area of the Triangle is: 25.0
The area of the Circle is: 153.86
```

Program 5

Bank program

Algorithm

lab program - 5

Bank Account

```
→ import java.util.Scanner;

class Account {
        ~~public~~ private String custName;
        private String accNO;
        private double balance;

        public Account (String custName, String
                accNo, double balance) {
                this.custName = custNam;
                this.accNO = accNO;
                this.balance = balance;
        }

        public double getBalance() {
                return this.balance;
        }

        public void deposit (double amount) {
                if (amount >0) {
                        this.balance += amount;
                        System.out.println("The current balance
                                is " + this.balance);
                }
                else {
                        System.out.println("Amount should
                                not be negative");
                }
```

```
public void withdraw (double amount){
    if (amount > 0 && (balance - amount) >= 0){
        this.balance -= amount;
        System.out.println("Withdraw
            successful. Current balance:
            " + this.balance);
    }
    else {
        System.out.println("Withdraw is
            not possible");
    }
}
}

Class SavingsAccount {

    private double InterestRate;
    private Account account;

    public SavingsAccount (String custName, String
        accNo, double balance, double
        interestRate){

        this.interestRate = interestRate;
        this.account = new Account (custName, accNo, balance);
    }

    public void addInterest (){
        double interest = account.getBalance() * this.interestRate;
        account.deposit(interest);
    }
```

```
    public Account getAccount(){
        return account;
    }
}

Class CurrentAccount {
    private double minBalance;
    private Account account;

    public CurrentAccount (String custName, String
        accNo, double balance, double minBalance){

        this.minBalance = minBalance;
        this.account = new Account (custName,
            accNo, balance);
    }

    public void withdraw(double amt){
        if (amt > 0 && (account.getBalance() - amt) >=
            minBalance){
            account.withdraw(amt);
        } else {
            System.out.println("Withdraw is
                not possible");
        }
    }

    public Account getAccount(){
        return account;
    }
}
```

```
public class Bank{
    public static void main(String [] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name:");
        String name = sc.nextLine();
        System.out.println("Enter the account
                number:");
        String acnt = sc.nextLine();

        while(true){
            System.out.println("Enter your choice:");

            double SavingsBalance = sc.nextDouble();
            System.out.println("Enter the interest rate");
            double InterestRate = sc.nextDouble();
            SavingsAccount SavingsAccount = new
                    SavingAccount(name, acnt,
                    SavingsBalance, interestRate);

            SavingsAccount.addInterest();
            break;

        case 2:
            System.out.println("Enter initial balance:");
            double currentBalance = sc.nextDouble();
            System.out.println("Enter minimum balance:");
            double minBalance = sc.nextDouble();
            CurrentAccount currentAccount = new current
                Account(name, acnt, currentBalance, minBalance);
            System.out.println(currentAccount.getAccount().
                    getBalance());
            break;

        case 3:
            System.out.println("Exiting....");
            sc.close();
            return;

        default:
            System.out.println("Invalid choice,
                    please try again:");
        }
    }
}
```

Seen 30/11/24

~~Output~~

1) Savings Account
2) Current Account

Choice: 1

Enter name: Akhilesh
Enter acc num: 1221
Enter initial balance: 1000

Code:

```
import java.util.Scanner;

abstract class Account {
    String customerName;
    int accountNumber;
    double balance;
    String accountType;

    Account(String customerName, int accountNumber, String accountType, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful. New balance: " + balance);
    }

    void display() {
        System.out.println("Balance: " + balance);
    }

    abstract void interest();

    abstract void withdraw(double amount);
}

class SavAcct extends Account {
    double interestRate = 0.05;

    SavAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, "Savings", balance);
    }

    @Override
    void interest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added. New balance: " + balance);
```

```
    }

    @Override
    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

class CurAcct extends Account {
    double minbalance = 1000.00;
    double scharge = 50.00;
    double[] chequetransactions = new double[100];
    int chequeid = 0;

    CurAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, "Current", balance);
    }

    @Override
    void interest() {
        System.out.println("Interest cannot be calculated for current account.");
    }

    @Override
    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            if (balance < minbalance) {
                balance -= scharge;
                System.out.println("Penalty of " + scharge + " has been deducted. The new balance is: " +
balance);
            } else {
                System.out.println("The updated balance is: " + balance);
            }
            chequetransactions[chequeid++] = amount; // Log the transaction
        } else {
            System.out.println("Insufficient balance. The withdrawal amount is greater than balance.");
        }
    }
```

```
    void displayTransaction() {
        for (int i = 0; i < chequeid; i++) {
            System.out.println("Transaction " + (i + 1) + ": " + chequetransactions[i]);
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Abhishek patil \nUSN:1BM23CS048");
        System.out.println("Enter account type (1 for Savings, 2 for Current): ");
        int choice = scanner.nextInt();
        scanner.nextLine();

        System.out.println("Enter customer name: ");
        String name = scanner.nextLine();

        System.out.println("Enter account number: ");
        int accNum = scanner.nextInt();

        System.out.println("Enter initial balance: ");
        double balance = scanner.nextDouble();

        Account account = null;
        if (choice == 1) {
            account = new SavAcct(name, accNum, balance);
        } else if (choice == 2) {
            account = new CurAcct(name, accNum, balance);
        } else {
            System.out.println("Invalid account type selected.");
            return; // Exit if invalid account type
        }

        int exit = 0;

        do {
            System.out.println("Enter the function to be done:");
            System.out
                    .println("1. Deposit\n2. Display Balance\n3. Compute and deposit interest\n4.
Withdrawal\n5. Exit");

            int option = scanner.nextInt();
            switch (option) {
```

```
            case 1:
                System.out.println("Enter deposit amount:");
                double depositAmount = scanner.nextDouble();
                account.deposit(depositAmount);
                break;
            case 2:
                account.display();
                if (choice == 2) {
                    ((CurAcct) account).displayTransaction();
                }
                break;
            case 3:
                account.interest();
                break;
            case 4:
                System.out.println("Enter withdrawal amount:");
                double withdrawAmount = scanner.nextDouble();
                account.withdraw(withdrawAmount);
                break;
            case 5:
                System.out.println("Exiting...");
                exit = 1; // Change exit flag
                break;
            default:
                System.out.println("Invalid option.");
        }
    } while (exit == 0); // Loop until exit flag is set

    scanner.close();
  }
}
```

Output:

```
D:\CS013>javac Bank.java

D:\CS013>java Bank
Abhishek patil
USN:1BM23CS048
Enter account type (1 for Savings, 2 for Current):
1
Enter customer name:
Abhishek patil
Enter account number:
212
Enter initial balance:
10000
Enter the function to be done:
1. Deposit
2. Display Balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
3
Interest added. New balance: 10500.0
Enter the function to be done:
1. Deposit
2. Display Balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
2
Balance: 10500.0
Enter the function to be done:
1. Deposit
2. Display Balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
```

Program 6

Packages

Algorithm

lab- 6

Q) Create a package CIE which has two classes - Student and Internals. The class Student has members usn, name and sem. The class Internals derived from Student has an array that stores internals marks ~~stored~~ by a st scored in 5 subjects of the current semester.
Create another package SEE which has the class ~~Exte~~ External which is a ~~der~~ derived class of student. This class has an array that stores the SEE marks scored in 5 courses of the current semester of the student. Import the 2 packages in a file that declare the final marks of n students in all five courses.

CIE.

Q Student.java

→ package CIE;

```
public class Student {
        ~~prot~~ String usn;
        String name;
        int sem;
        public Student (String usn, String name,
                        int sem) {
            this.usn = usn;
            this.name = name;
            this.sem = sem;
        }
```

```
    public void displayDetails()
    {
        System.out.println("USN" + usn);
        System.out.println("Name :" + name);
        System.out.println("semester:" + sem);
    }
}
```

-> CIE/Internals

```
package CIE;
import java.util.Scanner;

public class Internals extends Student{
    private
    int [] internalsMarks;

    Internals(String usn, String name, int sem,
            int[] internalMarks){

        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }

    public void displayInternalMarks(){

        System.out.println("Internal Marks:");
        for (int i=0; i < internalsMarks.length; i++)
        {
            System.out.println("Course"+(i+1)+
                    ":" + internalMarks[i]);
        }
    }
    public int calculateInternalTotal(){
        int total=0
        for (int mark : internalMarks){
            total += mark;
        }
        return total;
    }

    public int calculateInternalTotal(){
        int total = 0;
        for (int mark : internalMarks) {
            total += mark;
        }

    public static int[] takeInternalMarksInput(){
        Scanner scanner = new Scanner(System.in);
        int[] marks = new int[5];
        System.out.println("Enter internal marks
                for 5 courses:");
        for (int i=0; i<marks.length; i++){

            System.out.print("Course" +(i+1)+":");
            marks[i] = scanner.nextInt();
        }
        return marks;
    }
}
```

5) SEE / External

```
package SEE;

import CIE.Student;
import java.util.Scanner;

public class External extends Student {
        private int[] external Marks;

        public External (String usn, String name,
                int sem, int[] externalMarks){

            super(usn, name, sem);
            this.externalMarks = externalMarks;
        }

    public void displayExternal Marks(){
            System.out.println("External Marks:");
            for(int i=0; i<external Marks.length;i++)
            {
                System.out.println("Course"+(i+1)+":"+
                        external Marks[i]);
            }
    }

public int calculateExternalTotal(){
        int total = 0
            for(int mark : external Marks) {
                    total += mark;
            }
            return total;
}
```

```
public static int[] takeExternalMarksInput(){

        Scanner scanner = new Scanner(System.in);
        int[]marks = new int[5];
        System.out.println("Enter External Marks:");
        for (int i=0; i<marks.length;i++)
            {
                System.out.print("Course"+(i+1)+":");
                marks[i] = scanner.nextInt();
            }
            return marks;
        }
    }
```

5) Main

```
import CIE.Internals;
import SEE.Externals;
import java.util.Scanner;

class Main {
        public static void main(String[] args) {

            Scanner scanner = new Scanner(System.in);

            System.out.println("Enter details:");
            System.out.print("Enter USN:");
            String usn = scanner.nextline();
            int sem = scanner.nextInt();
            Scanner.nextLine();
```

```
int[] internalMarks = internals.takeInternalMarkInput();

Internal internals = new Internals(usn, name, sem, internalMarks);
internals.display();
internals.marks();

int[] externalMarks = External.takeExternalMarkInput();

external external = new external(usn, name, sem);
external.display();

int total = internals.calculateInternalTotal();
int total_marks = external.calculateExternalTotal();
System.out.println("Total Internal: " + total);
System.out.println("External: " + total_marks);
System.out.println("Final: " + (total + total_marks));
```

Output:

```
Enter Student details:
Enter USN : 1BM23CS013
Enter name : Abhishek patil
Enter Semester : 3
Enter Internal marks for 5 courses:
course 1 : 89
" 2 : 70
" 3 : 67
" 4 : 98
" 5 : 85
```

```
Enter External mark for 5 course:
Course 1 : 88
2 : 89
3 : 90
4 : 87
5 : 90
```

Seen

Total Internal Marks : 410
Total External marks : 434
Final : 844

Code:

```
package CIE;
import java.util.Scanner;

public class Student {
        protected String usn = new String();
        protected String name = new String();
        protected int sem;
```

```
		public void inputStudentDetails(){
			Scanner sc = new Scanner(System.in);
			System.out.println("Enter the name of the student \n");
this.name=sc.nextLine();
			System.out.println("Enter the USN of the student \n");
this.usn=sc.nextLine();
			System.out.println("Enter the semester the student is studying in \n");
			this.sem=sc.nextInt();
		}

		public void displayStudentDetails(){
System.out.println("Name: " + this.name);
			System.out.println("USN: " +this.usn);
			System.out.println("Semester: " +this.sem);
		}
}

package CIE;
import java.util.Scanner;

public class Internals extends Student {
		protected double ciemarks[] = new
double[5];		Scanner sc= new Scanner(System.in);
	public void inputCIEmarks(){
			for(int i=0; i<5; i++){
				System.out.println("Enter the CIE marks of"+ (i+1)+"th subject");
				this.ciemarks[i]= sc.nextDouble();
			}
		}
}

package SEE; import
CIE.*;
import java.util.Scanner;

public class Externals extends Internals{
				protected double seemarks[] = new double[5];
			protected double finalMarks[] = new double[5] ;

				public void inputSEEmarks() {
					Scanner sc = new Scanner(System.in);
					for(int i=0; i<5; i++){
						System.out.println("Enter the SEE marks of"+ (i+1)+"th
subject");
```

```
                                        this.seemarks[i]= sc.nextDouble();
                                }
                        }

                        public void definefinalmarks(){
                                for(int i=0; i<5; i++){
                                        this.finalMarks[i] = ciemarks[i] + (seemarks[i]/2.0);
                                }


                        }


                        public void displayfinalmarks(){
                                System.out.println("The final marks of the student is \n");
                                for(int i=0; i<5; i++){
                                        System.out.println("The marks of the" + (i+1)+"th
subject is \t");
                                        System.out.println(this.finalMarks[i]);
                                }

                        }
}

import SEE.Externals;
import java.util.Scanner;

class Main{
                public static void main(String args[]){
                int n;
                        Scanner sc = new Scanner(System.in);
                        System.out.println("Name: Anirudh R");
                        System.out.println("USN: 1BM23CS036");
                        System.out.println("Enter the number of students \t");
                        n= sc.nextInt();
                        Externals e[] = new Externals[n];
        for(int i=0; i<n; i++){
                        e[i] = new Externals();
                        System.out.println("Enter the" +(i+1)+"th student details \n");
                e[i].inputStudentDetails();
                        System.out.println("Enter the" +(i+1)+"th student's CIE marks\n");
                        e[i].inputCIEmarks();
                        System.out.println("Enter the" +(i+1)+"th student's SEE marks\n");
                        e[i].inputSEEmarks();
                        System.out.println("The details of the" +(i+1)+"th student is");
```

```
                    e[i].displayStudentDetails();
                    e[i].definefinalmarks();
                    System.out.println("The final marks of the" +(i+1)+"th student is");
                    e[i].displayfinalmarks();
                }

            }
}
```

```
Enter student details:
Enter USN: 1BM23CS013
Enter Name: Abhishek patil
Enter Semester: 3
Enter internal marks for 5 courses:
Course 1: 89
Course 2: 70
Course 3: 67
Course 4: 98
Course 5: 86
USN: 1BM23CS013
Name: Abhishek patil
Semester: 3
Internal Marks:
Course 1: 89
Course 2: 70
Course 3: 67
Course 4: 98
Course 5: 86
Enter external marks for 5 courses:
Course 1: 88
Course 2: 89
Course 3: 90
Course 4: 87
Course 5: 80
External Marks:
Course 1: 88
Course 2: 89
Course 3: 90
Course 4: 87
Course 5: 80
Total Internal Marks: 410
Total External Marks: 434
Final Marks (Internal + External): 844
```

<u>Program 7</u>

Exception handling

Algorithm

Date 20/11/24

Lab-7 (week-7)

8) Write a program that demonstrate handling of exceptions in inheritance tree. Create a base class called "Father" and a derived class called "Son" which extends base class. In Father's class implement a constructor which takes age and throws ~~expetition~~ exception wrong age when the input is less than zero. In son class implement a constructor that uses father and son age and throws an exception if son's age is greater than equal to father age

```
import java.util.scanner;
class WrongAgeException extends Exception{
        public WrongAgeException (String message){
                super( message);
        }
}

class SonAgeException extends Exception{
        public SonAge Exception (String message){
                super(message);
        }
}

class Father {
        private int age;
        public Father(int age) throws WrongAge-
                                Exception {
                if (age<0)
                        throw new WrongAgeException(
                                message: "Wrong age");
                }
                this.age = age;
        }
        public int getAge(){
                return age;
        }
}

class son extends Father{
        private int sonAge;
        public Son (int fatherAge, int SonAge)
                throws WrongAgeException, SonAge,
                        ExceptionB{
                super( father-Age);

                if ( sonAge >= father Age) {
                        throw new SonAgeException(
                        message: "Son's age cannot be greater
                        than or equal to father's age");
                }
                this. sonAge = sonAge;
        }
        public int getSonAge(){
                return sonAge;
        }
}
```

```
public class FatherSon {
    Run | Debug
    public static void main (String[] args){
        while (true){

        Scanner sc = new Scanner(System.in);
        System.out.print(s: "Enter Father's Age:");
        int fatherAge = sc.nextInt();
        System.out.print (s: "Enter son's Age:");
        int sonAge = sc.nextInt();
        try {
            Son son = new son(fatherAge, SonAge);
            System.out.println(x: "Accepted
                                Succesfully");
        }
        catch (WrongAgeException e){
            System.out.println(e.getMessage());
        }
        catch (SonAgeException e){
            System.out.println(e.getMessage());
        }
        System.out.println(x: "Would you like to
                re-enter details (y/n)");
        String input = sc.next();
        if (input.equalsIgnoreCase("n")) {
            break;
        }
    }
}
}
```

→ import

class

Output: Enter Father's Age: 25
Enter Son's Age: 35
Son's age cannot be greater than or equal to father's age

Code:

```
import java.util.Scanner;

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

class SonAgeException extends Exception {
    public SonAgeException(String message) {
        super(message);
    }
}

class Father {
    private int age;

    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Wrong age");
        }
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException, SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException("Son's age cannot be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
```

```
    }
}

public class FatherSon {
    public static void main(String[] args) {
        while (true) {
            Scanner sc = new Scanner(System.in);
            System.out.println("Name : Abhishek patil\nUSN : 1BM23CS013");
            System.out.print("Enter Father's Age: ");
            int fatherAge = sc.nextInt();
            System.out.print("Enter Son's Age: ");
            int sonAge = sc.nextInt();
            try {
                Son son = new Son(fatherAge, sonAge);
                System.out.println("Accepted Succesfully");
            } catch (WrongAgeException e) {
                System.out.println(e.getMessage());
            } catch (SonAgeException e) {
                System.out.println(e.getMessage());
            }
            System.out.println("Would you like to re-enter details (Y/n)");
            String input = sc.next();
            if (input.equalsIgnoreCase("n")) {
                break;
            }
        }
    }
}
```

Output

```
D:\CS013>javac FatherSon.java

D:\CS013>java FatherSon
Name : Abhishek patil
USN : 1BM23CS013
Enter Father's Age: 34
Enter Son's Age: 56
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/n)
n
```

Program 8

Multithreading

Algorithm

Date 27/11/24

lab program-8

Q) Write a program which creats two threads, one thread displaying "BMS college of Engineering". Once Every ~~time~~ seconds and another displaying as "CSE"

```
class FirstThread extend Thread{
        public void run(){
            while (true){
                System.out.println("BMS college of
                                Engineering");
                try{
                    Thread.sleep(millis:10000);
                }catch(InterruptException e){
                    System.out.println(e);
                }
            }
        }
}

class SecondThread extends Thread{

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

```
class thread {
    public static void main(String [] args)
    {
        FirstThread thread1 = new FirstThread();
        SecondThread thread2 = new SecondThread();

        thread1.start();
        thread2.start();
    }
}
```

<u>Output:</u>

```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```

Code:

```
class FirstThread extends
Thread {
   public void run() {
      while (true) {

System.out.println("BMS
College of Engineering");
         try {

Thread.sleep(10000);  //
Sleep for 10 seconds
         } catch
(InterruptedException e) {

System.out.println(e);
         }
      }
   }
}

class SecondThread extends
Thread {
   public void run() {
      while (true) {

System.out.println("CSE");
         try {
            Thread.sleep(2000);
// Sleep for 2 seconds
         } catch
(InterruptedException e) {

System.out.println(e);
         }
      }
   }
}

class thread{
   public static void
main(String[] args) {
      // Create the threads
      FirstThread thread1 =
new FirstThread();
```

```
        SecondThread thread2 =
new SecondThread();

        // Start the threads
        thread1.start();
        thread2.start();
    }
}
```

Output

```
D:\abhishek\java programs>java thread
NAME: Abhishek patil
USN: 1BM23CS013
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
```

Program 9

Integer division with user interface

Algorithm

Date 27/11/24

Lab program - 9

SwingDemo

10) Write a program that creat a user interface to perform integer division The user enter two numbers in the next fields, Num1 and Num2. The division of The division Num1 and Num2 is displayed in the Result Field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

O/P: Divider App

Enter the divider and divident

[4] [3]

[calculate] A=4 B=3 Ans=1

Code: import
javax.swing.*; import
java.awt.*; import
java.awt.event.*; class
SwingDemo{
SwingDemo(){
 // create jframe container
 JFrame jfrm = new JFrame("Divider App");
jfrm.setSize(275, 150); jfrm.setLayout(new
FlowLayout());
 // to terminate on close
 jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 // text label
 JLabel jlab = new JLabel("Enter the divider and divident:");
 // add text field for both numbers
 JTextField ajtf = new JTextField(8);
 JTextField bjtf = new JTextField(8);
 // calc button
 JButton button = new JButton("Calculate");
 // labels
 JLabel err = new JLabel();
 JLabel alab = new JLabel();
 JLabel blab = new JLabel();
 JLabel anslab = new JLabel();
 // add in order :)
 jfrm.add(err); // to display error bois
jfrm.add(jlab); jfrm.add(ajtf);
jfrm.add(bjtf); jfrm.add(button);
jfrm.add(alab); jfrm.add(blab);
 jfrm.add(anslab);

 ActionListener l = new ActionListener() { public
void actionPerformed(ActionEvent evt) {
System.out.println("Action event from a text field");
 }
};
 ajtf.addActionListener(l);
bjtf.addActionListener(l);
 button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent evt) {
try{ int a = Integer.parseInt(ajtf.getText()); int b
= Integer.parseInt(bjtf.getText()); int ans = a/b;

```
 alab.setText("\nA = " + a);
blab.setText("\nB = " + b);
 anslab.setText("\nAns = "+ ans);
 }
 catch(NumberFormatException e){
 alab.setText("");
blab.setText("");
anslab.setText("");
 err.setText("Enter Only Integers!");
 }
 catch(ArithmeticException e){
 alab.setText("");
blab.setText("");
anslab.setText("");
 err.setText("B should be NON zero!");
 }
 }
 });
 // display frame
jfrm.setVisible(true);
 }
 public static void main(String args[]){  //
create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable(){
 public void run(){
 new SwingDemo();
 }
 });
 }
}
```

Output

Program 10

Inter process communication and deadlock

Algorithm

Date 27/11/24

program - 1b

Q) Program to demonstrate deadlock

O/P:

Main thread entered A.Foo
Racin Thread entered B.bar
Main thread trying to call B.last()
Racing thread trying to call A.last()
Inside A.last
Inside B.last
Back in main thread
Back in other thread

Q) Program to demonstrate inter process Communication

O/P: Press control_C to stop

Put : 0	Put : 11
Put : 1	Put : 12
Put : 2	Put : 13
Put : 3	Put : 14
Put : 4	Put : 14
Put : 5	Got : 14
Put : 6	Got : 14
Put : 7	⋮
Put : 8	
Put : 9	
Put : 10	

Seen 27/10/24

Code

```
class A {

synchroniz
ed void
foo(B b) {

String
name =
Thread.cur
rentThread
().getName
();

System.out
.println(na
me + "
entered
A.foo");

try {

Thread.slee
p(1000);

}
catch(Exce
ption e) {

System.out
.println("A
Interrupted
");

}
System.out
.println(na
me + "
trying to
call
B.last()");

b.last();

}
```

```
void last()
{

System.out
.println("In
side
A.last");

}

}

class B {

synchroniz
ed void
bar(A a) {

String
name =
Thread.cur
rentThread
().getName
();

System.out
.println(na
me + "
entered
B.bar");

try {

Thread.slee
p(1000);

}
catch(Exce
ption e) {

System.out
.println("B
Interrupted
");
```

```
}
System.out
.println(na
me + "
trying to
call
A.last()");

a.last();

}

void last()
{

System.out
.println("In
side
A.last");

}

}

class
Deadlock
implement
s Runnable
{

A a = new
A();

B b = new
B();

Deadlock()
{
Thread.cur
rentThread
().setName
("MainThr
ead");
```

```
Thread t =
new
Thread(thi
s,"RacingT
hread");

t.start();

a.foo(b); //
get lock on
a in this
thread.

System.out
.println("B
ack in
main
thread");

}

public void
run() {

b.bar(a); //
get lock on
b in other
thread.

System.out
.println("B
ack in
other
thread");

}

public
static void
main(Strin
g args[]) {

System.out
.println("\n
\nName:A
```

```
bhishek
patil");
System.out
.println("U
SN:1BM2
3CS013");

new
Deadlock()
;

}

}
```

IPC

```
class Q {

int n;

boolean
valueSet =
false;

synchroniz
ed int get()
{

while(!valu
eSet)

try {

System.out
.println("\n
Consumer
waiting\n")
;

wait();

}
catch(Inter
ruptedExce
ption e) {
```

```
System.out
.println("In
terruptedE
xception
caught");

}
System.out
.println("G
ot: " + n);

valueSet =
false;

System.out
.println("\n
Intimate
Producer\n
");

notify();

return n;

}

synchroniz
ed void
put(int n) {

while(valu
eSet)

try {

System.out
.println("\n
Producer
waiting\n")
;

wait();
```

```
}
catch(Inter
ruptedExce
ption e) {

System.out
.println("In
terruptedE
xception
caught");

}

this.n = n;

valueSet =
true;

System.out
.println("P
ut: " + n);

System.out
.println("\n
Intimate
Consumer\
n");

notify();

}

}

class
Producer
implement
s Runnable
{

Q q;

Producer(
Q q) {
```

```
this.q = q;

new
Thread(thi
s,
"Producer"
).start();

}

public void
run() {

int i = 0;

while(i<15
) {

q.put(i++);

}

}

}
class
Consumer
implement
s Runnable
{

Q q;

Consumer(
Q q) {

this.q = q;

new
Thread(thi
s,
"Consumer
").start();

}
```

```
public void
run() {

int i=0;

while(i<15
) {

int
r=q.get();

System.out
.println("c
onsumed:"
+r);

i++;

}

}

}

class
PCFixed{
public
static void
main(Strin
g args[]) {
System.out
.println("\n
Name:Abh
ishek
patil");
System.out
.println("U
SN:1BM2
3CS013\n"
);
Q q = new
Q();
```

```
new
Producer(q
);

new
Consumer(
q);

System.out
.println("Pr
ess
Control-C
to stop.");

}

}

}}
```

Output

```
D:\abhishek\java programs>java Deadlock


Name:Abhishek patil
USN:1BM23CS013
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
Inside A.last
Back in other thread
MainThread trying to call B.last()
Inside A.last
Back in main thread
```

```
D:\abhishek\java programs>java PCFixed

Name:Abhishek patil
USN:1BM23CS013

Press Control-C to stop.
Put: 0

Intimate Consumer


Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer


Producer waiting

consumed:0
Got: 1

Intimate Producer

consumed:1
Put: 2

Intimate Consumer


Producer waiting

Got: 2

Intimate Producer

consumed:2
Put: 3
```

```
Intimate Consumer


Producer waiting

Got: 3

Intimate Producer

consumed:3
Put: 4

Intimate Consumer


Producer waiting

Got: 4

Intimate Producer

consumed:4
Put: 5

Intimate Consumer


Producer waiting

Got: 5

Intimate Producer

consumed:5
Put: 6

Intimate Consumer


Producer waiting

Got: 6

Intimate Producer

consumed:6
Put: 7
```

```
Intimate Consumer


Producer waiting

Got: 7

Intimate Producer

consumed:7
Put: 8

Intimate Consumer


Producer waiting

Got: 8

Intimate Producer

consumed:8
Put: 9

Intimate Consumer


Producer waiting

Got: 9

Intimate Producer

consumed:9
Put: 10

Intimate Consumer


Producer waiting

Got: 10

Intimate Producer

consumed:10
Put: 11
```

```
Intimate Consumer


Producer waiting

Got: 11

Intimate Producer

consumed:11
Put: 12

Intimate Consumer


Producer waiting

Got: 12

Intimate Producer

consumed:12
Put: 13

Intimate Consumer


Producer waiting

Got: 13

Intimate Producer

consumed:13
Put: 14

Intimate Consumer

Got: 14

Intimate Producer

consumed:14
```