B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

NIHAL REDDY S
1BM22CS179

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
11	Complete Scanned Observation Book	12/12/2023- 20/02/2024

Lab Programs

1. 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 discriminate b 2 -4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
class quad {
   public static void main(String args[]) {
      int a, b, c;
      double r1, r2, d;
      Scanner s = new Scanner(System.in);
      System.out.println("Nehal A K\n1BM22CS176");
      System.out.println("Enter the coefficients of a,b,c");
      a = s.nextInt();
      b = s.nextInt();
      c = s.nextInt();
      while (a == 0) {
         System.out.println("Not a quadratic equation");
         System.out.println("Enter a non zero value for a:");
         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("Root1 = " + r1 + " + i" + r2);
            System.out.println("Root1 = " + r1 + " - i" + r2);
        }

    }
}
```

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.

```
import java.util.Scanner;
class Subject {
    int subjectMarks;
    int credits;
    int grade;
}

class Student
    { String
    name;
    String usn;
    double SGPA;
    Subject[] subject;
    Scanner s;



    Student() {
        int i;
        subject = new Subject[9];
```

```
        subject[i] = new Subject();
    s = new Scanner(System.in);
  }


  void getStudentDetails()
    { System.out.println("Enter student
    name"); name = s.next();
    System.out.println("Enter student usn");
    usn = s.next();
  }


  void getMarks() {
    int i;
    for (i = 0; i < 9; i++) {
      System.out.println("Enter marks for subject " + (i + 1));
      subject[i].subjectMarks = s.nextInt();
      System.out.println("Enter credits for subject " + (i + 1));
      subject[i].credits = s.nextInt();


      if (subject[i].subjectMarks >= 90) {
        subject[i].grade = 10;
      } else if (subject[i].subjectMarks >= 80 && subject[i].subjectMarks
< 90) {
        subject[i].grade = 9;
```

```
        } else if (subject[i].subjectMarks >= 70 && subject[i].subjectMarks
< 80) {
            subject[i].grade = 8;
        } else if (subject[i].subjectMarks >= 60 && subject[i].subjectMarks
< 70) {
            subject[i].grade = 7;
        } else if (subject[i].subjectMarks >= 50 && subject[i].subjectMarks
< 60) {
            subject[i].grade = 6;
        } else if (subject[i].subjectMarks >= 40 && subject[i].subjectMarks
< 50) {
            subject[i].grade = 5;
        } else
            { System.out.println("Failed"
            ); System.exit(0);
        }
    }
  }


  void computeSGPA()
    { int totalCredits = 0;
    int creditsGained = 0;
    int i;


    for (i = 0; i < 9; i++) {
      totalCredits += subject[i].credits;
```

```
            creditsGained += subject[i].credits * subject[i].grade;
        }

        SGPA = (double) creditsGained / totalCredits;
    }


    void displayResult()
        { System.out.println("Name = " +
        name); System.out.println("Usn = " +
        usn); System.out.println("SGPA = " +
        SGPA);
}

public class Main {
    public static void main(String args[])
        { Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        s1.displayResult();
    }
}
```

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.

```
import java.util.Scanner;


class Book
  { String
  name; String
  author; int
  price;
  int numPages;


  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";
```

```
    }
}

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

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



        Book[] books = new Book[n];



        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for Book " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = s.next();
            System.out.print("Author: ");
            String author = s.next();
            System.out.print("Price: ");
            int price = s.nextInt();
            System.out.print("Number of pages: ");
            int numPages = s.nextInt();
            books[i] = new Book(name, author, price, numPages);
        }
```

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

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.

```
import java.util.*;

abstract class AbsArea {
    int a, b;

    AbsArea(int x)
        { a = x;
    }

    AbsArea(int x, int y) {
        a = x;
        b = y;
    }

    abstract void area();
}

class rec extends AbsArea {
```

```
   rec(int a, int b) {
      super(a, b);
   }

   void area() {
      System.out.println("The area of the rectangle is: " + a * b);
   }
}

class tri extends AbsArea {
   tri(int a, int b) {
      super(a, b);
   }

   void area() {
      System.out.println("The area of the triangle is: " + (a * b) / 2);
   }
}

class cir extends AbsArea {
   cir(int a) {
      super(a);
   }

   void area() {
      System.out.println("The area of the circle is: " + 3.14 * a * a);
   }
}

class Main {
   public static void main(String args[])
{ System.out.println("This is done by Nehal
AK\n1BM22CS176");
      int x, y;
      Scanner n = new Scanner(System.in);

      // Input for rectangle dimensions (x and y) with validation
```

```
        x = n.nextInt();
        y = n.nextInt();
        if (x < 0 || y < 0) {
            System.out.println("Invalid input for rectangle. Please enter
positive values.");
            // You might want to handle this situation differently, such as
asking the user to enter values again.
            System.exit(1); // Exiting with status code 1 (indicating abnormal
exit)
        }

        AbsArea r = new rec(x, y);
        r.area();

        // Input for triangle dimensions (x and y) with validation
        System.out.println("Give input for triangle");
        x = n.nextInt();
        y = n.nextInt();
        if (x < 0 || y < 0) {
            System.out.println("Invalid input for triangle. Please enter positive
values.");
            System.exit(1);
        }

        AbsArea t = new tri(x, y);
        t.area();

        // Input for circle radius (x) with validation
        System.out.println("Give input for circle");
        x = n.nextInt();
        if (x < 0) {
            System.out.println("Invalid input for circle. Please enter a positive
value.");
            System.exit(1);
        }

        AbsArea c = new cir(x);
```

```
        c.area();
      }
   }
```

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.

```
import java.util.Scanner;
class Account {
   String CustomerName;
   double AccNo, Balance;

   Account(String CustomerName, double AccNo, double Balance) {
      this.CustomerName = CustomerName;
      this.AccNo = AccNo;
      this.Balance = Balance;
   }

   public void Deposit(double Amount) {
      Balance += Amount;
      System.out.println("DEPOSIT SUCCESSFUL");
      DisplayBalance();
   }
```

```
    void DisplayBalance() {
        System.out.println("BALANCE:" + Balance);
    }
}

class CurrentAccount extends Account {
    double MinBalance = 500.0;
    double Charges = 10.0;

    CurrentAccount(String CustomerName, double AccNo, double Balance) {
        super(CustomerName, AccNo, Balance);
    }

    void Withdraw(double Amount) {
        if (Balance >= Amount) {
            Balance -= Amount;
            System.out.println(Amount + " withdrawn successfully");
            DisplayBalance();
        } else {
            System.out.println("insufficient Balance");
        }
    }

    void UpdateBalance() {
        if (Balance <= MinBalance) {
            Balance -= Charges;
```

```
            System.out.println("service charge applied for maintaining low
balance");
            DisplayBalance();
        }
    }
}

class SavingsAccount extends Account {
    SavingsAccount(String CustomerName, double AccNo, double Balance) {
        super(CustomerName, AccNo, Balance);
    }

    double interest = 0.05;

    void UpdateBalance() {
        Balance = Balance + (interest * Balance);
        DisplayBalance();
    }

    void Withdraw(double Amount) {
        if (Balance >= Amount)
            { Balance -= Amount;
            DisplayBalance();
        } else {
            System.out.println("insufficient Balance");
        }
    }
```

```
}

class Bank {
   public static void main(String args[]) {
      String CustomerName;
      double AccNo, Balance;
      double amt, amt1;
      Scanner in = new Scanner(System.in);

      System.out.println("enter name:");
      CustomerName = in.next();

      System.out.println("enter AccNo:");
      AccNo = in.nextDouble();
      System.out.println("enter Balance:");
      Balance = in.nextDouble();
      System.out.println("enter amount to deposit");
      amt = in.nextDouble();
      System.out.println("enter amount to withdraw");
      amt1 = in.nextDouble();

      CurrentAccount c = new CurrentAccount(CustomerName, AccNo,
Balance);
      c.Deposit(amt);
      c.Withdraw(amt1);
      c.UpdateBalance();
```

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


        SavingsAccount s = new SavingsAccount(CustomerName, AccNo,
Balance);
        s.Deposit(amt);
        s.Withdraw(amt1);
        s.UpdateBalance();
    }
}
```

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

```
package CIE;
public class Student
{
    public String name;
    public String usn;
    public int sem;

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

```
package CIE;


public class Internals extends CIE.Student
{
    public int [] InternalMarks;


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

```
package SEE;
import CIE.Student;

public class Externals extends Student
  {
      public int [] SeeMarks;

      public Externals(String name , String usn , int sem , int []SeeMarks)
      {
          super(name , usn , sem);
           this.SeeMarks=SeeMarks;
      }
  }
```

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


public class FinalMarks
{
    public static void main(String [] args)
    {
        Scanner s1=new Scanner(System.in);

       System.out.println("Enter the number of Students");
        int n=s1.nextInt();
```

```
        String []names=new String[n];
          String []usn=new String[n];
         int []sem = new int[n];
         int [][] InternalMarks = new int[n][5];
        int [][] SeeMarks = new int[n][5];

for(int i=0 ; i<n; i++)
{
      System.out.println("Enter details for Student" + (i+1) + ":");
      System.out.println("Name:");
      names[i]=s1.next();
      System.out.println("USN:");
      usn[i]=s1.next();
      System.out.println("SEM:");
      sem[i]=s1.nextInt();

      System.out.println("Enter Internal marks for 5 courses:");
for(int j=0; j<5; j++)
{
      System.out.println("Course"+(j+1)+":");
      InternalMarks[i][j]=s1.nextInt();
}
    System.out.println("Enter Internal marks for 5 courses:");
for(int j=0; j<5; j++)
{
   System.out.println("Course"+(j+1)+":");
   SeeMarks[i][j]=s1.nextInt();
}
}

int [][]FinalMarks =  new int[n][5];
for(int i=0 ; i<n ; i++)
{
        Internals I1 = new Internals(names[i] , usn[i] , sem[i] , InternalMarks[i]);
        Externals E1 = new Externals(names[i] , usn[i] , sem[i] , SeeMarks[i]);


for(int j=0; j<5 ;j++)
{
        FinalMarks[i][j] = I1.InternalMarks[i] + E1.SeeMarks[j];
 }
        System.out.println("Finals Marks for " + n+ "Students in 5 courses:");
for(i=0 ;i<n ;i++)
{
        System.out.println(names[i] +":");

for(int j=0; j<5;j++)
{
        System.out.println(FinalMarks[i][j] +  ":");
```

```
}
        System.out.println();
      }
    s1.close();
}}
}
```

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.

```
import java.util.Scanner;


class WrongAge extends Exception {
 WrongAge(String message) {
        super(message);
  }

}

class InputScanner {
  static Scanner sc = new Scanner(System.in);
}

class Father extends InputScanner {
  int fatherAge;

  Father() throws WrongAge
    { System.out.println("Enter Father's
    age"); fatherAge = sc.nextInt();
    if (fatherAge < 0) {
       throw new WrongAge("Age cannot be negative");

    }
```

```
    }

    void display() {
        System.out.println("Father's age is " + fatherAge);
    }

}

class Son extends Father {
    int sonAge;

    Son() throws WrongAge
        { System.out.println("Enter son's
        age"); sonAge = sc.nextInt();
        if (sonAge > fatherAge) {
            throw new WrongAge("Son's age cannot be greater than father's age");
        } else if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }

    }

    void display() {
        System.out.println("Son's age is " + sonAge);
    }

}

public class ExceptionHandling {
    public static void Main(String args[]) {
        try{
            Son son= new Son();
            son.display();
        }
        catch(WrongAge e){
            System.out.println("Exception "+ e.getMessage());
        }

    }
    }
```

Lab - 1

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 eqn");
        System.out.println(" Enter non-zero a value");
        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(" Root = " + 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 & 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("Root1 = " + r1 + " +i" r2);
        System.out.println("Root2 = " + r1 + " -i" + r2);
    }
  }
}
class QuadraticMain{
    public static void main(String args[]){
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
    }
}
```

Output

Enter the coefficients of a, b, c

4 -3 2

Roots are real & distinct

Root1 = 2 Root2 = +4

Enter the coefficients of a, b, c

1 2 1

Roots are real and equal

Root = -1

Lab - 2

Student marks and SGPA

```
import java.util.Scanner;
class Subject {
        int subjectMarks;
        int credits;
        int grade;
}
class Student {
        String name;
        String USN;
        double SGPA;
        Subject[] subject;
        Scanner s;

    void int Student() {
        int i;
        subject = new Subject[9];
        for(i=0; i<9; i++){
            subject[i] = new Subject();
        }
    Scanner s2 = new Scanner(System.in);
}
    void getStudentDetails() {
        System.out.println("Enter student name:");
        name = s.next();
        System.out.println("Enter student USN:");
        USN = s.next();
```

```
void getMarks() {
    int i;
    for(i=0; i<9; i++){
    System.out.println("Enter marks for subject " + (i+1));
    Subject[i].subjectMarks = s.nextInt();
    System.out.println("Enter credits for subject " + (i+1));
    subject[i].credits = s.nextInt();
    if(subject[i].subjectMarks >=90) {
        subject[i].grade = 10;
    }
    else if(subject[i].subjectMarks >= 80){
        subject[i].grade = 9;
    }
    else if(subject[i].subjectMarks >= 70){
        subject[i].grade = 8;
    }
    else if(subject[i].subjectMark >= 60){
        subject[i].grade = 7;
    else if(subject[i].subjectMarks >= 50){
        subject[i].grade = 6;
    else if(subject[i].subjectMark >= 40){
        subject[i].grade = 5;
    else {
        System.out.println("Failed");
        System.exit(0);
    }
}
}
void computeSGPA{
    int totalCredits = 0;
    int creditsGained = 0;
    int i;
```

```
for (i=0 ; i<9 , i++)
  {
   totalCredits += subject[i].credits;
   creditsGained += subject[i].credits * subject[i].grade
  }
  SGPA = double (creditsGained / totalCredits);
}
void displayResult() {
   System.out.println("Name:" + name);
   System.out.println("USN: " + USN);
   System.out.println("SGPA: " + SGPA);
  }
public class Main{
   public static void main (String args[]) {
    Student s1 = new Student();
    s1. getStudentDetails;
    s1. getMark;
    s1. computeSGPA();
    s1. displayResult();
  }
}
```

output

Enter student name

Nikhil

Enter USN

1BM22CS179

Enter marks for subject 1

90

Enter credits for subject 1

4

:

Lab-3

Java program for a library to store book info.

```
import java.util.Scanner;
class Books {
    String name;
    String author;
    int price;
    int numPages;
    Books (name, author, price, numPages) {
        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 = "Book price : " + this.price + "\n";
    numPages = "no. of pages : " + this.numPages + "\n";
    return (name + author + price + numPages);
}
}
class Books_main {
    public static void main (String args[]) {
        Scanner s = new Scanner (System.in);
        int n;
        String name, author;
        int price, numPages, $;
        n = s.nextInt();
```

```
Books[] b = new Books[n];
System.out.println("\n");
for (i=0; i<n ; i++) {
    System.out.println("Enter name of book : " + i);
    name = s.nextLine();
    System.out.println("Enter book's author name:");
    author = s.nextLine();
    System.out.println("Enter the price of book" + i);
    price = s.nextInt();
    System.out.println("Enter pages in book" + i);
    numPages = s.nextInt();
    s.nextLine();
    b[i] = new Books(name, author, price, numPages);
    System.out.
}
System.out.println("the book details are:");
for (i=0 ; i<n, i++) {
        System.out.println(b[i].toString());
    }
}
}
```

Output

```
Enter the number of books: 2
Enter the details for book 1
Enter name of book 1 Trump
Enter book's author name: Donald
Enter the price of book 1 300
Enter the pages in book 1 40
```

Enter name of book 2 Friends
Enter book's author name Dale
Enter the price of book2 4000
Enter pages in book 2 250

Details of the books:

Book 1:

Book name: Trump
Book author name: donald
Book price: 700
Number of pages: ~~250~~ 40

Book 2:

Book name: Friends
Author name: Dale
Book price: 4000
No. of pages: 250

Run
23/2/2024

SURYA Gold
Date 2-09-24 Page

Lab - 4

Q. Abstract class shape, three classes Rectangle, Triangle and Circle extending class shape & method printArea().

```
import java.util.Scanner;

class inputScanner {
    int a, b;

    void input() {
    Scanner in = new Scanner(System.in);
    System.out.println("Enter two numbers");
    a = in.nextInt();
    b = in.nextInt();
    }
}

abstract class Shape extends inputScanner {
    abstract void printArea();
}

class Rectangle extends Shape {
    void printArea() {
        System.out.println("THE AREA OF THE RECTANGLE
                           (a*b));
    }
}

class Triangle extends Shape {
    void printArea() {
        System.out.println("the area of the triangle:" + (a*b
    }
}
```

```
class Circle extends Shape {
    void printArea() {
        System.out.println("the area of circle = " + (3.14*a*a));
    }
}

public class Main {
    public static void main (String args[]) {
        Scanner sc = new Scanner (System.in);
        double a,b;

        System.out.println("enter side of rectangle : ");
        a = sc.nextDouble ();
        b = sc.nextDouble ();
        Rectangle r = new Rectangle (a,b);

        System.out.println("Enter height & base of triangle : ");
        a = sc.nextDouble ();
        b = sc.nextDouble ();
        Triangle t = new Triangle (a,b);

        System.out.println("Enter radius of circle :");
        a = sc.nextDouble ();
        Circle c = new Circle (a,a);

        Shape s;
        s = r;
        s.printArea ();
        s = t;
        s.printArea ();
        s = c;
        s.printArea ();
    }
}
```

OUTPUT

Enter sides of rectangle:

1

2

Enter height and base of triangle:

5

2

Enter radius of circle:

5

Area of rectangle = 2.0

Area of triangle = 5.0

Area of circle = 78.53975

23/2/24

Lab - 5

Java program for bank

```
import java.util.Scanner;
class Account {
    String CustomerName;
    double AccNo, Balance;
    Account (String customerName, double AccNo, double Balance){
        this.customerName = customerName;
        this.AccNo = AccNo;
        this.Balance = Balance;
    }
public void Deposit (double Amount){
    Balance += Amount;
    System.out.println (" Deposit successful ");
    DisplayBalance();
    }
void DisplayBalance () {
    System.out.println (" Balance " + Balance);
    }
}
class CurrentAcc extends Account{
    double minBalance = 500.0;
    double charges = 10.0;
    CurrentAcc(String CustomerName, double AccNo, double balance){
        super (CustomerName, AccNo, Balance);
    }
void Withdraw (double Amount) {
    if (Balance >= Amount){
        Balance -= Amount;
        System.out.println (Amount + " withdrawn successfully");
        DisplayBalance();
    } else {
```

```
system.out.println("Insufficient balance");
DisplayBalance();
}
}
void UpdateBalance(){
if (Balance <= MinBalance){
Balance -= charges;
system.out.println("service charge for maintaining low
balance");
DisplayBalance();
}
}
Class SavingAccount extends Account{
SavingsAccount(String CustomerName, double AccNo, double
balance){
super(CustomerName, AccNo, Balance);
}
double intrest = 0.05;
void updateBalance(){
Balance = Balance + (intrest * Balance);
DisplayBalance();
}

class Bank{
public static void main(String args[]){
String CustomerName;
double AccNo, Balance;
double Amt, amtd;
Scanner in = new Scanner(System.in);
system.out.println("Enter name:");
CustomerName = in.next();
```

```
System.out.println(" Enter Acc No ");
AccNo = in.nextDouble();
System.out.println(" Enter Balance :");
Balance = in.nextDouble();
System.out.println(" Enter amount to deposit ");
amt = in.nextDouble();
System.out.println(" Enter amount to withdraw");
amt1 = in.nextDouble();
CurrentAcc c = new CurrentAcc(CustomerName, AccNo, Balance);
c.Deposit(Amt);
c.withdraw(amt1);
c.updateBalance();
SavingsAccount s = new SavingsAccount(CustomerName, AccNo, Balance);
s.Deposit(amt);
s.withdraw(amt1);
s.updateBalance();
}
}
```

Output:

Enter Name: Nihal
Enter Acc No: 110
Enter balance: 90000
Enter amount to deposit: 10000
Enter amount to withdraw: 5000

DEPOSIT SUCCESSFUL
BALANCE = 100000.0
5000.0 WITHDRAWN SUCCESSFULLY

Lab-6

// Student

```
package CIE;
public class Student {
    public string name;
    public string USN;
    public int sem;

public Student (string name, string USN, int sem){
    this.name = name;
    this.USN = USN;
    this.sem = sem;
  }
}
```

// Internal

```
package CIE;
public class Internals extends student {
    public int [] Internalmarks;
    public Internals (string name, string USN, int sem, int[] internalm){
        super (name, USN, sem);
        this.InternalM = Internalm;
      }
}
```

// External

```
package SEE;
import CIE, Student;
public class Externals extends Student {
    public int [] seemarks;
    public Externals (string name, string USN, int sem, int[] seem){
        super (Name, USN, sem);
```

```
        this.seem = seem;
    }
}
```

7 final marks

```
import CIE student;
import CIE Externals;
import SEE External;
import java.util.scanner;

public class final marks {
    public static void main (string args[]){
        scanner s = new scanner (System.in);
        system.out.println("Enter no. of students:");
        int n= s.nextInt();

string[] names = new string [n];
string [] USN = new string [n];
int[] sem= new int [n];
int[][] internalm = new int [n][5];
int[][] seem = new int [n][5];

for (int i=0; i<n; i++){
    System.out.println("Enter student details" + (i+1));
    System.out.println("Name:");
    names[i] = sc.next();
    System.out.println("USN:");
    USN[i] = sc.next();
    System.out.println("sem");
    sem[i] = sc.nextint();
    System.out.println("Enter internal marks for 5 courses");
```

```
for (int j=0; j<5; j++){
System.out.println("Course" + (j+1) + ":");
Internal marks[i][j] = sl.nextInt();
}

for (j=0; j<5; j++){
    System.out.println("Course" + (j+1));
    seem[i][j] = sl.nextInt();
}

int[][] finalm = new int[n][5];
for(i=0; i<n; i++){
    Internals Il = new Internals(names[i], USN[i], sem[i], Internalm[i]);
    Externals El = new Externals(names[i], USN[i], sem[i], Externalm[i]);

for(int j=0; j<5; j++){
    finalm[i][j] = Il.Internalm[i] + El.seem[i];
}

System.out.println("Final marks for "+n+" student in 5 courses");
for(i=0; i<n; i++){
    System.out.println(name[i] + ":");
    for(int j=0; j<5; j++){
        System.out.println("Final marks[i][j] + ":");
}
}
sl.close();
}
}
```

Final Internals

Output

Enter no. of students : 1
Enter name : A
Enter USN : 1
Enter Sem : 1
Enter internal mark for 5 courses
16
30
10
9
9
Enter see mark for 5 courses
10
1
9
10
9
Final marks of 5 courses
A : 17.

week-7/ lab-7

```
import java.util.Scanner;

class WrongAge extends Exception {
    WrongAge() {
        super("Age Error");
    }
    WrongAge(String message) {
        super(message);
    }
}
class InputScanner {
    Scanner s = new Scanner(System.in);
}
class Father extends InputScanner {
    int FatherAge;

    Father() throws WrongAge {
        System.out.print("Enter father's age: ");
        FatherAge = s.nextInt();
        if (FatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
    void display() {
        System.out.println("Father's age: " + FatherAge);
    }
}
class Son extends Father {
    int SonAge;

    Son() throws WrongAge {
        super();
```

```
System.out.print("Enter son's age: ");
sonage = S.nextInt();
if (sonAge >= fatherage) {
    throw new WrongAge("Son's age cannot be >= father's age");
} else if (sonage < 0) {
    throw new WrongAge("Son's age cannot be -ve");
}
}
void display() {
    super.display();
    System.out.println("Son's age:" + sonage);
}
}

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

o/p

```
Enter father's age
60
Enter son's age
21
Father's age is 60
Son's age is 21.
```

Lab-8

1)
```
class MessagePrinter implements Runnable {
    private final String message;
    private final long intervalMillis;

    public MessagePrinter(String message, long intervalMillis) {
        this.message = message;
        this.intervalMillis = intervalMillis;
    }

    @Override
    public void run() {
        while (true) {
            try {
                System.out.println(message);
                Thread.sleep(intervalMillis);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Thread thread1 = new Thread(new MessagePrinter("BMSCE",
        Thread thread2 = new Thread(new MessagePrinter("CSE",

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

```
try {
    Thread.sleep(3000);
} catch (InterruptedException e) {
    System.out.print("Main thread interrupted");
}

thread1.interrupt();
thread2.interrupt();

System.out.println("Main thread exiting");
}
}
```

O/P

BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
Main thread exiting
Thread1 interrupted
Thread2 interrupted.

Pm
22/2/2024

lab-9

user - interface to perform integer divisions.

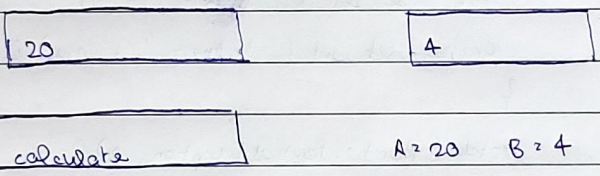
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter 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);
        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) {
    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("");
        err.setText("Enter Only Integers!");
    }
      catch (ArithmeticException e) {
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("B should be NON zero!");
      }
  }
};
jfrm.setVisible(true);
}
```

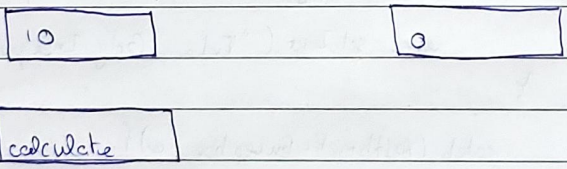
```
public static void main (String args[]){
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new swingDemo();
        }
    });
}
}
```

Output

Enter the divisor and dividend



Enter the divisor and dividend.



B should be NON zero!

AWT functions

1. Jframe: It is a class in Java that is part of the swing library, which is used for creating GUI's in Java applications.

2. setsize(): setsize() is a method which is used with components such as Jframe & Jpanel to set their size.

3. setLayout(): It is a method which is used to set the layout manager for a container such as JFrame or any other container component.

4. JLabel: It is a class which is used to display non-editable text or images on a GUI.

5. setDefaultCloseOperation: It is a method in Java swing used to specify the default close operation for a "JFrame".

6. JTextfield: Class in java swing that produces a text input field in a GUI. It allows the user to enter and edit single text line.

7. addActionListener: It is a method in Java swing that is used to register an action for a component. eg: button that generates action events.

8. setText: It is a method used in Java swing to set the text content of text based component.

Rm
23/2/2024

SURYA Gold

Date Page

Lab-10

Demonstrate IPC and deadlock

```
class Q {
  int n;
  boolean valueSet = false;
  synchronized int get() {
    while (valueSet) {
      try {
        System.out.println("\n Consumer waiting \n");
        wait();
      } catch (InterruptedException e) {
        System.out.println("InterruptedException caught");
      }
      System.out.println("Got: " + n);
      valueSet = false;
      System.out.println("\n Producer waiting \n");
      wait();
    } catch (InterruptedException e) {
        System.out.println("I.E. caught");
    }
    this.n = n;
    valueSet = true;
    System.out.println("Put: " + n);
    System.out.println("\n Initiate 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 Main {
    public static void main (String args[]) {
        Q q = new Q();
        new Producer (q);
        new consumer (q);
        System.out.println("Press ctrl+C to stop");
    }
}
```

O/P

Press Ctrl + C to stop

Put : 0

Initiate consumer

Producer waiting

Got : 0

Initiate producer's

Put : 1

:

:

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

Output

MainThread entered A.foo

RacingThread entered B.bar

MainThread trying to call B.bar()

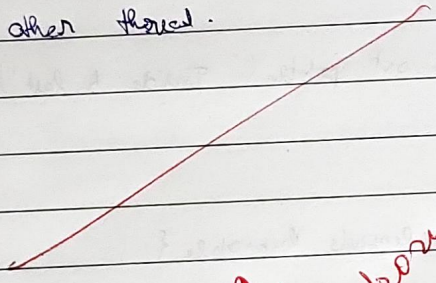
Inside A.bar()

Back in MainThread

RacingThread trying to call A.bar()

Inside A.bar

Back in otherThread.



23/2/2024