



B. M. S. COLLEGE OF ENGINEERING
(AUTONOMOUS COLLEGE UNDER VTU, BELGAUM)
BANGALORE – 560019

2022-23

LAB RECORD

OBJECT ORIENTED JAVA PROGRAMMING (23CS3PCOOJ)

Submitted by :

NAME : Shreya S Rudagi

USN : 1BM22CS267

SEMESTER: III

SECTION : E

Submitted to
Dr. Seema Patil
Assistant Professor
Dept. of CSE, BMSCE

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 discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions

```
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("Root1 = " + r1 + " + i" + r2);
    System.out.println("Root1 = " + r1 + " - i" + r2);

}

}

class QuadraticMain {

    public static void main(String args[]) {

        Quadratic q = new Quadratic();
        q.getd();

        q.compute();
        System.out.println("Shreya S Rudagi-1BM22CS267");

    }

}

```

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.*;  
  
class Subject {  
  
    int subjectMarks;  
  
    int credits;  
  
    int grade;  
  
}  
  
  
class Student {  
  
    Subject subject[];  
  
    String name;  
  
    String usn;  
  
    double sgpa;  
  
    Scanner sc;  
  
  
    Student() {  
  
        int i;  
  
        subject = new Subject[8];  
  
        sc = new Scanner(System.in);  
  
        for (i = 0; i < 8; i++)  
  
            subject[i] = new Subject();  
  
    }  
  
  
    void getstudentdetails() {  
  
        System.out.println("Enter your name:");  
  
        this.name = sc.next();  
  
        System.out.println("Enter your USN:");  
  
        this.usn = sc.next();  
    }  
}
```

```
}
```

```
void getMarks() {  
    for (int i = 0; i < 8; i++) {  
        System.out.println("Enter marks for subject " + (i + 1) + ":");  
        subject[i].subjectMarks = sc.nextInt();  
        System.out.println("Enter credits for subject" + (i + 1) + ":");  
        subject[i].credits = sc.nextInt();  
        subject[i].grade = subject[i].subjectMarks / 10 + 1;  
        if (subject[i].grade == 11)  
            subject[i].grade = 10;  
        if (subject[i].grade <= 4)  
            subject[i].grade = 0;  
    }  
}
```

```
void computeSGPA() {  
    int effectiveScore = 0;  
    int totalCredits = 0;  
    for (int i = 0; i < 8; i++) {  
        effectiveScore += (subject[i].grade * subject[i].credits);  
        totalCredits += subject[i].credits;  
    }  
    sgpa = (double) effectiveScore / (double) totalCredits;  
}
```

```
}
```

```
class Main {
```

```
public static void main(String[] args) {  
    Student s1 = new Student();  
    s1.getstudentdetails();  
    s1.getMarks();  
    s1.computeSGPA();  
    System.out.println("Name=" + s1.name);  
    System.out.println("USN:" + s1.usn);  
    System.out.println("SGPA=" + s1.sgpa);  
}  
}
```

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 Books
{
    String name;
    String author;
    int price;
    int numPages;

    Books(String name, String author, int price, int 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="Price:" +this.price+ "\n";
        numPages="Number of pages:" +this.numPages+ "\n";
        return name+author+price+numPages;
    }
}
```

```
}
```

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

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

        Books b[];
        b=new Books[n];

        for(i=0;i<n;i++)
        {
            System.out.println("Enter the details of book" + (i+1) + ":");

            System.out.println("Enter the name of the book:");
            name=s.next();

            System.out.println("Enter the author name:");
            author=s.next();

            System.out.println("Enter the price:");
            price=s.nextInt();
        }
    }
}
```

```
System.out.println("Enter the number of pages:");
numPages=s.nextInt();

b[i]=new Books(name,author,price,numPages);

}

System.out.println("Book Details:");
for(i=0;i<n;i++)
{
    System.out.println(b[i]);
}

}
```

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.*;
import java.math.*;

class InputScanner {

    Scanner sc;

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

}

abstract class Shape extends InputScanner {

    double a;
    double b;

    abstract void getInput();

    abstract void displayArea();

}

}
```

```
class Rectangle extends Shape {  
    void getInput() {  
        System.out.println("Enter the length and breadth:");  
        a = sc.nextDouble();  
        b = sc.nextDouble();  
    }  
}
```

```
void displayArea() {  
    System.out.println("Area of rectangle is :" + (a * b));  
}  
}
```

```
class Triangle extends Shape {  
    void getInput() {  
        System.out.println("Enter the length and height:");  
        a = sc.nextDouble();  
        b = sc.nextDouble();  
    }  
}
```

```
void displayArea() {  
    System.out.println("Area of triangle is :" + (a * b * 0.5));  
}  
}
```

```
class Circle extends Shape {  
    void getInput() {  
        System.out.println("Enter the radius:");  
        a = sc.nextDouble();  
    }  
}
```

```
void displayArea() {  
    System.out.println("Area of circle is :" + (a * a * Math.PI));  
}  
}  
  
class ShapeMain {  
    public static void main(String[] args) {  
        Rectangle r = new Rectangle();  
        Triangle t = new Triangle();  
        Circle c = new Circle();  
        r.getInput();  
        r.displayArea();  
        t.getInput();  
        t.displayArea();  
        c.getInput();  
        c.displayArea();  
    }  
}
```

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.

```
import java.util.Scanner;

class account {

    String name;
    int accno;
    String type;
    double balance;

    account(String name, int accno, String type, double balance) {
        this.name = name;
        this.accno = accno;
        this.type = type;
        this.balance = balance;
    }

    void deposit(double amount) {
        balance += amount;
    }
}
```

```
}

void withdraw(double amount) {
    if ((balance - amount) >= 0) {
        balance -= amount;
    } else {
        System.out.println("Insufficient balance,cant withdraw");
    }
}
```

```
void display() {
    System.out.println("Name:" + name + "\nAccno:" + accno + "\nType:" + type +
"\nBalance:" + balance);
}
```

```
class savAcct extends account {
```

```
private static double rate = 5;
```

```
savAcct(String name, int accno, double balance) {
```

```
super(name, accno, "Savings", balance);
```

```
}
```

```
void interest() {
```

```
balance += balance * (rate) / 100;
```

```
System.out.println("Balance:" + balance);
```

```
}
```

```
}
```

```
class curAcct extends account {
```

```
    private double minBal = 500;
```

```
    private double serviceCharges = 50;
```

```
    curAcct(String name, int accno, double balance) {
```

```
        super(name, accno, "Current", balance);
```

```
}
```

```
    void checkmin() {
```

```
        if (balance < minBal) {
```

```
            System.out.println("Balance is less than min balance,service charges imposed:" +  
serviceCharges);
```

```
            balance -= serviceCharges;
```

```
            System.out.println("Balance is:" + balance);
```

```
}
```

```
}
```

```
}
```

```
class Bank {
```

```
    public static void main(String a[]) {
```

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

```
        System.out.println("Enter the name,type(current/savings),account number,initial  
balance:");
```

```
String name = s.next();
String type = s.next();
int accno = s.nextInt();
double balance = s.nextDouble();
int ch;
double amount1, amount2;
account acc = new account(name, accno, type, balance);
savAcct sa = new savAcct(name, accno, balance);
curAcct ca = new curAcct(name, accno, balance);
while (true) {
    if (acc.type.equals("savings")) {
        System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute interest 4.display");
        System.out.println("Enter the choice:");
        ch = s.nextInt();
        switch (ch) {
            case 1:
                System.out.println("Enter the amount:");
                amount1 = s.nextInt();
                sa.deposit(amount1);
                break;
            case 2:
                System.out.println("Enter the amount:");
                amount2 = s.nextInt();
                sa.withdraw(amount2);
                break;
            case 3:
                sa.interest();
                break;
            case 4:
```

```
    sa.display();
    break;

case 5:
    System.exit(0);

default:
    System.out.println("invalid input");
    break;
}

} else {
    System.out.println("\nMenu\n1.deposit 2.withdraw 3.display");
    System.out.println("Enter the choice:");
    ch = s.nextInt();

    switch (ch) {

case 1:
    System.out.println("Enter the amount:");
    amount1 = s.nextInt();
    ca.deposit(amount1);
    break;

case 2:
    System.out.println("Enter the amount:");
    amount2 = s.nextInt();
    ca.withdraw(amount2);
    ca.checkmin();
    break;

case 3:
    ca.display();
    break;

case 4:
```

```
        System.exit(0);

    default:
        System.out.println("Invalid input");
        break;
    }

}

}

}
```

GENERICSS

```
class GenericStack<T> {  
    private Object[] stackArray;  
    private int top = -1;  
    private static final int MAX_SIZE = 5;  
  
    public GenericStack() {  
        stackArray = new Object[MAX_SIZE];  
    }  
  
    public void push(T value) {  
        if (top < MAX_SIZE - 1) stackArray[++top] = value;  
        else System.out.println("Stack is full. Cannot push more elements.");  
    }  
  
    @SuppressWarnings("unchecked")  
    public T pop() {  
        if (top >= 0)  
            return (T) stackArray[top--];  
        else {  
            System.out.println("Stack is empty. Cannot pop more elements.");  
            return null;  
        }  
    }  
  
    public boolean isEmpty() {  
        return top == -1;  
    }
```

```
public boolean isFull() {
    return top == MAX_SIZE - 1;
}

}

class Main{
    public static void main(String[] args) {
        GenericStack<Integer> integerStack = new GenericStack<>();
        GenericStack<Double> doubleStack = new GenericStack<>();

        // Push integers to the integer stack
        for (int i = 1; i <= 5; i++) {
            integerStack.push(i);
        }

        // Push doubles to the double stack
        for (double i = 1.0; i <= 5.0; i++) {
            doubleStack.push(i);
        }

        // Pop and print integers from the integer stack
        System.out.println("Popped integers from the stack:");
        while (!integerStack.isEmpty()) {
            System.out.println(integerStack.pop());
        }

        // Pop and print doubles from the double stack
        System.out.println("Popped doubles from the stack:");
        while (!doubleStack.isEmpty()) {
            System.out.println(doubleStack.pop());
        }
    }
}
```

```
    }  
}  
}  
}
```

Abstract class prog

```
import java.lang.Math;
```

```
abstract class Shape {
```

```
    double a;  
    double b;  
    double c;
```

```
    abstract void calculateArea();
```

```
    abstract void calculatePerimeter();
```

```
}
```

```
class Triangle extends Shape {
```

```
    Triangle(double x, double y, double z) {  
        a = x;  
        b = y;  
        c = z;  
    }
```

```
    void calculateArea() {
```

```
        double s = (a + b + c) / 2;  
        System.out.println("Area=" + (Math.sqrt(s * (s - a) * (s - b) * (s - c))));  
    }
```

```
    void calculatePerimeter() {
```

```
        System.out.println("Perimeter=" + (a + b + c));
    }

}

class Circle extends Shape {
    Circle(double r) {
        a = r;
    }

    void calculateArea() {
        System.out.println("Area=" + (Math.PI * a * a));
    }

    void calculatePerimeter() {
        System.out.println("Perimeter=" + (2 * Math.PI * a));
    }
}

class ShapeM {
    public static void main(String[] args) {
        Triangle t = new Triangle(2.0, 3.0, 5.0);
        Circle c = new Circle(5.0);
        t.calculateArea();
        t.calculatePerimeter();
        c.calculateArea();
        c.calculatePerimeter();
    }
}
```

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;
```

```
import java.util.Scanner;
```

```
public class Internals extends Student {  
    protected int marks[] = new int[5];  
  
    public void inputCIEmarks() {  
        Scanner s = new Scanner(System.in);  
        System.out.println("Enter Internal Marks for " + name);  
        for (int i = 0; i < 5; i++) {  
            System.out.print("Subject " + (i + 1) + " marks: ");  
            marks[i] = s.nextInt();  
        }  
    }  
}
```

```
package CIE;
```

```
import java.util.Scanner;
```

```
public class Student {
```

```
protected String usn = new String();
protected String name = new String();
protected int sem;

public void inputStudentDetails() {
    Scanner s = new Scanner(System.in);
    System.out.print("Enter USN: ");
    usn = s.next();
    System.out.print("Enter Name: ");
    name = s.next();
    System.out.print("Enter Semester: ");
    sem = s.nextInt();
}

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

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];  
    finalMarks = new int[5];  
}  
  
public void inputSEEmarks() {  
    Scanner s = new Scanner(System.in);  
    System.out.println("Enter SEE Marks for " + name);  
    for (int i = 0; i < 5; i++) {  
        System.out.print("Subject " + (i + 1) + " marks: ");  
        marks[i] = s.nextInt();  
    }  
}  
  
public void calculateFinalMarks() {  
    for (int i = 0; i < 5; i++)  
        finalMarks[i] = marks[i] / 2 + super.marks[i];  
}  
  
public void displayFinalMarks() {  
    displayStudentDetails();  
    for (int i = 0; i < 5; i++)  
        System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);  
}  
}  
import SEE.Externals;  
  
public class Pkgmain {
```

```
public static void main(String args[]) {  
    int numOfStudents = 2;  
    Externals finalMarks[] = new Externals[numOfStudents];  
  
    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("Displaying data:\n");  
  
    for (int i = 0; i < numOfStudents; i++) {  
        finalMarks[i].calculateFinalMarks();  
        finalMarks[i].displayFinalMarks();  
    }  
}
```

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

class InputScanner
{
    protected Scanner s;
    public InputScanner()
    {
        s = new Scanner(System.in);
    }
}

class Father extends InputScanner
{
    protected int fatherAge;
    public Father() throws WrongAge
    {
        System.out.println("Enter Father's Age:");
    }
}
```

```
fatherAge=s.nextInt();

if(fatherAge<0)
{
    throw new WrongAge("Age cannot be negetive:");
}

}

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=s.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");
    }
}

public void display()
{
    super.display();
    System.out.println("Son's Age: " + sonAge);
}

}

public class FatherSonAge
{
    public static void main(String args[])
    {
        try
        {
            Son son=new Son();
            son.display();
        }

        catch (WrongAge e)
        {

```

```
        System.out.println("Error: " + e.getMessage());  
    }  
}  
  
}
```

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.

```
class DisplayMessageThread extends Thread {  
    private final String message;  
    private final long interval;  
  
    DisplayMessageThread(String message, long interval) {  
        this.message = message;  
        this.interval = interval;  
    }  
  
    public void run() {  
        try {  
            while (true) {  
                System.out.println(message);  
                Thread.sleep(interval);  
            }  
        } catch (InterruptedException e) {  
            System.out.println(Thread.currentThread().getName() + " interrupted.");  
        }  
    }  
  
}  
  
public class TwoThreadDemo {  
    public static void main(String[] args) {  
        DisplayMessageThread thread1 = new DisplayMessageThread("BMS College of  
Engineering", 10000);  
        DisplayMessageThread thread2 = new DisplayMessageThread("CSE", 2000);  
    }  
}
```

```
thread1.setName("Thread 1");
thread2.setName("Thread 2");

thread1.start();
thread2.start();

try {
    Thread.sleep(30000);
} catch (InterruptedException e) {
    System.out.println("Main thread interrupted.");
}

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

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

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 ArithmeticException. Display the exception in a message dialog box.

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

class SwingDemo {
    SwingDemo() {
        // create jframe container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

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

        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        // calc button
        JButton button = new JButton("Calculate");

        // labels
```

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

// add in order :)

jfrm.add(err); // to display error bois

jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

ActionListener l = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from a text field");
    }
};

ajtf.addActionListener(l);
bjtf.addActionListener(l);

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a / b;
        }
    }
});
```

```
alab.setText("\nA = " + a);
blab.setText("\nB = " + b);
anslab.setText("\nAns = " + ans);
} catch (NumberFormatException e) {
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("Enter Only Integers!");
} catch (ArithmaticException e) {
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("B should be NON zero!");
}
}

});

// display frame
jfrm.setVisible(true);

}

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

}

}

10a) Interprocess communication using consumer and producer

```
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;  
    }  
}
```

```
    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 < 6) {  
            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 < 6) {
        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.");
    }
}
```

10b) 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 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); // get lock on a in this thread.  
        System.out.println("Back in main thread");  
    }  
  
    public void run() {  
        b.bar(a); // get lock on b in other thread.  
        System.out.println("Back in other thread");  
    }  
  
    public static void main(String args[]) {  
        new Deadlock();  
    }  
}
```

ଶ୍ରୀ ଅଯୋଦ୍ଧା

CE

IBM 22CS264



NAME: Shreya S Rodegi STD.: III SEC.: C ROLL NO.: _____ SUB.: 001

Develop a Java program that prints all real solns to the quadric equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadric formula. If the $D = b^2 - 4ac$ is -ve, display a message stating that there are no real solns.

```

import java.util.Scanner;
class quadratic
{
    int a, b, c;
    double s1, s2, d;
    void getd()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter coeffc of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while(a==0)
        {
            System.out.println("Not a quad eq");
            System.out.println("Enter non-zero a");
            Scanner sc = new Scanner(System.in);
            a = sc.nextInt();
            d = b*b - 4*a*c;
        }
    }
}

```

if ($d == 0$)

$$\text{r1} = (-b) / (2 * a);$$

System.out.println ("Roots are real & equal.");

System.out.println ("R1=R2=" + r1);

{

else if ($d > 0$)

{

$$\text{r1} = ((-b) + (\text{Math.sqrt}(d))) / (\text{double})(\delta + a);$$

$$\text{r2} = ((-b) - (\text{Math.sqrt}(d))) / (\text{double})(\delta + a);$$

System.out.println ("Roots are real & distinct.");

System.out.println ("R1=" + r1 + " R2=" + r2);

{

else if ($d < 0$)

{

System.out.println ("Roots are imaginary");

$$\text{r1} = (-b) / ((2 * a));$$

$$\text{r2} = \text{Math.sqrt}(-d) / (\delta + a);$$

System.out.println ("R1=" + r1 + "+i" + r2);

System.out.println ("R2=" + r1 + "-i" + r2);

{

{

class quadraticmain

{

public static void main (String args[])

{

Quadratic q = new quadratic();

q.getd();

q. compute();

?

?

Output:

SHREYA S RUDAGI
1BM22CS069

Enter the coefficients of a, b, c

0

9

8

Not a quadratic equation

Enter a non zero value for a:

5

Roots are imaginary

Root 1 = 0.0 + i0.88

Root 2 = 0.0 - i0.88

Enter the coefficients of a, b, c

2

4

2

Roots are real and equal

Root 1 = Root 2 = +1

(16)

Sum
12/2/23

Enter the coefficients of a, b, c.

2

8

9

Roots are real and distinct

Root 1 = +0.2649

Root 2 = 3.438

19/10/23

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 .

$$\rightarrow \text{SGPA} = \frac{\sum (\text{course credits}) (\text{grade points})}{\sum (\text{course credits})}$$

(including F grade)

$$\text{CGPA} = \frac{\sum (\text{course credits}) (\text{grade points})}{\sum (\text{course credits})}$$

(excluding F grade)

```
import java.util.*;
```

```
class Subject
```

{

```
    int subjectmarks;
```

```
    int credits;
```

```
    int grade;
```

}

```
class Student
```

{

```
    Subject subject[];
```

```
    String name;
```

```
    String USN;
```

```
double GPA);  
Scanner s;  
Student()  
{  
    int i;  
    subject = new Subject[3];  
    for(i=0; i<3; i++)  
        subject[i] = new Subject();  
    s = new Scanner(System.in);  
}
```

~~void getStudentDetails()~~

```
{  
    System.out.println("Enter your name");  
    name = s.next();  
    System.out.println("Enter your USN");  
    USN = s.next();  
}
```

~~void getMarks()~~

```
{  
    int i;  
    for(i=0; i<3; i++)  
    {  
        System.out.println("Enter the marks of"  
                           + (i+1) + ":");  
        Subject[i].subjectmarks = s.nextInt();  
        System.out.println("Enter credits " + (i+1) + ":" );  
        Subject[i].grade = (Subject[i].subjectmarks / 10) + 1;  
    }  
}
```

```
if (subject[i].grade == 11)  
    subject[i].grade = 10;  
if (subject[i].grade <= 4)  
    subject[i].grade = 0;
```

{

void computeSGPA()

{

int effectiveScore = 0;

int totalCredits = 0;

int i;

for (i = 0; i < 8; i++)

{

effectiveScore += (subject[i].grade *

subject[i].credits);

totalCredits += subject[i].credits;

}

SGPA = (double) effectiveScore / (double) totalCredits

{

public class Main

{

public static void main (String args [])

Student s1 = new Student();

s1.getStudentDetails();

s1.getMarks();

s1.computeSGPA();

```

System.out.println("Name: " + s1.name);
System.out.println("USN: " + s1.USN);
System.out.println("SGPA: " + s1.SGPA);

```

3

3

Output:

Enter your name: Shreya.

Enter your USN: IBM22CS267

Enter marks of Subject 1: 83

Enter credits of subject 1: 4

Enter marks of subject 2: 50

Enter credits of subject 2: 5

Enter marks of subject 3: 60

Enter credits of subjects 3: 8

Name: shreya - SHREYA S RODA

USN: IBM22CS267 - IBM22CS267

SGPA: 5.623

10

10/18/23

{ Output 2 of print2 using }

System.out.println("Hello World");

"Hello" + "World" + "Hello World" = Hello

"A" + "B" + "C" + "D" + "E" = ABCDE

"A" + "B" + "C" + "D" + "E" = ABCDE

HelloWorld + "Hello" + "Hello World" = HelloWorldHelloHelloWorld

26/12/23

(3) Create a class Book which contains four members name, author, price, num_pages. Include a constructor to set the values of the members. Include methods to set 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.*;  
class Books {  
    String name;  
    String author;  
    int price;  
    int numpages;  
    Books(String name, String author, int price,  
          int 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 = "Price :" + this.price + "\n";  
    numpages = "No of pages " + this.numpages + "\n";  
    return name + author + price + numpages;  
}
```

```
class Books {  
    public static void main(String args[]) {  
        int n;  
        String name;  
        String author;  
        int price, numPages;  
        System.out.println("Enter no. of books:");  
        Scanner s = new Scanner(System.in);  
        n = s.nextInt();  
        Books b[];  
        b = new Books[n];  
        int i;  
        for (i=0; i<n; i++)  
    }  
}
```

```
Scanner sc = new Scanner (System.in);  
name = sc.next();  
author = sc.next();  
price = sc.nextInt();  
numPages = sc.nextInt();  
b[i] = new Books(name, author, price,  
                 numPages);
```

{

```
for (i=0; i<n; i++)  
{
```

```
    System.out.println(b[i].toString());  
}
```

{

Output:

Enter number of books: 9

Hello

Hi

83

39

Oh!

Om

33

63

Book name: Hello

Author name: Hi

Price: 83

Number of Pages: 39

Book name: Oh!

Author name: Om

Price: 33

Number of Pages: 63

26/12/23

Q4

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.Scanner;  
  
class InputScanner {  
    Scanner s;  
    s = new Scanner(System.in);  
    input Scanner()  
    {  
        s = new Scanner(System.in);  
    }  
}
```

```
abstract class Shape extends InputScanner {  
    double a;  
    double b;  
    abstract void getArea();  
    abstract void displayArea();  
}
```

class rectangle extends shape {

 public void getInput() {

 System.out.println("Enter the details");

 a = s.nextInt();

 b = s.nextInt();

}

 public void displayArea {

 System.out.println("Area is: " + a * b);

}

class triangle extends shape {

 public void getInput() {

 System.out.println("Enter details: ");

 a = s.nextInt();

 b = s.nextInt();

}

 public void DisplayArea {

 System.out.println("Area is: " + 0.5 * a * b);

}

class circle extends circle {

 public void getInput() {

 System.out.println("Enter details: ");

 a = s.nextInt();

}

 public void DisplayArea {

 System.out.println("Area is " + 3.14 * a * a);

}

}

public class MainShape{

 public static void main (String args[]){

 rectangle r = new rectangle();

 triangle t = new triangle();

 circle c = new circle();

 r. get input();

 r. display Area();

 t. get input();

 t. display Area();

 c. get input();

 c. display Area();

{

}

Output:

Enter details:

2

4

Area is 8

Enter details: 2

2

Area is 2

Enter details: 1

Area is 3.14

26/01/24

Shreya S Ridaei
IRM22CS263

classmate
ate

Date
Page

9/1/24 Lab-5

- Q) Develop a Java program to create a bank that maintains two kinds of accounts.

```
import java.util.*;
```

class account?

```
String customer_name;
```

int acc_no;

String types

double balance

account(string name, int acc, string type,
double balance) {

name: customer-name

acc-no: acc;

~~this type: types~~

~~this.balance = balance;~~

void deposit (double amount) {

balance + = amount; }

void withdraw(double amount){

if ((balance-amount)>0){

balance = amount; }
balance = amount;

else {

~~System.out.println ("Insufficient balance");~~

```
void display() {
```

~~System.out.println ("Name: " + customer_name);~~

"acc_no" + "acc_no" + "Type:" + type +

"Balance" + balance); }

1

```
class sav_acc extends account {  
    private static double rate = 8;  
    sav_acc (String name, int acc_no,  
             double balance) {  
        super (name, acc_no, "Savings", balance);  
    }
```

```
    void interest() {  
        balance += balance * (rate) / 100;  
        System.out.println ("balance" + balance);  
    }
```

```
class cur_acc extends account {  
    cur_acc (String name, int acc_no,  
             double balance) {  
        super (name, acc_no, "Current", balance);  
    }
```

```
    private double min_bal = 250;  
    private double service_charge = 100;  
    void checkmin() {  
        if (balance < min_bal) {  
            System.out.println ("Insufficient  
balance, service charges are added");  
            balance -= service_charge;  
            System.out.println ("Balance is:" + balance);  
        }  
    }
```

```
public class bank {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter name");
        String name = s.nextLine();
        System.out.println("Enter type");
        String type = s.nextLine();
        System.out.println("Enter acc. no");
        int acc_no = s.nextInt();
        System.out.println("Enter initial balance");
        double balana = s.nextDouble();
        int ch;
        double amt1, amt2;
        account acc = new account(name, acc_no,
                                   type, balana);
        sav_acc sa = new sav_acc(name, acc_no,
                                 balance);
        cur_acc ca = new cur_acc(name, acc_no,
                                 balance);
        while (true) {
            if (acc.type.equals("savings"))
                System.out.println("Menu: 1. Deposit 2.
WithDraw 3. Interest 4. Display");
                System.out.println("Enter the choice");
                ch = s.nextInt();
                switch (ch)
                {
```

```
case 1: System.out.println ("Enter amt:");
amt1 = s.nextInt();
sa.deposit(amt1);
break;
case 2: System.out.println ("Enter amt:");
amt2 = s.nextInt();
sa.withdraw(amt2);
break;
case 3: sa.interest();
break;
case 4: sa.display();
break;
default: System.out.println ("Invalid input");
break;
```

else {

```
System.out.println ("Menu 1. Deposit 2.
withdraw. 3. Display");
System.out.println ("Enter choice");
ch = s.nextInt();
switch(ch)
```

```
case 1: System.out.println ("Enter amt");
amt1 = s.nextInt();
sa.deposit(amt1);
break;
case 2: System.out.println ("Enter amt");
amt2 = s.nextInt();
sa.withdraw(amt2);
break;
```

case 3: ca.display();
break;

case 4:

default: System.out.println("Invalid Input");
System.exit(0);

}

}

}

O/p:

Enter name: Shreya

enter type: current

enter acc. number: 8288

Enter initial balance: 2000

Menu 1. deposit 2. withdraw 3. display

Enter choice: 1

Enter amount: 4000

Enter choice : 3

Name: shreya

type : current

acc. number: 8288

balance : 6000

SHREYA S RUDALI

18M21CS267

Lab-6

classmate

Date _____
Page _____

Write a Java program to create a generic class stack which hold 5 integers and 5 double values

```
import java.util.Scanner;  
class Stack<E> {  
    E stack[];  
    int top;  
    final int size = 10;  
    Stack() {  
        stack = (E[]) new Object[size];  
        top = -1;  
    }  
    void push(E item) {  
        if (top == size - 1) {  
            System.out.println("Stack is full");  
        }  
        else {  
            stack[++top] = item;  
        }  
    }  
    E pop() {  
        if (top < 0) {  
            System.out.println("Stack is empty");  
            return null;  
        }  
        else {  
            return stack[top--];  
        }  
    }  
}
```

```
public class generic {
    public static void main (String args[]){
        Stack<Integer> mystack1 = new Stack<Integer>();
        Stack<Double> mystack2 = new Stack<Double>();
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the elements");
        for (int i=0; i<5; i++){
            int n = s.nextInt();
            mystack1.push(n);
        }
        System.out.println ("Enter the elements");
        for (int i=0; i<5; i++){
            int m = s.nextDouble();
            mystack2.push(m);
        }

        System.out.println ("Elements of stack1");
        for (int i=0; i<5; i++){
            System.out.println (mystack1.pop());
        }

        System.out.println ("Elements of stack2");
        for (int i=0; i<5; i++){
            System.out.println (mystack2.pop());
        }

        s.close();
    }
}
```

Output :

Enter the elements into the integer stack

- 1 13 - entered from 1 to 13
- 2 12 - entered from 12 to 12
- 3 11 - entered from 11 to 11
- 4 10 - entered from 10 to 10
- 5

Enter the elements into double stack

- 5
- 6
- 7
- 4
- 3

Elements of stack 1 : 5

- 4
- 3
- 2

Elements of stack 2 :

- 3.0
- 4.0
- 7.0
- 6.0
- 5.0

SHREYA S PUDIGI

1RM22C1267

Write a java program to create an abstract class Shape with abstract methods calculateArea() & calculatePerimeter(). Create sub-class Circle & Triangle that extends the Shape class and implement the respective methods to calculate area and perimeter of each shape.

```
import java.lang.Math;  
abstract class Shape {  
    double a;  
    double b;  
    double c;  
    abstract void calculateArea();  
    abstract void calculatePerimeter();  
}
```

```
class Triangle extends Shape {  
    Triangle(double x, double y, double z)  
    {  
        a = x;  
        b = y;  
        c = z;  
    }  
    void calculateArea(){  
        double s = (a+b+c)/2;
```

```
        System.out.println("Area" +  
            (Math.sqrt(s * (s-a)*(s-b)*(s-c))));  
    }
```

void calculatePerimeter() {
System.out.println("Perimeter = " +
(a+b+c));
}

} class Circle extends Shape {
Circle(double r) {
a=r;
}

void calculateArea() {
System.out.println("Area = " + (Math.PI *
a*a));
}

void calculatePerimeter() {
System.out.println("Perimeter = " +
(2 * Math.PI * a));
}

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

Triangle t = new Triangle(2.0, 3.0, 5.0);

Circle c = new Circle(5.0);

t.calculateArea();

t.calculatePerimeter();

c.calculateArea();

c.calculatePerimeter();
}

}

O/P:

Area : 4.14578098794925

perimeter = 111.0

Area = 98.5398163

perimeter = 31.41598653.

Demonstrate string length, String distance,
String concat.

Public class String {

 public static void main (String args [])

 System.out.println (" Demonstrate str length");
 char char [] = {"a", "b", "c"};

 String s1 = new String (char);

 System.out.println (s1.length ());

 String. out.println ("String concat");

 String age = "19";

 String s2 = "He is" + age + "years old";

 System.out.println (s2);

 System.out.println ("demonstrate literals");

 System.out.println ("abc".length());

}

Q.P:

Demonstrating string length:

5

s1 sr. concat

He is 11 years old

Demonstrate literals. : 3

Use getchchar() to extract BMSCE from
"Welcome to BMSCE college".

public class String

public static void main (String args[])

{

String s = "Welcome to BMSCE college";

int start = 10;

int end = 16;

char buf [] = new char [end - start];

s.getChars (start, end, buf, 0);

System.out.println (buf);

3

3

Q.P:

BMSCE

SHREYA S RUPAIS

IBM2LC267

16/01/24

Lab 6 Strings

(1) BMSCE

BMSCE

(2) 3

3

Roll no 10 is present.

(3) Dimensions are 10.0 by 14.0 by 12.0

book b: Dimensions are 10.0 by 14.0 by 12.0

(4) bmsce

(5) 65

66

67

Welcome to bmsce college

(6) Bmse equals Bmse → true

Bmse equals college → false

Bmse equals Ignorance BMSCE → true

(7) Substring is matched

s1 = "Bmse college"

s2 = "Welcome to Bmse college of Engineering"

(8) true

false

⑨ - false
true

⑩ - Hello equals +Hello → true
+Hello == +Hello → false

⑪ The names in alphabetical order are
apple
ball
cat
lion
watch

⑫ Sorted Numbers (Ascending order) =
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

⑬ There was a test. This was too

⑭ hello world

⑮ commage

⑯ Hello friends

⑰ student 1

name: prateek

Reg no: 123

Semester: 3

CGPA: 8.87

student 2

name: Hareesh

Reg no: 121

semester: 4

cGPA : 9.85

⑧ chart 3 is 'A'

reverse of SAS is SAS

⑨ Eagle is flying

Eagle makes a sound

Hawk is moving

Hawk is making a sound

⑩ Circle-Area: 78.5398

Perimeter - 31.4259

Triangle Area: 5.0

Perimeter - 12.0

- Shreya S Rudagi

- IBM2215867

Lab-7.

23/11/24

- Q) Create a package for called maths having a class called number (add and substrac) method. Implement a simple class mathe demo. to use maths (outside package mathe) that makes use of class which is provided by mathe.
- Q) Create a package abc which has two classes Student and Internals. The class Student has members like osn, name, sem. The class Internals derived from student has an array that stores the internal marks scored in 5 courses of the current semester of the student. Create another package xyz which has the class. extranal which is derived class of student. This class has an array that stores the SEE marks scored in 5 courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

(import abc.*; import xyz.*;

(int osn; int name; int sem;

(int m1 + "marks1"); int m2 + "marks2";

(int m3 + "marks3"); int m4 + "marks4";

// 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() {

Scanner sc = new Scanner (System.in);

System.out.println ("Enter USN: ");

usn = sc.next();

System.out.println ("Enter Name: ");

name = sc.next();

System.out.println ("Enter semester: ");

sem = sc.nextInt();

}

public void DisplayStudentDetails() {

System.out.println ("USN: " + usn);

System.out.println ("Name: " + name);

System.out.println ("Semester: " + sem);

}

}

l1internals.java

package C1E;

import java.util.Scanner;

public class internals extends student {

protected int marks[] = new int[5];

public void input() {marks()}

Scanner sc = new Scanner(System.in);

System.out.println ("Enter Internals marks
for :" + name);

for (int i=0; i<5; i++)

{

System.out.println ("Subject " +(i+1) +
" marks : ");

marks[i] = sc.nextInt();

}

}

External.java

package SEE;

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

```
public class External extends internals {  
    protected int marks[];  
    protected int finalmarks[];
```

```
    public External() {  
        marks = new int[5];  
        finalmarks = new int[5];  
    }
```

```
    public void inputSeemarks() {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter SEE marks " + name);  
        for (int i = 0; i < 5; i++) {  
            System.out.println("Subject " + (i + 1) +  
                " marks: ");  
            marks[i] = sc.nextInt();  
        }  
    }
```

```
    public void calculateFinalMarks() {  
        for (int i = 0; i < 5; i++) {  
            finalmarks[i] = marks[i] / 2 + super.marks[i];  
        }  
    }
```

```
public void displayFinalMarks()
```

{

```
    displayStudentDetails();
```

```
    for (int i=0; i<5; i++)
```

{

```
        System.out.println ("Subject: " + (i+1)  
                           + ": " + finalMarks[i]);
```

}

}

}

```
11 finalmarks.java.
```

```
import SEE-external;
```

```
public class finalmarks
```

```
public static void main (String args[])
```

{

```
    int numofStudents = 2;
```

```
    external finalMarks [] = new
```

```
    external [numofStudents];
```

```
for (int i=0; i< numofStudents ; i++)
```

{

```
    finalMarks [i] = new external();
```

```
    finalMarks [i]. inputStudentDetails();
```

```
    System.out.println ("Enter CIE marks");
```

```
    finalMarks [i]. inputCIEmarks();
```

```
System.out.println("Enter 5C marks");  
finalmarks[i].input5Cmarks();  
}
```

```
System.out.println("Displaying data:\n");
```

```
for(int i=0; i< numofstudents ; i++)  
{
```

```
    finalmarks[i].calculateFinalMarks();  
    finalmarks[i].displayFinalMarks();
```

{
}
}

Output:

Enter USN: 12

Enter name: Shreya

Enter Sem: 3

Enter CIE marks

Enter Internal marks for Shreya

Subject 1 marks: 33

Subject 2 marks: 23

Subject 3 marks: 22

Subject 4 marks: 33

Subject 5 marks: 11

Enter 5C marks for Shreya

Subject 1 marks: 89

Subject 2 marks: 90

subject 3 marks: 99

subject 4 marks: 88

subject 5 marks: 89

Enter USN: 22

Enter Name: Praneeta

Enter sem: 3

Enter CIE marks for Praneeta:

Subject 1 marks: 34

Subject 2 marks: 22

Subject 3 marks: 34

Subject 4 marks: 45

Subject 5 marks: 22

Enter SEE marks for praneeta:

Subject 1 marks: 78

Subject 2 marks: 8

Subject 3 marks: 89

Subject 4 marks: 90

Subject 5 marks: 99

Displaying data:

USN: 19

Name: Shreya

Semester: 3

Subject 1: 87

Subject 2: 78

Subject 3: 71

Subject 4: 77

Subject 5: 55

USN: 82

Name: Poaneeta

Semester: 3

Subject 1: 73

Subject 2: 26

Subject 3: 78

Subject 4: 90

Subject 5: 71

23/10/14

SHREYA S RUDAGI

IBM11CS267

2N

2G

~~10/10/14~~~~8F~~~~8~~~~P8~~~~OP~~~~PP~~

Total weight

G1: 120

Speed 2 - 3 m/s

E = m/s

Lab-9:

20/1/24
 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 & throws the exception WrongAge() when input age < 0. In Son class, implement a constructor that takes both father and son's age and throws an exception if son's age is \geq father's age.



```
import java.util.Scanner;
```

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

```
class InputScanner {
    Scanner s = new Scanner (System. in);
}
```

```
class Father extends InputScanner {
    int FatherAge;
}
```

```
public Father () throws WrongAge {
    System.out.println ("Enter father's Age:");
    FatherAge = s.nextInt();
}
```

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

```
3  
public void display() {  
    System.out.println ("Father's age" +  
        fatherAge);  
}
```

3

```
class Son extends Father {  
    int sonAge;  
    public 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 greater than father's  
            age");  
        }  
    }
```

```
3  
else if (sonAge < 0) {  
    throw new WrongAge ("Age  
    cannot be negative");  
}
```

```
3  
public void display () {  
    super.display();  
}
```

```
System.out.println ("Son's age: " + sonAge);
```

{

{

```
public class Age {
```

```
    public static void main (String [] args)
```

{

```
        try {
```

```
            Son son = new Son ();
```

```
            son.display ();
```

{

```
        catch (WrongAge e) {
```

```
            System.out.println ("Error: " +  
                e.getMessage ());
```

{

{

{

Output 1:

Enter Father's age: 35

Enter son's age: 15

Father's age: 35

Son's age: 15

Output 2:

Enter father's age: 45

Enter son's age: 55

Error: Son's age cannot be greater than
father's age.

Shreya S Rupagni
30.01.24

SHREYA S RUPAGNI
IBM22CS262

6/2/24

Lab-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.

class DisplayMessageThread extends Thread

{

```
private final String message;
private final long interval;
```

```
DisplayMessageThread (String message, long interval) {
```

```
    this.message = message;
```

```
    this.interval = interval;
```

}

```
public void run() {
```

```
    try {
```

```
        while (true) {
```

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

```
            Thread.sleep(interval);
```

}

```
} catch (InterruptedException e) {
```

```
    System.out.println(Thread.currentThread().  
        getName() + ". interrupted.");
```

}

{

```
} // End of Main class
```

public class TwoThread Demo {

 public static void main (String [] args)

 {

 DisplayMessageThread thread1 = new
 DisplayMessageThread ("BMS College of
 Engineering", 10000);

 DisplayMessageThread thread2 = new
 DisplayMessageThread ("CS6",
 2000);

 thread1.setName ("Thread 1");
 thread2.setName ("Thread 2");

 thread1.start();
 thread2.start();

 try {

 Thread.sleep (30000);

 } catch (InterruptedException e) {

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

 }

// if you want to stop threads

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

 System.out.println ("Main thread
 existing");

}

}

Output :

BMS college of Engineering

CSE

CSG

CSE

CSE

CSG

BMS college of Engineering

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering

Main thread exiting

Thread Q interrupted

Thread I interrupted

RHRYA & RUDRAJ

IBM 22CS267

```
import java.io.File;
import java.io.IOException;
public class DemoFile
{
    public static void main (String [] args)
    {
        File
    }
}
```

Lab 10

- (a) Demonstrate interprocess communication.

class Q {

int n;

boolean valueSet = false;

synchronized int get()

{

while (!valueSet)

{

try

{

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

wait();

};

catch (InterruptedException e)

{

System.out.println ("Interrupted Exception

caught");

}

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

synchronized void put(int n)
{
    while (valueSet)
        try {
            System.out.println("Producer waiting\n");
            wait();
        }
        catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
    this.n = n;
    valueSet = true;
    System.out.println("Put:" + n);
    System.out.println("Intimate Consumer");
    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 < 10)

{

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 < 10)

{

int r=q.get();

```
System.out.println("consumed" + x);  
    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");
```

{

{

Output:

Press control-C to stop.

Put: 0

Intimate Consumer

Producer waiting

Got: 0

Intimate producer

Put: 1

Intimate consumer

Producer waiting

consumed: 0

Got: 1

Intimate producer.

consumed = 1

Put: 2

Intimate consumer.

Producer waiting

Got: 2

Intimate producer.

consumed = 2

Put: 3

Intimate consumer.

Producer waiting

Got: 3

Intimate producer.

consumed = 3

SHREYA SRIVASTAVA

IRM22C267

(b) Deadlock:

class A

{

 synchronized void foo(B b)

 {

 String name = Thread.currentThread().

 getname();

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

 try {

 Threads sleep(1000);

}

```
catch(InterruptedException 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(InterruptedException 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  
        ("Main Thread");
```

```
    Thread t = new Thread(this, "Racing  
        Thread");
```

```
    t.start();
```

```
    a.foo(b);
```

```
    System.out.println("Back in main  
        thread");
```

```
}
```

```
    b.bar(a);
```

```
    System.out.println("Back in other  
        thread");
```

```
}
```

```
public static void main (String args[])
```

{}

new deadlock();

{

}

O/P:

Main thread entered A.foo

Racing Thread entered B.bar

Main Thread trying to call B.last()

Inside A.last

Back in main thread

Racing Thread trying to call A.last()

Inside A.last

Back in other thread

SHREYA S RUPALI

13M22CS262

Shri
13.02.24

20/01/24 Lab-9

WAP that creates a user interface to perform integer divisions. The user enters two nos 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 & num2 are not an integer, the no would throw a numberformatexception. If num2 were zero, the program would throw an arithmetic exception display the exception in a message box.

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

```
class SwingDemo {
    swingDemo() {
```

```
        JFrame jfrm = new JFrame("Divide app");
        jfrm.setSize(875, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.
            EXIT_ON_CLOSE);
```

```
        JLabel jlab = new JLabel ("Enter divider and
            divident.");
```

~~```
 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)
```

```
        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(tf1.getText());
    int b = Integer.parseInt(tf2.getText());
    int ans = a/b;
```

```
    alab.setText("\nA = " + a);
```

```
    blab.setText("\nB = " + b);
```

```
    anslab.setText("\nans = " + ans);
```

{}

catch (NumberFormatException e)

{

```
    alab.setText(" ");
```

```
    blab.setText(" ");
```

```
    anslab.setText(" ");
```

```
    ex.setText("Enter only integers!");
```

{}

catch (ArithmaticException e)

{

```
    alab.setText(" ");
```

```
    blab.setText(" ");
```

```
    anslab.setText(" ");
```

```
    ex.setText("B should be NON zero!");
```

{}

{}

{};

```
jfrm.setVisible(true);
```

{}

```

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

```

Q1:

Enter the divider and dividend:

$$A=50 \quad B=5 \quad \text{Ans}=10$$

Enter the divider and dividend:

~~B should be non zero!~~

SHREYA & RUPALI

IBU22C8267

Functions used:

- `JFrame` → The `java.awt.swing.JFrame` class is a type of container which inherits the `java.awt.Frame` class. `JFrame` works like the main window where components like labels, textfields are added to create a GUI.
- `setSize(int width, int height)` → used to resize a frame using width and height parameters
- `setLayout()` - method allows you to set the layout of the container. The layout manager helps layout the components held by this container.
- `setDefaultCloseOperation()` - method is to specify one of several options for the close button.
`JFrame.EXIT_ON_CLOSE` - Exit the application.
- `JLabel` - The object of `JLabel` class is a component for placing text in a container. It is used to display a single line of read only text.
- `JTextField` - The object of a `JTextField` class is a text component that allows the

editing of a single line text. It inherits JTextComponent class.

- add(frame) - adds new frame in the existing frame
- ActionListener - The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent. This interface is found in java.awt.event package.
- setText() - This method substitutes new text for all or part of the text in the text field. This works only with the first line of multi-line text fields.
- ~~setVisible()~~ - is a method that has return type boolean.

By: 08A24
20.08.20