B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

Submitted in partial fulfillment of the requirements for

Lab Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

RAGHAVENDRA N

(1BM22CS213)

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019

2023-2024.

INDEX

Sl.No.	Title	Date
1	Lab 1	12/12/2023
2	Lab 2	19/12/2023
3	Lab 3	26/12/2023
4	Lab 4	02/01/2024
5	Lab 5	09/01/2024
6	Lab 6	16/01/2024
7	Lab 7	23/01/2024
8	Lab 8	30/01/2024
9	Lab 9	06/02/2024
10	Lab 10	20/02/2024

WEEK 1

1. Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.

Program:

```
import java.util.Scanner;

class Quadratic {
   int a, b, c;
   double r1, r2, d;

   void getCoefficients() {
      Scanner scanner = new Scanner(System.in);
      System.out.println("Enter the coefficients of a, b, and c:");
      a = scanner.nextInt();
      b = scanner.nextInt();
      c = scanner.nextInt();
   }

   void computeRoots() {
      while (a == 0) {
         System.out.println("Not a quadratic equation");
         System.out.println("Enter a non-zero value for a:");
         Scanner scanner = new Scanner(System.in);
         a = scanner.nextInt();
      }

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

      if (d == 0) {
         r1 = -b / (2.0 * a);
         System.out.println("Roots are real and equal");
         System.out.println("Root1 = Root2 = " + r1);
      } else if (d > 0) {
```

```
            r1 = (-b + Math.sqrt(d)) / (2.0 * a);
            r2 = (-b - Math.sqrt(d)) / (2.0 * a);
            System.out.println("Roots are real and distinct");
            System.out.println("Root1 = " + r1 + " Root2 = " + r2);
        } else {
            System.out.println("Roots are imaginary");
            r1 = -b / (2.0 * a);
            r2 = Math.sqrt(-d) / (2.0 * a);
            System.out.println("Root1 = " + r1 + " + i" + r2);
            System.out.println("Root2 = " + r1 + " - i" + r2);
        }
    }
}

class QuadraticMain {
    public static void main(String[] args) {
        System.out.println("My Name is Raghavendra N");
        System.out.println("My USN is 1BM22CS213");
        Quadratic quadratic = new Quadratic();
        quadratic.getCoefficients();
        quadratic.computeRoots();
    }
}
```

LAB - I

[1] // Printing Hello world.

```
class HelloWorld {
    Public static void main(String[] args){
        System.out.println(" My name is Raghavendra N ");
        System.out.println(" USN is 1BM22CS213 ");
        System.out.println(" Hello World ");
    }
}
```

Out Put :-

My name is Raghavendra N
USN is 1BM22CS213
Hello world

[2] // area of rectangle.

```
class RectangleArea {
    public static void main(String args[]){
        int length, breadth;
        length = Integer.parseInt(args[0]);
        breadth = Integer.parseInt(args[1]);

        int area = length * breadth;
        System.out.println("length of rectangle
                = " + length);
        System.out.println(" breadth of
                rectangle = " + breadth);
        System.out.println(" area of rectangle
                = " + area);
    }
```

Output :

length of rectangle = 5
breadth of rectangle = 3
area of rectangle = 15

[3] // Quadratic

```
import java.util.scanner;
class Quadratic
{
    int a,b,c;
    double r1, r2, d;
    void getd()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficients
                            of a,b,c");
        a=s.nextInt();
        b=s.nextInt();
        c=s.nextInt();
    }
    void compute()
    {
        while(a==0)
        {
            System.out.println("Not a Quadratic
                                equation");
            system.out.println("Enter a non zero
                                value for a:");
            Scanner s=new Scanner(System.in);
            a=s.nextInt();
        }
```

```
d = b*b-4*a*c;
if (d==0)
{
    r1 = (-b)/(2*a);
    System.out.println("Roots are real
                and equal");
    system.out.println("Root1= Root2="
                +r1);
}
else if(d>0)
{
    r1 = ((-b)+(math.sqrt(d)))/(double)
                (2*a);
    r2 = ((-b)+(math.sqrt(d)))/(double)
                (2*a);
    system.out.println("Roots are real
            and distinct");
    System.out.println("Root1=" +r1+"
            Root2=" +r2);
}
else if(d<0)
{
    system.out.println("Roots are
        imaginary");
    r1 = (-b)/(2*a);
    r2 = math.sqrt(-d)/(2*a);
    system.out.println("Root=" +r1+
            "+i" +r2);
    system.out.println("Root1=" +r1
            +"-i" +r2);
}
}
}
```

```
class Quadratic main
{
    Public static void main(String
                        args[])
    {
        Quadratic q = new quadratic();
        q.getd();

        q.computed();
    }
}
```

Output :-

```
My name is Raghavendra.N
USN is 1BM22CS213
enter the coefficient of a, b, c
1
5
3
Root 1 = 0.6972243622680054 and,
Root 2 = 4.30775633100 5
```

[4] // example of scanner

```
import java.util.scanner;
class HelloWorld{
    Public static void main(String args[])
    {
        int a; float b; String s;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a string");
        s = in.nextInt();
        System.out.println("You entered string
                            +s);
```

```
        System.out.println("Enter an integer");
        a=in.nextInt();
        system.out.println("you entered integer"
                                        + a);
        system.out.println("Enter a float");
        b= in.nextFloat();
        System.out.println("you entered
                                float " + b);
    }
}
```

[5]

```
// factorial of a number

class factorial{
    public static void main(String args[])
    {
        int fac=1;
        system.out.println("Enter a number:");
        Scanner sc= new Scanner(System.in);
        int n=sc.nextInt();
        for(int i=1; i<=n; i++){
            fac=fac*i;
        }
        system.out.println("the factorial:
                                " + fac);
    }
}
```

[6] // Example to know the usage of array.

```
class AutoArray {
    Public static void main(String args[]) {
        int month_days[] = {31,28,31,30,31,30,31,31,
                            30,31,30,31};
        System.out.println("April has " + month_days[
                                + " days."); }
}
```

Output :-

April has 31 days

[7] // Palindrome or not

```
class palindrome {
    Public static void main(String args[])
    {
        int n, t, rem, rev=0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a 5 digit number:");
        n = sc.nextInt();
        t = n;
        while(t>0) {
            rem = t%10;
            rev = rev * 10 + rem;
            t = t/10;
        }
        if(rev == n) {
            System.out.println("Palindrome");
        }
        else {
            System.out.println("not palindrome");
        }
    }
}
```

[8] // prime or not

```
import java.util.*;
class isprime
{
    static void isprime(int n)
    {
        int i, m=0, flag=0;
        m=n/2;
        if(n==0||n==1)
        {
            system.out.println(n+" is not a prime
                                number.");
        }
        else {
            for(i=2; i<=m; i++)
            {
                if(n%i==0)
                {
                    system.out.println(n+" is
                    not a prime number");
                    flag=1;
                    break;
                }
            }
            if(flag==0)
            {
                system.out.println(n+" is a
                prime number.");
            }
        }
    }
    public static void main(String args[])
    {
        int i;
        Scanner sc=new Scanner(System.in);
```

```
        System.out.println("Enter the value of i :");
        i= sc.nextInt();
        isPrime(i);
      }
    }
```

[9] // sum of digit

```
class sumofdigits{
    public static void main(String args[]){
        long number, sum;
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter a 5-digit number :");
        number = sc.nextLong();
        for(sum=0; number!=0; number=number/10)
        {
            sum=sum+number%10
        }
        System.out.println("sum of digits "+sum);
    }
}
```

[10] // largest of 3 number.

```
class largest method{
    static void largest(int i, int j, int k)
    {
        if(i>=j && i>=k){
            System.out.println(i+" is the largest number");
        }
```

```
if(j>=i && j>=k){
    system.out.println(j+"is the largest
                        number");
}
if(k>=i && k>=j){
    system.out.println(k+"is the
                largest number");
}
}
Public static void main(String args[])
{
    int i,j,k;
    i=Integer.parseInt(args[0]);
    j=Integer.parseInt(args[1]);
    k=Integer.parseInt(args[2]);
    largest(i,j,k);
}
}
```

WEEK2

2.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.

Program:

```
import java.util.Scanner;
import java.io.*;

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

    public Student(String usn, String name, int numSubjects) {
        this.usn = usn;
        this.name = name;
        this.credits = new int[numSubjects];
        this.marks = new int[numSubjects];
    }

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

        System.out.println("Enter details for student " + name + " (USN: " + usn +
")");
        for (int i = 0; i < credits.length; i++) {
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            credits[i] = scanner.nextInt();

            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
    }

    public void displayDetails() {
        System.out.println("Details for student " + name + " (USN: " + usn + ")");
```

```
        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + " - Credits: " + credits[i] + ",
Marks: " + marks[i]);
        }
    }

    public double calculateSGPA() {
        double totalCredits = 0;
        double totalGradePoints = 0;

        for (int i = 0; i < credits.length; i++) {
            totalCredits += credits[i];
            totalGradePoints += calculateGradePoints(marks[i]) * credits[i];
        }

        return totalGradePoints / totalCredits;
    }

       private double calculateGradePoints(int marks) {
        if (marks >= 90) {
            return 10.0;
        } else if (marks >= 80) {
            return 9.0;
        } else if (marks >= 70) {
            return 8.0;
        } else if (marks >= 60) {
            return 7.0;
        } else if (marks >= 50) {
            return 6.0;
        } else {
            return 0.0;
        }
    }
}

public class StudentSGPA {
    public static void main(String[] args) {
        System.out.print("my name is Raghavendra N");
        System.out.print("my USN is 1BM22CS213");
```

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

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

        System.out.print("Enter the USN: ");
        String usn = scanner.next();

        System.out.print("Enter the name: ");
        String name = scanner.next();

        Student student = new Student(usn, name, numSubjects);
        student.acceptDetails();

        student.displayDetails();
        System.out.println("SGPA: " + student.calculateSGPA());
    }
}
```

LAB - II

Date: 19/12/2023

[1] // SGPA & CGPA of Student.

```
import java.util.Scanner;

class Subject {
    int subjectMarks;
    int credits;
    int grade;
}
class Student {
    String name;
    String usn;
    double SGPA;
    Subject[] subjects;
    Scanner scanner;

    // constructor
    Student() {
        subjects = new Subject[8];
        for(int i=0; i<8; i++) {
            subjects[i] = new Subject();
        }
        scanner = new Scanner(System.in
    }

    // Method to get Student details
    void getStudentDetails() {
        System.out.println("Enter student
                            name: ");
        name = scanner.nextLine();

        System.out.println("Enter student USN:
        usn = scanner.nextLine();
    }
```

```
// Method to get subject marks & credits
void getMarks() {
    for (int i=0; i<8; i++) {
        System.out.println("Enter marks for
            subject " + (i+1) + ":");
        subjects[i].subjectMarks = scanner.
            nextInt();

        System.out.println("Enter credits for
            subject " + (i+1) + ":");
        subjects[i].credits = scanner.nextInt();

// calculate grade based on marks
        if(subject[i].subjectMarks >= 90) {
            subjects[i].grade = 10;
        } else if(subjects[i].subject marks >= 80)
        {
            subjects[i].grade = 9;
        } else if(subjects[i].subject marks >= 70) {
            subjects[i].grade = 8;
        } else if(subjects[i].subject marks >= 60) {
            subjects[i].grade = 7;
        } else if(subjects[i].subject marks >= 50)
        {
            subjects[i].grade = 6;
        } else if(subjects[i].subject marks >= 40) {
            subjects[i].grade = 5;
        } else {
            subjects[i].grade = 0;
        }
    }
}
```

```
// Method to compute SGPA
void computeSGPA() {
    double totalCredits = 0;
    double totalGradePoints = 0;

    for(int i=0; i<8; i++) {
        totalCredits += subjects[i].credits;
        totalGradePoints += subjects[i].grade
                * subjects[i].credits;
    }

    SGPA = totalGradePoints / totalCredits;
}

public class studentSGPA {
    Student S1 = new Student();
    S1.getStudentDetails();
    S1.getMarks();
    S1.computeSGPA();
    System.out.println("\n student details:");
    System.out.println("Name: " + S1.name);
    System.out.println("USN: " + S1.USN);
    System.out.println("SGPA: " + S1.SGPA);
}
```

Out Put :-

```
Enter studen name:
raghu
Enter student USN:
CS213
```

Enter Marks for Subject 1 :
68
Enter credits for subject 1 :
4
Enter Marks for subject 2 :
69
Enter Credits for subject 2 :
4
Enter marks for subject 3 :
59
Enter credits for subject 3 :
3
Enter marks for subject 4 :
60
Enter ~~Marks~~ credits for subject 4 :
3
Enter Marks for subject 5 :
65
Enter Credits for subject 5 :
3
Enter marks for subject 6 :
97
Enter credits for subject 6 :
1
Enter marks for subject 7 :
95
Enter credits for subject 7 :
1
Enter marks for subject 8 :
99
Enter ~~Marks~~ credits for subject 8 :
1

Date
Page

Student Details:

Name : raghu

USN : CS213

SGPA : 7.5

19/12/23

WEEK3

3.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.

Program:

```
import java.util.Scanner;

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

    public String toString() {
        String bookDetails = "Book name: " + this.name + "\n"
                + "Author name: " + this.author + "\n"
                + "Price: " + this.price + "\n"
                + "Number of pages: " + this.numPages + "\n";
        return bookDetails;
    }
}

public class BookStore{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("my name is Raghavendra N");
        System.out.print("my USN is 1BM22CS213");
        System.out.print("Enter the number of books: ");
```

```
        int n = scanner.nextInt();

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter name of the book: ");
            scanner.nextLine(); // consume the newline character
            String name = scanner.nextLine();

            System.out.print("Enter author of the book: ");
            String author = scanner.nextLine();

            System.out.print("Enter the price of the book: ");
            int price = scanner.nextInt();

            System.out.print("Enter the number of pages of the book: ");
            int numPages = scanner.nextInt();

            books[i] = new Book(name, author, price, numPages);
        }

        System.out.println("\nBook Details:");
        for (int i = 0; i < n; i++) {
            System.out.println("Book " + (i + 1) + ":\n" + books[i]);
        }
    }
}
```

LAB-3

26/12/2023

// create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include a ~~constructor to set the~~ 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 numPages;
    void BOOKS(string name, string author, int price
                                    int num page){
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String toString()
    {
        String name, author, price, numPages;
        name = "Book name: " + this.name + "\n";
        author = "Author name:" + this.author + "\n";
        price = "Price:" + this.price + "\n";
        numPages =
```

→

```
Public string to string() {
    String bookDetails = "Book name: " + this.name
                         + "\n"
        + "Author name: " + this.author + "\n"
        + "Price: " + this.price + "\n"
        + "Number of pages: " + this.numPages + "\n";
    return bookDetails;
  }
}

Public class lab3 {
    Public static void main (String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.Print("Enter the number of books:");
        int n= Scanner.nextInt();

        Book[] books = new Book[n];

        for (int i=0; i<n; i++) {
            System.out.Print("Enter name of the book
                              :");
            Scanner.nextLine();
            String name = Scanner.nextLine();

            System.out.Print("Enter author of the book:"
            Scanner.nextLine();
            String author= Scanner.nextLine();

            System.out.Print("Enter the price of book:"
            int Price = Scanner.nextInt();

            System.out.Print("Enter the number of pages of
                              the book:");
```

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

        books[i] = new Book(name, author, price,
                                    num pages);
    }

    System.out.println("\n Book Details: ");
    for (int i=0; i<n; i++) {
        System.out.println("Book " + (i+1) + ":\n" +
                                    books[i]);
    }
}
```

Sample out put :-

Enter number of ~~pages~~ Books : 2
Enter name of the book : the Legend
Enter name of author : unknown
Enter Price of the book : 999
Enter number of pages of book : 420
Enter name of the book : Malgudi days
Enter author name of book : Shankar nag
Enter Price of the book : 1000
Enter number of pages of book : 500

Book Details :
Book 1 :
Book name : the Lengend
~~Boo~~ autho name : unknown
price : 999
Numb. of pages : 420

Book 2 :
Book name : malgudi days
author name : Shankar nag
price : 1000
Number of pages : 500

WEEK4

4.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.

Program:

```
import java.util.Scanner;

abstract class Shape {
    protected int side1;
    protected int side2;

    public Shape(int side1, int side2) {
        this.side1 = side1;
        this.side2 = side2;
    }

    public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        super(length, width);
    }

    @Override
    public void printArea() {
        int area = side1 * side2;
        System.out.println("Rectangle Area: " + area);
    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {
        super(base, height);
    }
```

```
    @Override
    public void printArea() {
        double area = 0.5 * side1 * side2;
        System.out.println("Triangle Area: " + area);
    }
}
class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);
    }
    @Override
    public void printArea() {
        double area = Math.PI * side1 * side1;
        System.out.println("Circle Area: " + area);
    }
}
public class Area {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("my name is Raghavendra N");
        System.out.print("my USN is 1BM22CS213");
        System.out.println("Enter length and width for Rectangle:");
        int rectLength = scanner.nextInt();
        int rectWidth = scanner.nextInt();
        Rectangle rectangle = new Rectangle(rectLength, rectWidth);
        rectangle.printArea();

        System.out.println("Enter base and height for Triangle:");
        int triBase = scanner.nextInt();
        int triHeight = scanner.nextInt();
        Triangle triangle = new Triangle(triBase, triHeight);
        triangle.printArea();

        System.out.println("Enter radius for Circle:");
        int circleRadius = scanner.nextInt();
        Circle circle = new Circle(circleRadius);
        circle.printArea();
        scanner.close();
    }
}
```

WEEK 4

21/10/24

Develop a java program to create an abstract class named shape that contains two integers and an empty method named printbreac() Provide three classes named rectangle, triangle and circle such that each one of the classes contain the class shape Each one of the classes contain only the method Printarea () that prints the area of given shape

```
import java.util.Scanner;

abstract class shape {
    Protected int side1;
    Protected int side2;

    Public shape( int side1, int side2) {
        this.side1 = side1;
        this.side2 = side2;
    }
    Pubic abstract void Printarea();
}

class Rectangle extends shape {
    public Rectangle (int length, int width) {
        super (length, width);
    }
    Public void printare() {
        int area = side1 * side2;
        system.out.println(" Rectangle are: "+ area
    }
}
```

```
class Triangle extends Shape {
    public Triangle (int base, int height) {
        super (base, height);
    }
    Public void printArea () {
        double area = 0.5 * side1 * side2;
        System.out.println ("Triangle area : " + area);
    }
}
class Circle extends Shape {
    public Circle (int radius) {
        super (radius, 0);
    }
    Public void printArea () {
        double area = Math.PI * side1 * side1;
        System.out.println ("Circle area : " + area);
    }
}

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

        System.out.println ("Enter length and width
                             of rectangle : ");
        int rectLength = scanner.nextInt();
        int rectWidth = scanner.nextInt();
        Rectangle rectangle = new Rectangle
                              (rectLength, rectWidth);

        System.out.println ("Enter base & width
                             for Triangle : ");
        int triBase = scanner.nextInt();
```

```
int triheeght = scanner.nextInt();
Triangle Triangle = new Triangle(triBase,
                                 triheight);
triangle.PrintArea();

System.out.println("Enter radius
                    for circle:");
int circleRadius = scanner.nextInt();
circle circle = new circle(circleRadius);
circle.PrintArea();
scanner.close();
}
}
```

Output :-

```
Enter length & width for rectangle:
4
4
Rectangle area = 28

Enter base and height for Triangle:
4
8
Triangle Area = 16.0

Enter radius for circle:
9
circle are: 254.46900494077323
```

WEEK5

5.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:

a) Accept deposit from customer and update the balance.
b) Display the balance.
c) Compute and deposit interest
d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

Program:

```
import java.util.Scanner;

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

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

    public void displayBalance() {
        System.out.println("Account Balance: $" + balance);
    }
    public abstract void withdraw(double amount);
```

```
}

class CurrAcct extends Account {
    double minimumBalance;
    double serviceCharge;

    public CurrAcct(String customerName, long accountNumber, double balance) {
        super(customerName, accountNumber, "Current Account", balance);
        this.minimumBalance = 1000;
        this.serviceCharge = 50;
    }

    @Override
    public void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Remaining balance: $" +
balance);
        } else {
            System.out.println("Insufficient funds. Service charge of $" +
serviceCharge + " applied.");
            balance -= serviceCharge;
            System.out.println("Remaining balance after service charge: $" + balance);
        }
    }
}

class SavAcct extends Account {
    double interestRate;

    public SavAcct(String customerName, long accountNumber, double balance) {
        super(customerName, accountNumber, "Savings Account", balance);
        this.interestRate = 0.05; // Set interest rate (5%)
    }

    @Override
    public void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal successful. Remaining balance: $" +
```

```
balance);
        } else {
            System.out.println("Insufficient funds. Cannot complete withdrawal.");
        }
    }

    public void depositInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest deposited. Updated balance: $" + balance);
    }
}

public class Bank1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
       System.out.print("my name is Raghavendra N");
        System.out.print("my USN is 1BM22CS213");
        System.out.print("Enter customer name: ");
        String customerName = scanner.nextLine();

        System.out.print("Enter account number: ");
        long accountNumber = scanner.nextLong();

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

        System.out.print("Enter account type (Current/Savings): ");
        String accountType = scanner.next();

        Account account;
        if (accountType.equalsIgnoreCase("Current")) {
            account = new CurrAcct(customerName, accountNumber, initialBalance);
        } else if (accountType.equalsIgnoreCase("Savings")) {
            account = new SavAcct(customerName, accountNumber, initialBalance);
        } else {
            System.out.println("Invalid account type. Exiting program.");
            return;
        }

        int choice;
```

```
        do {
            System.out.println("\n1. Deposit");
            System.out.println("2. Display Balance");
            System.out.println("3. Deposit Interest for Savings Account");
            System.out.println("4. Withdraw");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter deposit amount: ");
                    double depositAmount = scanner.nextDouble();
                    account.balance += depositAmount;
                    System.out.println("Deposit successful. Updated balance: $" +
account.balance);
                    break;

                case 2:
                    account.displayBalance();
                    break;

                case 3:
                    if (account instanceof SavAcct) {
                        ((SavAcct) account).depositInterest();
                    } else {
                        System.out.println("This option is applicable for Savings Account
only.");
                    }
                    break;

                case 4:
                    System.out.print("Enter withdrawal amount: ");
                    double withdrawalAmount = scanner.nextDouble();
                    account.withdraw(withdrawalAmount);
                    break;

                case 5:
                    System.out.println("Exiting program. Goodbye!");
                    break;
```

```
                default:
                    System.out.println("Invalid choice. Please enter a valid option.");
            }

        } while (choice != 5);

        scanner.close();
    }
}
```

WEEK 5

9/04/24

// Develop a Java program to create a class Bank that maintain two accounts to store customer name, account number, type of account, Accept deposit from customer & 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

```
import java.util.Scanner;

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

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

    Public void displayBalance() {
        System.out.println("Account Balance: $" +balance);
    }

    //Abstract method for withdrawal
    public abstract void withDraw(double
                                  amount);
}
```

```
class CurrAcct extends Account {
    double minimumBalance;
    double serviceCharge;

    public CurrAcct(String customerName, long
            accountNumber, double balance) {
        super(customerName, accountNumber, "Current
                Account", balance);
        this.minimumBalance = 1000; // set minimum ba
        this.serviceCharge = 50; // set service charge
    }

    @Override
    public void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal Succesful
                Remaining balance: $" + balance);
        } else {
            System.out.println("Insufficient funds,
                service charge of $" + serviceCharge +
                " applied.");
            balance -= serviceCharge;
            System.out.println("Remaining balance
            after service charge: $" + balance);
        }
    }
}

class SavAcct extends Account {
    double interestRate;

    Public SavAcct(String customerName, long accou
            Number, double balance) {
```

```
super(customerName, accountNumber, accountType: "Savings Account", balance);
this.interestRate = 0.05;
}

@override
public void withdraw(double amount) {
    if(balance - amount >= 0) {
        balance -= amount;
        System.out.println("withdrawal successful.
                Remaining balance: $" + bal);
    }
    else {
        System.out.println("Insufficient funds.
                Cannot complete withdrawal.");
    }
}

public void depositInterest() {
    double interest = balance * interestRate;
    balance += interest;
    System.out.println("Interest deposited, updated
            : $" + balance);
}
}

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

        System.out.print("Enter customer name: ");
        String customerName = scanner.nextLine();
```

```
System.out.print("Enter acc num:");
long accountNumber = scanner.nextLong();

System.out.print("Enter account type");
String accountType = scanner.next();

Account a account;
if (accountType.equalsIgnoreCase(another
                    String: "current")) {
    account = new CurrAcct(customerName, account
                    number, initialBalance);
} else if(accountType.equalsIgnoreCase(another
                    string: "savings")) {
    account = new SavAcct(customerName, accountNum
                    , initialBalance);
} else {
    System.out.println("Invalid account
                    type. exiting program.");
    return;
}

int choice;
do {
    System.out.println("1. Deposit");
    System.out.println("2. Display Balance");
    System.out.println("3. Deposit");
    System.out.println("4. Withdraw");
    System.out.println("5. Exit");
    System.out.print("Enter choice:");

    choice = scanner.nextInt();

switch (choice) {
```

```
case 1 :
    System.out.print("Enter deposit amount: ");
    double depositAmount = scanner.nextDouble();
    account.balance += depositAmount;
    System.out.println("Deposit successful");
    break;

case 2;
    account.displayBalance();
    break;

case 3:
    if (account instanceof SavAcct) {
        ((SavAcct) account).depositInterest();
    } else {
        System.out.println("Only Sav acc");
    }
    break;

case 4:
    System.out.println("Enter withdrawal amt");
    double withdrawalAmount = scanner.nextDouble();
    account.withdraw(withdrawalAmount);
    break;

case 5:
    System.out.println("Exiting Program. Goodbye!");
    break;

default:
    System.out.println("Invalid choice.");

} while(choice != 5);
    scanner.close();
}
```

WEEK6

6.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.

Program:

```
// MainProgram.java
import CIE.Internals;
import SEE.External;
import java.util.Scanner;

public class MainProgram {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("my name is Raghavendra N");
        System.out.print("my USN is 1BM22CS213");
        System.out.println("Enter the number of students:");
        int numStudents = scanner.nextInt();

        Internals[] internalsArray = new Internals[numStudents];
        External[] externalsArray = new External[numStudents];

        for (int i = 0; i < numStudents; i++) {
            System.out.println("Enter details for Student " + (i + 1));
            System.out.print("USN: ");
            String usn = scanner.next();

            System.out.print("Name: ");
            String name = scanner.next();

            System.out.print("Semester: ");
            int sem = scanner.nextInt();
```

```
            System.out.println("Enter Internal Marks for 5 courses:");
            int[] internalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                System.out.print("Course " + (j + 1) + ": ");
                internalMarks[j] = scanner.nextInt();
            }

            internalsArray[i] = new Internals(usn, name, sem, internalMarks);

            System.out.println("Enter SEE Marks for 5 courses:");
            int[] seeMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                System.out.print("Course " + (j + 1) + ": ");
                seeMarks[j] = scanner.nextInt();
            }

            externalsArray[i] = new External(usn, name, sem, seeMarks);
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < numStudents; i++) {
            int[] internalMarks = internalsArray[i].getInternalMarks();
            int[] seeMarks = externalsArray[i].getSeeMarks();

            int[] finalMarks = new int[5];
            int totalMarks = 0;

            System.out.println("Student " + (i + 1) + " - " +
internalsArray[i].getName());

            for (int j = 0; j < 5; j++) {
                finalMarks[j] = internalMarks[j] + seeMarks[j];
                totalMarks += finalMarks[j];
                System.out.println("Course " + (j + 1) + ": " + finalMarks[j]);
            }

            System.out.println("Total Marks: " + totalMarks + "\n");
        }
    }
}
```

```
// CIE/Internals.java
package CIE;

public class Internals extends Student {
    protected int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }

    public int[] getInternalMarks() {
        return internalMarks;
    }
}

// CIE/Student.java
package CIE;

public class Student {
    protected String usn;
    protected String name;
    protected int sem;

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

    public Student() {
        this("", "", 0);
    }

    public String getName() {
        return name;
    }
}
```

```
// SEE/External.java
package SEE;

import CIE.Student;

public class External extends Student {
    protected int[] seeMarks;

    public External(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }

    public int[] getSeeMarks() {
        return seeMarks;
    }
}
```

Week - 6

23/01/2024

Classmate
Date
Page

+ Create a Package CIE which has two classes student and internals, class student should have USN, Name, Sem. The class internals derived from Student has an array that stores the internal marks scored in five courses of current sem of students create a package CIE and import both Packages.

+ Student.java

```
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() {
    make scanner S object,
    read S.USN
    read S.name
    read S.Sem
}

public void displayStudentDetails() {
    display USN, name, Sem
}
```

+ Internals.java

```
package CIE;

import java.util.Scanner;

Public class Externals extends Internals {
    protected int marks[] = new int[5];
    Protected int finalMarks[];

    public void inputCIEmarks()
    {
        initialise Scanner objects
        apply for login and read 5 set of
            marks using S objects
    }
}
```

+ Externals.java

```
package SEE;

import CIE.Internals;

import java.util.Scanner;

Public class Externals extends Internals{

    protected int marks[];
    protected int finalMarks[];

    public Externals() {
        marks: new int[5];
        final Marks = new int[5];
    }
```

```
public void inputSEEMarks() {
    Scanner s = new Scanner(System.in);
    for (int i=0; i<5; i++) {
        System.out.print("subject "+(i+1)+"
                marks:");
        marks[i] = s.nextInt();
    }
}

Public void calculatefinalMarks() {
    for(i=0; i<5; i++) {
        finalMarks[i] = marks[i]/2 + sup an.
                marks[i];
    }
public void displayFinalMarks() {
    display StudentDetails();
    for (int i=0; i<5; i++)
        System.out.println("subject "+(i+1)"
                :" + finalMarks[i]);
    }
}
```

+ main.Java.

```
import SEE.ExternalS;

class main {
    public static void main(String args[])
    {
        int numofstudents = 2;
        Externals finalMarks[] = new Externals[numof
                students];
```

```
for (int i=0 ; i < numofstudents; i++)
{
    finalMarks[i] = new Externals();
    finalMarks[i].inputStudentDetails();
        System.out.println("Enter CIE Marks");
    finalMarks[i].inputCIEMarks();
    System.out.println("Enter SEE Marks");
    finalMarks[i].inputSEEmarks();
}
System.out.println("Display data: \n");
    for (int i=0; i<numofstudents; i++)
    {
        finalmarks[i].calculateFinalMarks();
        finalMarks[i].displayFinalMarks();
    }
}
```

Sample Out Put :-

Enter no of students for Internals : 2

Enter details of students 1

USN = 01BM21CS001

Name = ANVISH

Semster : 3

Enter internal marks for 5 courses of student 1

80, 75, 90, 87, 90

Enter student details of 2.

USN: 1BMXXCSXX2.
Name: Abhishek
Semster: 3.

Enter internal marks for 5 courses of student
85, 78, 92, 87, 90.

Final marks of students 2;

Student 1: 746.
Student 2: 422.

WEEK7

7.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.

Program:

```
import java.util.Scanner;

class WrongAge extends Exception {
    public WrongAge() {
        super("Age Error");
    }

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

class InputScanner {
    protected Scanner scanner;

    public InputScanner() {
        scanner = new Scanner(System.in);
    }

    public int nextInt() {
        return scanner.nextInt();
    }
}

class Father extends InputScanner {
    protected int fatherAge;

    public Father() throws WrongAge {
        System.out.println("Enter father's age:");
```

```
        fatherAge = super.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    public void display() {
        System.out.println("Father's age: " + fatherAge);
    }
}
class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
        super();
        System.out.println("Enter son's age:");
        sonAge = super.nextInt();

        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to father's
age");
        } else if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
    public void display() {
        super.display();
        System.out.println("Son's age: " + sonAge);
    }
}
public class ExceptionHandlingDemo {
    public static void main(String[] args) {

        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

```
        System.out.println("Enter father's age: ");
        fAge = nextInt();
        if (fAge < 0) {
            throw new WrongAge("Age cannot be
                                negative");
        }
    }

    Public int getAge() {
        return fAge;
    }

    public void display() {
        System.out.println("Father's age:" +fAge);
    }
}

class son extends Father {
    private int sonAge;

    public son() throws WrongAge {
        super();
        System.out.println("Enter son's age: ");
        sonAge = nextInt();

        if (sonAge >= getFAge()) {
            throw new wrongAge("son age cannot
                    greater than or equal to father's age");
        }
        else if (sonAge < 0) {
            throw new WrongAge("Age cannot be
                                negative");
        }
```

```
}
public void displaysonAge() {
    System.out.println("son's Age:" + sonAge);
}
}

Public class ExceptionInheritance {
    Public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
            son.displaysonAge();
        } catch (WrongAge e) {
            System.out.println("Error:" +
                    e.getMessage());
        }
    }
}
```

Sample Output :-

```
Enter Father's age : 45
Enter Son's age : 60
Error : son's age cannot be greater than or equal
        to father's age.

Enter Father's age : 45
Enter son's age : -5
Error : Age cannot be negative.

Enter Father's age : 45
Enter son's age : 25
Father's age : 45
son's age : 25
```

20/01/24

8.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.

Program:

```
class DisplayThread extends Thread {
    private String message;
    private int interval;

    public DisplayThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }
    public void run() {
        try {
            while (true) {
                System.out.println(message);
                Thread.sleep(interval);
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
public class week8 {
    public static void main(String[] args) {
    System.out.print("my name is Raghavendra N");
     System.out.print("my USN is 1BM22CS213");
        DisplayThread thread1 = new DisplayThread("BMS College of Engineering",
10000);
        DisplayThread thread2 = new DisplayThread("CSE", 2000); // 2 seconds

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

WEEK 8

06/02/2024

// JAP which creates two threads, one thread displaying "BMS college of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```
class DisplayThread extends Thread {
    private string message;
    private int interval;

    Public DisplayThread(string message, int interval) {
        this.message = message;
        this.interval = interval;
    }

    @override
    Public void run() {
        while (true) {
            system.out.println(message);
            try {
                Thread.sleep(interval);
            } catch (InterruptedException e) {
                e.PrintStackTrace();
            }
        }
    }
}

public class threads {

    public static void main (string[] args) {
        // create two threads
```

```
        DisplayThread thread1 = new DisplayThread
                (message: "BMS college of Engineering",
                 interval: 2000); // 2 seconds.

        DisplayThread thread2 = new DisplayThread
                (message: "CSE", interval: 1000);
                                // 1 seconds.

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

Example output :-

```
BMS college of Engineering
CSE
CSE
BMS college of Engineering
CSE
CSE
BMS college of Engineering
CSE
CSE
BMS college of Engineering
.
.
.
```

WEEK9

9.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.

Program:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo{
   SwingDemo(){
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(275, 200);
      jfrm.setLayout(new FlowLayout());
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      JLabel jlab = new JLabel("Enter the divider and dividend:");

      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);

      JButton button = new JButton("Calculate");

      JLabel err = new JLabel();
      JLabel alab = new JLabel();
      JLabel blab = new JLabel();
      JLabel anslab = new JLabel();

      jfrm.add(err);  // to display error message
      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) {

   }
};
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());
         if (b == 0) {
            throw new ArithmeticException();
         }
         int ans = a/b;

         alab.setText("A = " + a);
         blab.setText("B = " + b);
         anslab.setText("Ans = "+ ans);
         err.setText(""); // Clear any previous error message
      }
      catch(NumberFormatException e){
         clearLabels();
         err.setText("Enter Only Integers!");
      }
      catch(ArithmeticException e){
         clearLabels();
         err.setText("B should be NON zero!");
      }
   }

   private void clearLabels() {
      alab.setText("");
      blab.setText("");
      anslab.setText("");
```

```
        }
    });

    jfrm.setVisible(true);
}

public static void main(String args[]){
    SwingUtilities.invokeLater(new Runnable(){
    System.out.print("my name is Raghavendra N");
    System.out.print("my USN is 1BM22CS213");
        public void run(){
            new SwingDemo();
        }
    });
}
```

WEEK - 9

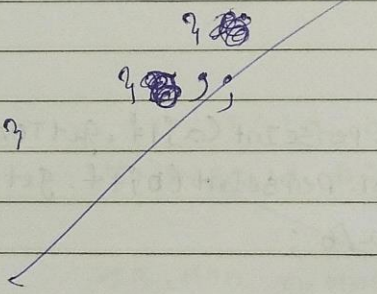
20/02/2024

// WAP that creates 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 not an integer, throw NumberFormatException. Num2 is zero ArithmaticException

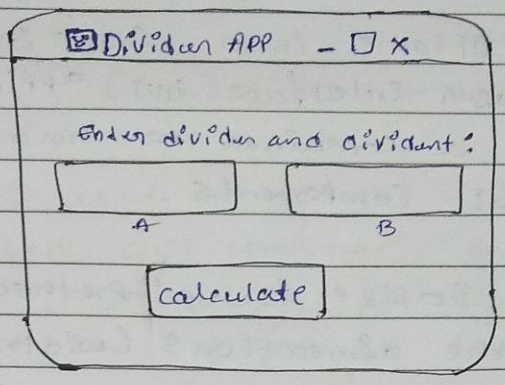
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo{
    SwingDemo(){
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(width:275, height:150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON
                                      -close);

        JLabel jlab = new JLabel(text:" Enter the
                                 dividen and divident:");

        JTextField ajtf = new JTextField(columns:8);
        JTextField bjtf = new JTextField(columns:8);

        JButton button = new JButton(text:"Calculate");

        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
}
```

```
jfrm.add(errr);
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("In A= "+a);
      blab.setText("In B= "+b);
      anslab.setText("In Ans= "+ans);
    }
    catch(NumberFormatException e) {
      alab.setText("");
      blab.setText("");
```

```
            anslab.setText(text:" ");
            err.setText(text:"Enter only integers!");
        }
        catch(ArithmeticException e){
            alab.setText(text:" ");
            blab.setText(text:" ");
            anslab.setText(text:" ");
            err.setText(text:"B should be NON zero!");
        }
    }
});

    jfrm.setVisible(b:true);
}

public static void main(String args[]){

    SwingUtilities.invokeLater(new Runnable()
    {
        public void run(){
            new SwingDemo();
        }
    });
}
```



Sample out Put :-



A = 10

B = 0

err : B should be NON zero

A = string.

B = 20

err : enter only integers!

* functions used.

1) JFrame : A JFrame is a window in a Java graphical user interface (GUI) application. It provides a container to hold and organize other GUI components.

2) setSize : setSize is a method in Java used to set the dimensions (width and height) of a GUI component, such as a JFrame, specifying its initial size.

3) setDefaultCloseOperation : This method is used in Java to set the default close operation for a JFrame. It defines the action to be taken when user clicks close button on window.

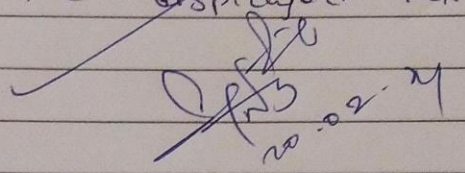
4) JLabel : JLabel is a GUI component in Java that is used to display a non-editable text or image. It provides a simple way to add descriptive text to a GUI.

5) JTextField : JTextField is a GUI component in Java that allows the user to input a single line of text. It is commonly used to receive textual input from the user.

6) addFrame : This generally refers to adding a JFrame to the display. In Java, it involves creating an instance of the JFrame class and displaying it using method like setVisible(true)

*) addactionListenner :- used to registers an ActionListener for an interactive GUI component, like a button. It listens for user actions such as button clicks and responds accordingly.

*) SetText : used to set the text component of a text-based GUI component, such as a JLabel or JTextField. It allows for dynamic updating of the displayed text.



WEEK10

10.Demonstrate Inter process Communication and deadlock

Program:

A.Deadlock

```
class A {
   synchronized void foo(B b) {
      String name = Thread.currentThread().getName();
      System.out.println(name + " entered A.foo");
      try {
         Thread.sleep(1000);
      } catch (Exception e) {
         System.out.println("A Interrupted");
      }
      System.out.println(name + " trying to call B.last()");
      b.last();
   }

   synchronized void last() {
      System.out.println("Inside A.last");
   }
}

class B {
   synchronized void bar(A a) {
      String name = Thread.currentThread().getName();
      System.out.println(name + " entered B.bar");
      try {
         Thread.sleep(1000);
      } catch (Exception e) {
         System.out.println("B Interrupted");
      }
      System.out.println(name + " trying to call A.last()");
      a.last();
   }

   synchronized void last() {
```

```
        System.out.println("Inside B.last");
    }
}

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();

    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }

    public static void main(String args[]) {
        new Deadlock();
    }
}
```

B. InterprocessCommunication:

```
class Q {
    int n;
    boolean valueSet = false;

    synchronized int get() {
        while (!valueSet)
            try {
                System.out.println("\nConsumer waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
```

```
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("\nIntimate Producer\n");
        notify();
        return n;
    }

    synchronized void put(int n) {
        while (valueSet)
            try {
                System.out.println("\nProducer waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        this.n = n;
        valueSet = true;
        System.out.println("Put: " + n);
        System.out.println("\nIntimate Consumer\n");
        notify();
    }
}

class Producer implements Runnable {
    Q q;

    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }

    public void run() {
        int i = 0;
        while (i < 15) {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {
    Q q;

    Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }
```

```
    public void run() {
        int i = 0;
        while (i < 15) {
            int r = q.get();
            System.out.println("consumed:" + r);
            i++;
        }
    }
}

class PCFixed {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}
```

Week 10 >>

16/08/2024

```
// Demonstrate Inter process Communication
        and deadlock

public class DeadlockExample {

    public static void main(String[] args) {
        final SharedResource sharedResource = new
                SharedResource();

        Thread process1 = new Thread(() -> {
            try {
                sharedResource.method1();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        });

        Thread process2 = new Thread(() -> {
            try {
                sharedResource.method2();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        });

        process1.start();
        process2.start();
    }
}

class SharedResource {
    private final Object lock1 = new Object();
    private final Object lock2 = new Object();
```

```
public void method2() throws InterruptedException {
    synchronized (lock2) {
        System.out.println("method2 acquired lock2");
        Thread.sleep(millis:1000);

        synchronized (lock1) {
            System.out.println("method2 acquired lock1");
        }
    }
}
```

Sample output:

```
main thread entered A.foo
        thread entered B.bar
main Thread trying to call B.last()
Inside A.last
Back in main thread
Racey Thread trying to call A.last()
Inside A.last
Back in other thread
```

15/2/2024

// An incorrect implementation of a producer and consumer. (inter process communication)

```
class Q {
    int n;

    synchronized void put(int n) {
        this.n = n;
        System.out.println("put : " + n);
    }
}

synchronized int get() {
    System.out.println("Got : " + n);
    return n;
}

Synchronized void put(int n) {
    this.n = n;
    System.out.println("Put : " + n);
}
}

class Producer implements Runnable {
    Q q;

    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }
```

```
Public void run() {
    int i=0;
    while(i<15) {
        q.put(i++);
    }
}

class consumer implements Runnable {
    Q, p;

    consumer(Q q) {
        this.q = q;
        new Thread(this, "consumer").start();
    }

    public void run() {
        int i=0;
        while(i<15) {
            int r = q.get();
            i++;
        }
    }
}

class PC {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new consumer(q);
        System.out.println("press control-c
                            to stop.");
    }
}
```

Sample Output :-

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

Got : 14
Got : 14
Got : 14
Got : 14
Got : 14
Got : 14
Got : 14
Got : 14
Got : 14
Got : 14
Got : 14
Got : 14
Got : 14
Got : 14

13 | 2 | 24