

B.M.S. COLLEGE OF ENGINEERING
Basavanagudi, Bengaluru- 560019
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



LAB REPORT

On

Object Oriented Java Programming
(23CS3PCOOJ)

Submitted By:

Abhinav Raghу
1BM22CS005

In partial fulfilment of
BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
2023-24

Faculty-In-Charge
Swathi Sridharan

Assistant Professor
Department of Computer Science and Engineering

Sl.No	Topic	Page No.
1	Quadratic Equation	1
2	Student SGPA Calculator	2
3	Book Problem	4
4	Shapes	7
5	Bank	8
6	Student's Internal and External Marks	14
7	Exception Handling	19
8	Threads	20
9	AWT	22

Java Lab Programs

Q1	Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.
Answer:	<pre>import java.util.*; public class prg_1 { public static void main(String[] args) { Scanner input = new Scanner(System.in); while (true) { System.out.print("Enter the coefficient of x^2: "); int a = input.nextInt(); System.out.print("Enter the coefficient of x: "); int b = input.nextInt(); System.out.print("Enter the constant value: "); int c = input.nextInt(); double discriminant = Math.pow(b, 2) - 4*a*c; if (discriminant > 0) { System.out.println("The Quadratic Equation has two real values for x."); double x1 = ((-b) + (Math.sqrt(discriminant))) / (2*a); double x2 = ((-b) - (Math.sqrt(discriminant))) / (2*a); System.out.println("Solution 1: " + x1); System.out.println("Solution 2: " + x2); } else if(discriminant == 0){ System.out.println("The Quadratic Equation has one real value for x."); double x = (-b) / (2*a); System.out.println("Solution : " + x); } else{ System.out.println("The Quadratic Equation has no real values for x."); } System.out.print("Do you want to continue? (y/n): "); String chk = input.next(); System.out.println(); if (!chk.equals("y")) { System.out.println(); System.out.println("-Abhinav Raghu \n 1B22CS005"); input.close(); break; } } } }</pre>

```

        else{
            continue;
        }
    }
}

```

Output:

```

Enter the coefficient of x^2: 1
Enter the coefficient of x: -3
Enter the constant value: 2
The Quadratic Equation has two real values for x.
Solution 1: 2.0
Solution 2: 1.0
Do you want to continue? (y/n): y

Enter the coefficient of x^2: 1
Enter the coefficient of x: -2
Enter the constant value: 1
The Quadratic Equation has one real value for x.
Solution : 1.0
Do you want to continue? (y/n): y

Enter the coefficient of x^2: 2
Enter the coefficient of x: 1
Enter the constant value: 3
The Quadratic Equation has no real values for x.
Do you want to continue? (y/n): n

-Abhinav Raghu
1B22CS005

```

Q2. Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

Answer:

```

import java.util.Scanner;

public class Student {
    private String usn;
    private String name;
    private int[] credits;
    private int[] marks;
    private double sgpa;

    public void acceptDetails(){
        Scanner input = new Scanner(System.in);

        System.out.print("Enter USN: ");
        usn = input.nextLine();

        System.out.print("Enter Name: ");
        name = input.nextLine();
        System.out.println();

        System.out.print("Enter Number of Subjects: ");
        int sub_count = input.nextInt();
        credits = new int[sub_count];
        marks = new int[sub_count];

        for (int i = 0; i < sub_count; i++) {
            System.out.print("Enter number of credits for Subject "+(i+1)+": ");

```

```

        credits[i] = input.nextInt();

        System.out.print("Enter marks received for Subject "+(i+1)+": ");
        marks[i] = input.nextInt();
        System.out.println();
    }

    input.close();
}

public void calculate(){
    int sum = 0;

    for (int i=0; i<credits.length; i++) {
        if ((marks[i]>=90) && (marks[i]<=100)) {
            sum = sum + 10 * credits[i];
        }
        else if ((marks[i]>=80) && (marks[i]<90)) {
            sum = sum + 9 * credits[i];
        }
        else if ((marks[i]>=70) && (marks[i]<80)) {
            sum = sum + 8 * credits[i];
        }
        else if ((marks[i]>=60) && (marks[i]<70)) {
            sum = sum + 7 * credits[i];
        }
        else if ((marks[i]>=55) && (marks[i]<60)) {
            sum = sum + 6 * credits[i];
        }
        else if ((marks[i]>=50) && (marks[i]<55)) {
            sum = sum + 5 * credits[i];
        }
        else if ((marks[i]>=40) && (marks[i]<50)) {
            sum = sum + 4 * credits[i];
        }
        else if ((marks[i]>=0) && (marks[i]<40)) {
            sum = sum + 0 * credits[i];
        }
    };
    int csum = 0;

    for (int c : credits) {
        csum += c;
    }

    sgpa = (double)sum / csum;
}

public void displayDetails(){
    System.out.println("Student details: ");
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
}

```

```

        for (int i = 0; i < credits.length; i++) {
            System.out.println("Credits for Subject "+(i+1)+": " + credits[i]);
            System.out.println("Marks received for Subject "+(i+1)+": " + marks[i]);
        }
        System.out.println("SGPA: "+ sgpa);
        System.out.println();
        System.out.println("-Abhinav Raghu \n 1B22CS005");
    }

    public static void main(String[] args) {
        Student s = new Student();
        s.acceptDetails();
        s.calculate();
        s.displayDetails();
    }
}

```

Output:

```

Enter USN: 1BM22CS005
Enter Name: Abhinav Raghu

Enter Number of Subjects: 5
Enter number of credits for Subject 1: 4
Enter marks received for Subject 1: 92

Enter number of credits for Subject 2: 3
Enter marks received for Subject 2: 89

Enter number of credits for Subject 3: 2
Enter marks received for Subject 3: 97

Enter number of credits for Subject 4: 1
Enter marks received for Subject 4: 98

Enter number of credits for Subject 5: 4
Enter marks received for Subject 5: 88

Student details:
USN: 1BM22CS005
Name: Abhinav Raghu
Credits for Subject 1: 4
Marks received for Subject 1: 92
Credits for Subject 2: 3
Marks received for Subject 2: 89
Credits for Subject 3: 2
Marks received for Subject 3: 97
Credits for Subject 4: 1
Marks received for Subject 4: 98
Credits for Subject 5: 4
Marks received for Subject 5: 88
SGPA: 9.5

-Abhinav Raghu
1B22CS005

```

Q3. Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

Answer:

```

import java.util.Scanner;

class Book{
    private String name;
    private String author;
    private double price;
    private int num_pages;
    private String details;
}

```

```

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

        public String getName() {
            return name;
        }
        public String getAuthor() {
            return author;
        }
        public double getPrice() {
            return price;
        }
        public int getNum_pages() {
            return num_pages;
        }
        public String getDetails() {
            return details;
        }

        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 setNum_pages(int num_pages) {
            this.num_pages = num_pages;
        }
        public void setDetails(String details) {
            this.details = details;
        }

        @Override
        public String toString() {
            return "Book [name=" + name + ", author=" + author + ", price=" + price + ",
num_pages=" + num_pages ;
        }

    }
    public class Prg_3 {

```

```

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

    System.out.print("Enter the number of Books: ");
    int n = input.nextInt();
    input.nextLine(); //-->Major issue if not written

    Book b[] = new Book[n];

    for (int i = 0; i < n; i++) {
        System.out.println();
        System.out.println("Enter details for Book " + (i + 1) + ":");
        System.out.print("Name: ");
        String name = input.nextLine();
        System.out.print("Author: ");
        String author = input.nextLine();
        System.out.print("Price: ");
        double price = input.nextDouble();
        System.out.print("Number of Pages: ");
        int numPages = input.nextInt();

        b[i] = new Book(name, author, price, numPages);
        input.nextLine();
        System.out.print("Details: ");
        String details = input.nextLine();
        b[i].setDetails(details);

    }
    System.out.println();
    for(int j=0; j<n; j++){
        System.out.print(b[j].toString());
        System.out.println(" ,details: "+ b[j].getDetails() + "]");
        System.out.println();
    }
    System.out.println();
    System.out.println("-Abhinav Raghu \n 1B22CS005");

    input.close();
}
}

```

Output	<pre> Enter the number of Books: 2 Enter details for Book 1: Name: The Catcher in the Rye Author: J.D. Salinger Price: 14.99 Number of Pages: 224 Details: A coming-of-age novel capturing teenage angst. Enter details for Book 2: Name: The Hobbit Author: J.R.R. Tolkien Price: 19.99 Number of Pages: 310 Details: Book [name=The Catcher in the Rye, author=J.D. Salinger, price=14.99, num_pages=224 ,details: A coming-of-age novel capturing teenage angst.] Book [name=The Hobbit, author=J.R.R. Tolkien, price=19.99, num_pages=310 ,details:] -Abhinav Raghu 1B22CS005 </pre>
Q4	<p>Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.</p>
Answer:	<pre> import java.util.*; abstract class Shape{ int p1; int p2; Shape(int p1, int p2){ this.p1 = p1; this.p2 = p2; } public void printArea(){ } } class Rectangle extends Shape{ Rectangle(int length, int breadth){ super(length, breadth); } @Override public void printArea() { System.out.println("Area of Rectangle: " + (p1*p2)); } } class Triangle extends Shape{ Triangle(int height, int base){ super(height,base); } @Override </pre>

```

        public void printArea() {
            System.out.println("Area of Triangle: "+ ( (0.5)*p1*p2) );
        }
    }

    class Circle extends Shape{
        Circle(int r){
            super(r, 0);
        }
        @Override
        public void printArea() {
            System.out.println("Area of circle: "+ (Math.PI * p1 * p1) );
        }
    }
    public class Prg_4 {
        public static void main(String[] args) {
            Shape r = new Rectangle(15, 12);
            Shape t = new Triangle(20, 4);
            Shape c = new Circle(5);
            r.printArea();
            t.printArea();
            c.printArea();
            System.out.println();

            System.out.println("-Abhinav Raghu \n 1B22CS005");
        }
    }
}

```

Output:

```

Area of Rectangle: 180
Area of Triangle: 40.0
Area of circle: 78.53981633974483

-Abhinav Raghu
1B22CS005

```

Q5

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, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- Accept deposit from customer and update the balance.
- Display the balance.
- Compute and deposit interest
- Permit withdrawal and update the balance

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

Answer:

```
import java.util.Scanner;

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

    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. Updated balance: " + balance);
    }

    void displayBalance() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Type: " + accountType);
        System.out.println("Current Balance: " + balance);
    }

    void depositInterest(double rate) {
        if (accountType.equalsIgnoreCase("Savings")) {
            double interest = balance * rate / 100;
            balance += interest;
            System.out.println("Interest deposited. Updated balance: " + balance);
        } else {
            System.out.println("Current account does not earn interest.");
        }
    }

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }
}

class CurAcct extends Account {
    double minBalance;
    double serviceCharge;
```

```

        CurAcct(String customerName, int accountNumber, double balance, double minBalance, double
serviceCharge) {
    super(customerName, accountNumber, "Current", balance);
    this.minBalance = minBalance;
    this.serviceCharge = serviceCharge;
}

@Override
void withdraw(double amount) {
    if (balance - amount >= minBalance) {
        super.withdraw(amount);
    } else {
        double excessAmount = Math.abs(amount-minBalance);
        System.out.println("Withdrawal successful, but below minimum balance.");
        System.out.println("Excess amount: " + excessAmount);
        System.out.println("Balance after excess withdrawal: " + balance);

        balance = balance - amount - serviceCharge;
        System.out.println("Service charge imposed. Updated balance: " + balance);
    }
}
}

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

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

        System.out.print("Enter the number of customers: ");
        int numCustomers = scanner.nextInt();

        for (int i = 1; i <= numCustomers; i++) {
            System.out.println("\nEnter details for Customer " + i + ":");
            System.out.print("Customer Name: ");
            String customerName = scanner.next();
            System.out.print("Account Type (Savings/Current): ");
            String accountType = scanner.next();
            System.out.print("Account Number: ");
            int accountNumber = scanner.nextInt();
            System.out.print("Initial Balance: ");
            double initialBalance = scanner.nextDouble();

            Account account;
            if (accountType.equalsIgnoreCase("Savings")) {
                account = new SavAcct(customerName, accountNumber, initialBalance);
            }
        }
    }
}

```

```

        } else if (accountType.equalsIgnoreCase("Current")) {
            System.out.print("Minimum Balance: ");
            double minBalance = scanner.nextDouble();
            System.out.print("Service Charge: ");
            double serviceCharge = scanner.nextDouble();
            account = new CurAcct(customerName, accountNumber, initialBalance,
minBalance, serviceCharge);
        } else {
            System.out.println("Invalid account type. Please enter 'Savings' or
'Current'.");
            i--;
            continue;
        }

        boolean continueOperations = true;
        while (continueOperations) {
            System.out.println("\nOperations on " + account.accountType + " Account:");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            if (accountType.equalsIgnoreCase("Savings")) {
                System.out.println("4. Deposit Interest");
            }
            System.out.println((i == numCustomers) ?
((accountType.equalsIgnoreCase("Savings")) ? "5. Exit" : "4. Exit"):
((accountType.equalsIgnoreCase("Savings")) ? "5. Move to next customer" : "4. Move to next
customer"));

            if (accountType.equalsIgnoreCase("Savings")) {
                System.out.print("Enter your choice (1-5): ");
            }
            else{

                System.out.print("Enter your choice (1-4): ");
            }

            int choice = scanner.nextInt();

            if (accountType.equalsIgnoreCase("Current")) {
                if (choice == 4) {
                    choice++;
                }
            }

            switch (choice) {
                case 1:
                    System.out.print("Enter deposit amount: ");
                    double depositAmount = scanner.nextDouble();
                    account.deposit(depositAmount);
                    break;
                case 2:
                    System.out.print("Enter withdrawal amount: ");
                    double withdrawAmount = scanner.nextDouble();

```

```

        account.withdraw(withdrawAmount);
        break;
    case 3:
        account.displayBalance();
        break;
    case 4:
        if (accountType.equalsIgnoreCase("Savings")) {
            System.out.print("Enter interest rate for savings account: ");
            double interestRate = scanner.nextDouble();
            ((SavAcct) account).depositInterest(interestRate);
        } else {
            System.out.println("Invalid choice for the current account
type.");
        }
        break;
    case 5:
        continueOperations = false;
        break;
    default:
        System.out.println("Invalid choice. Please enter a number between 1
and 5.");
    }
}
scanner.close();

}
}

```

Output:

```

Enter details for Customer 1:
Customer Name: A
Account Type (Savings/Current): Current
Account Number: 236754
Initial Balance: 90000
Minimum Balance: 20000
Service Charge: 1500

Operations on Current Account:
1. Deposit
2. Withdraw
3. Display Balance
4. Move to next customer
Enter your choice (1-4): 3
Account Number: 236754
Customer Name: A
Account Type: Current
Current Balance: 90000.0

Operations on Current Account:
1. Deposit
2. Withdraw
3. Display Balance
4. Move to next customer
Enter your choice (1-4): 2
Enter withdrawal amount: 10000
Withdrawal successful. Updated balance: 80000.0

```

```
Operations on Current Account:  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Move to next customer  
Enter your choice (1-4): 2  
Enter withdrawal amount: 70000  
Withdrawal successful, but below minimum balance.  
Excess amount: 50000.0  
Balance after excess withdrawal: 80000.0  
Service charge imposed. Updated balance: 85000.0  
  
Operations on Current Account:  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Move to next customer  
Enter your choice (1-4): 3  
Account Number: 236754  
Customer Name: A  
Account Type: Current  
Current Balance: 85000.0  
  
Operations on Current Account:  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Move to next customer  
Enter your choice (1-4): 1  
Enter deposit amount: 10000  
Deposit successful. Updated balance: 185000.0
```

```
Operations on Current Account:  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Move to next customer  
Enter your choice (1-4): 4  
  
Enter details for Customer 2:  
Customer Name: Lokesh  
Account Type (Savings/Current): Savings  
Account Number: 45763  
Initial Balance: 50000  
  
Operations on Savings Account:  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Deposit Interest  
5. Exit  
Enter your choice (1-5): 3  
Account Number: 45763  
Customer Name: Lokesh  
Account Type: Savings  
Current Balance: 50000.0  
  
Operations on Savings Account:  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Deposit Interest  
5. Exit  
Enter your choice (1-5): 1  
Enter deposit amount: 1400  
Deposit successful. Updated balance: 51400.0
```

```

Operations on Savings Account:
1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest
5. Exit
Enter your choice (1-5): 2
Enter withdrawal amount: 400
Withdrawal successful. Updated balance: 51000.0

Operations on Savings Account:
1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest
5. Exit
Enter your choice (1-5): 3
Account Number: 45763
Customer Name: Lokesh
Account Type: Savings
Current Balance: 51000.0

Operations on Savings Account:
1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest
5. Exit
Enter your choice (1-5): 5

Abhinav Raghu

```

Q6. Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class Internals 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.

Answer:

```

JavaPackages/CIE
  JavaPackages/CIE/Student
    package CIE;
    import java.util.*;

    public class Student
    {
        static String USN;
        static String Name;
        static String Sem;
        public static int n;

        public static String[][] student_info()
        {

            Scanner input = new Scanner(System.in);

            String[][][] arr_s = new String[n][3];

            for(int i=0; i < n; i++)
            {
                System.out.println("Student " + (i+1));
                System.out.println();
                System.out.print("Enter USN: ");

```

```

        USN = input.nextInt();
        input.nextLine();

        System.out.print("Enter Name: ");
        Name = input.nextLine();

        System.out.print("Enter Sem: ");
        Sem = input.nextLine()

        arr_s[i][0] = USN;
        arr_s[i][1] = Name;
        arr_s[i][2] = Sem;

        System.out.println();
    }
    return arr_s;
}
}

```

Java Packages/CIE/Internal

```

package CIE;
import java.util.Scanner;

public class Internal {
    public static int size;
    public static int[] internal_values() {

        Scanner input = new Scanner(System.in);

        int[] arr = new int[size];
        int check;

        for (int i = 0; i < arr.length; i++)
        {
            System.out.print("Course " + (i+1) + ": ");
            check = input.nextInt();

            if ((check>=0) && (check<=50))
            {
                arr[i] = check;
            }
            else
            {
                System.out.println("Wrong value, enter again!\n\n");
                i--;
            }
        }

        return arr;
    }
}

```

```
}
```

Java Packages / SEE

Java Packages / SEE / External

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

public class External {
    static int size = Internal.size ;
    public static int[] external_values() {

        Scanner input = new Scanner(System.in);

        int[] arr = new int[size];
        int check;

        for (int i = 0; i < arr.length; i++)
        {
            System.out.print("Course " + (i+1) + ": ");
            check = input.nextInt();

            if ((check>=0) && (check<=50))
            {
                arr[i] = check;
            }
            else
            {
                System.out.println("Wrong value, enter again!\n");
                System.out.println();
                i--;
            }
        }

        return arr;
    }
}
```

Java Packages

Java Packages / Calcuate

```
import CIE.*;
import SEE.*;
import java.util.*;
```

```

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

        Scanner input = new Scanner(System.in);
        System.out.print("Number of Students: ");
        Student.n = input.nextInt();

        String[][][] std_details = Student.student_info();

        int[][][] std_arr = new int[Student.n][];
        int[][][] std_ext = new int[Student.n][];

        System.out.print("Enter number of courses each student has: ");
        Internal.size = input.nextInt();

        System.out.println("\nEnter Internal marks out of 50.\n");
        for (int i = 0; i < Student.n; i++)
        {
            System.out.println("Student " + (i+1));
            std_arr[i] = Internal.internal_values();
            System.out.println();
        }

        System.out.println("\nEnter External marks out of 50.\n");
        for (int i = 0; i < Student.n; i++)
        {
            System.out.println("Student " + (i+1));
            std_ext[i] = External.external_values();
            System.out.println();
        }

        int total;

        for (int i = 0; i < std_arr.length; i++)
        {
            System.out.println("-----");
            System.out.println("Student " + (i+1));
            System.out.println("USN: " + std_details[i][0]);
            System.out.println("Name: " + std_details[i][1]);
            System.out.println("Sem: " + std_details[i][2]);
            System.out.println();

            System.out.println("Internal marks:");
            for (int j = 0; j < (std_arr[i].length); j++)
            {
                System.out.println("Course " + (j+1) + ": " + std_arr[i][j]);
            }
            System.out.println();

            System.out.println("External marks:");
        }
    }
}

```

```

        for (int j = 0; j < (std_ext[i].length); j++)
        {
            System.out.println("Course " + (j+1) + ": " + std_ext[i][j]);
        }
        System.out.println();

        System.out.println("Total marks:");
        for (int j = 0; j < (std_arr[i].length); j++)
        {
            System.out.println("Course " + (j+1) + ": " +
                (std_arr[i][j]+std_ext[i][j]));
        }
        System.out.println("-----");
        System.out.println();
    }
    input.close();
}
}

```

Output:

```

Number of Students: 2
Student 1

Enter USN: 1
Enter Name: A
Enter Sem: 3

Student 2

Enter USN: 2
Enter Name: B
Enter Sem: 2

Enter number of courses each student has: 5

Enter Internal marks out of 50.

Student 1
Course 1: 30
Course 2: 45
Course 3: 42
Course 4: 48
Course 5: 50

Student 2
Course 1: 25
Course 2: 30
Course 3: 44
Course 4: 36
Course 5: 38

```

```

External marks:
Course 1: 41
Course 2: 41
Course 3: 46
Course 4: 47
Course 5: 49

Total marks:
Course 1: 71
Course 2: 86
Course 3: 88
Course 4: 95
Course 5: 99
-----
-----

Student 2
USN: 2
Name: B
Sem: 2

Internal marks:
Course 1: 25
Course 2: 30
Course 3: 44
Course 4: 36
Course 5: 38

External marks:
Course 1: 38
Course 2: 40
Course 3: 43
Course 4: 40
Course 5: 46

Total marks:
Course 1: 63
Course 2: 70
Course 3: 87
Course 4: 76
Course 5: 84
-----
-----
```

Abhinav Raghu

Q7. Write a program that demonstrates handling of exceptions in inheritance tree. Create a base 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 cases both father and son’s age and throws an exception if son’s age is >=father’s age.

Answer:

```

class WrongAge extends Exception{
    public WrongAge(String Str){
        super(Str);
    }
}

class Father{
    Father(int age) throws WrongAge
    {
        if(age<0){
            throw new WrongAge("Invalid Age");
        }
    }
}
```

```

        }
    }

    class Son extends Father{
        Son(int sonAge, int fatherAge) throws WrongAge{
            super(fatherAge);
            if(sonAge>=fatherAge){
                throw new WrongAge("Son's age should be less than father's age");
            }
        }
    }

    public class prg_7 {
        public static void main(String[] args) {

            try {
                Father f1 = new Father(-1);
                Son s1 = new Son(10, 45);
            } catch (WrongAge e) {
                System.out.println("Error: " + e.getMessage());
            }
        }
    }
}

```

Output:

```
Father f1 = new Father(-1);
Son s1 = new Son(0, 15);
```

Error: Invalid Age

Abhinav Raghu

```
Father f1 = new Father(50);
Son s1 = new Son(36, 15);
```

Error: Son's age should be less than father's age

Abhinav Raghu

Q8.

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

Answer:

```
class B extends Thread {
    @Override
    public void run() {
        for (int i = 0; i < 5; i++) {
            System.out.println("BMS");
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
        }
    }

    class C extends Thread {
        @Override
        public void run() {
            for (int i = 0; i < 20; i++) {
                System.out.println("CSE");
                try {
                    Thread.sleep(2000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    public class prg_8{
        public static void main(String[] args) {
            B b1 = new B();
            C c1 = new C();
            b1.start();
            c1.start();
            System.out.println();

            System.out.println("Abhinav Raghav");
        }
    }
}
```

Output:	<pre>Abhinav Raghuram CSE BMS CSE CSE CSE CSE CSE BMS CSE CSE CSE CSE CSE BMS CSE CSE CSE CSE CSE CSE BMS</pre>
Q9.	<p>Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of 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.</p>
Answer:	<pre>import javax.swing.*; import java.awt.*; import java.awt.event.*; class SwingDemo{ SwingDemo(){ // create jframe container JFrame jfrm = new JFrame("Divide 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 divisor and dividend:"); // add text field for both numbers JTextField ajtf = new JTextField(8); JTextField bjtf = new JTextField(8); // calc button JButton button = new JButton("Calculate"); } }</pre>

```

// 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(ArithmaticException e){
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("B should be NON zero!");
        }
    }
});
jfrm.setVisible(true);
}

```

```
public static void main(String args[]){
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
```

