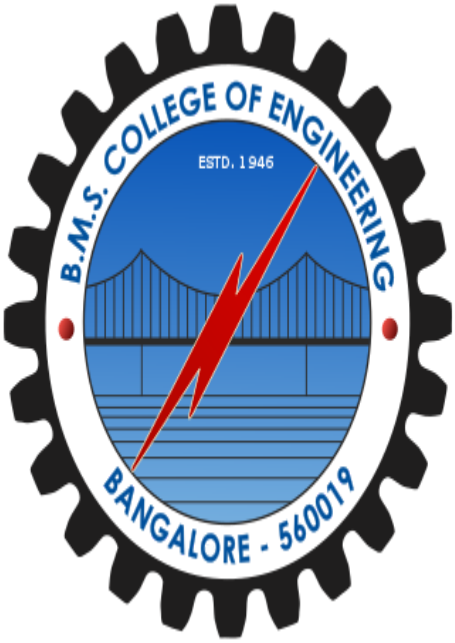
B.M.S. College of Engineering

(Autonomous Institution affiliated to VTU, Belagavi)

Department of Computer Science and Engineering



AAT

OOJ Lab
Record

(December 2023-March 2024)

Name: Sneha Prasanna
USN:1BM22CS284

Q1. Develop a Java program that prints all real solutions to the quadratic equation ax2 +bx+c=0. Read in a, b,
c and use the quadratic formula. If the discriminate 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(“Roo1 =”+ 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 + &quot; + i”+r2);
System.out.println(“Root1 = “+ r1 + &quot; - i”;+r2);
}

}
}

class QuadraticMain
{
public static void main(String args[])
{
Quadratic q = new Quadratic();
q.getd();



q.compute();
}
}
```

Q3. Create a class Book which contains four members:name,author,price,num_pages.Include a constructor to set the value for the members. Include methods to set and get the details of the objects. Include a string() 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;

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

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

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 b[] = 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();

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

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

Q6.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 {
    String usn;
    String name;
    int sem;

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

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

```
    // Constructor
    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}
// File: SEE package
package SEE;

public class External extends CIE.Student {
    int[] seeMarks;

    // Constructor
    public External(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
}
// File: Main program to declare final marks
import CIE.Internals;
import SEE.External;

public class FinalMarks {
    public static void main(String[] args) {
        // Example usage
        // Internal marks for a student in five courses
        int[] internalMarks1 = {80, 75, 90, 85, 88};
        Internals student1 = new Internals("1BM18CS001", "John Doe", 3, internalMarks1);

        // SEE marks for another student in five courses
        int[] seeMarks2 = {85, 78, 92, 89, 91};
        External student2 = new External("1BM18CS002", "Jane Doe", 3, seeMarks2);

        // Displaying information
        displayStudentDetails(student1);
        displayStudentDetails(student2);
    }

    // Method to display student details and marks
```

```
    public static void displayStudentDetails(Internals student) {
        System.out.println("USN: " + student.usn);
        System.out.println("Name: " + student.name);
        System.out.println("Semester: " + student.sem);
        System.out.println("Internal Marks:");
        for (int i = 0; i < student.internalMarks.length; i++) {
            System.out.println("Course " + (i + 1) + ": " + student.internalMarks[i]);
        }
        System.out.println("Total Internal Marks: " +
calculateTotalMarks(student.internalMarks));
        System.out.println("--------------------");
    }

    // Method to calculate total marks
    public static int calculateTotalMarks(int[] marks) {
        int total = 0;
        for (int mark : marks) {
            total += mark;
        }
        return total;
    }
}
```

Q5. Develop a Java program to create a class Bank that maintains two kinds of account for its
customers, one called savings account and the other current account. The savings account
provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service
charge is imposed.

• Create a class Account that stores customer name, account number and type of account.

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

• 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 Bank {
    public static void main(String[] args) {
        Sav_acc savingsAccount = new Sav_acc();
        savingsAccount.displayBalance();
        savingsAccount.compoundInterest();
        savingsAccount.withdraw();

        Curr_acct currentAccount = new Curr_acct();
        currentAccount.displayBalance();
        currentAccount.chequeBook();
    }
}

class Account {
    String cust_name;
    String acc_no;
    String acc_types;
    double min_balance;
    double service_charge;
    double balance;

    Scanner sc = new Scanner(System.in);

    Account() {
        System.out.println("Enter customer name:");
        cust_name = sc.nextLine();
        System.out.println("Enter account number:");
        acc_no = sc.nextLine();
        System.out.println("Enter the minimum balance:");
        min_balance = sc.nextDouble();
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of $" + amount + " successful.");
    }
```

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

    void withdraw(double amount) {
        if (balance - amount < min_balance) {
            System.out.println("Insufficient funds. Service charge of $" + service_charge + "
imposed.");
            balance -= service_charge;
        } else {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful.");
        }
    }
}

class Sav_acc extends Account {
    double interest_rate;

    Sav_acc() {
        System.out.println("Enter balance:");
        balance = sc.nextDouble();
        System.out.println("Enter interest rate:");
        interest_rate = sc.nextDouble();
    }

    void compoundInterest() {
        double interest = balance * interest_rate / 100;
        deposit(interest);
        System.out.println("Interest of $" + interest + " credited.");
    }

    void withdraw() {
        System.out.println("Enter withdrawal amount:");
        double amount = sc.nextDouble();
        super.withdraw(amount);
    }
}
```

```
class Curr_acct extends Account {
    Curr_acct() {
        System.out.println("Enter balance:");
        balance = sc.nextDouble();
        service_charge = 10.0; // Example service charge for current account
    }

    void chequeBook() {
        System.out.println("Cheque book issued successfully.");
    }
}
```

Q7.Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called
“Father” and derived class called “Son” which extends the base class. In Father class, implement a
constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class,
implement a constructor that cases both father and son’s age and throws an exception if son’s age is
>=father’s age.

```
import java.util.Scanner;

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

class Father {
    int fatherAge;

    Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.println("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 is " + fatherAge);
    }
}

class Son extends Father {
    int sonAge;

    Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the 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");
        }
    }

    void display() {
        super.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("Error: " + e.getMessage());
        }
    }
}
```

Q8. Write a program which creates two threads, one thread displaying “BMS College of Engineering” once

every ten seconds and another displaying “CSE” once
every two seconds.

```
class NewThread implements Runnable
{
Thread t;
NewThread()
{
t=new Thread(this, "NThread");
System.out.println("CT:"+t);
t.start();
}

public void run()
{
try
{
for(int n=5;n>0;n--)
{
System.out.println("CSE");
Thread.sleep(2000);
}
}
catch(InterruptedException ie) {
System.out.println("CSE thread interrupted");
 }

System.out.println("CSE thread quitting");
}
}
class Thread2
{
public static void main(String ss[])
{
new NewThread();
System.out.println("Back in main");
try
{
for(int n=5;n>0;n--)
{
```

```
System.out.println("BMS College of Engineering");
Thread.sleep(10000);
}
}
catch(InterruptedException ie)
{
System.out.println("BMS thread interrupted");
}
System.out.println("BMS thread quitting.");
System.out.println("Sneha Prasanna ------- 1BM22CS284");
}
}
```

Q9. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

```
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, 200);
      jfrm.setLayout(new FlowLayout());
      // to terminate on close
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

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

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

button.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent evt) {
      try {
         int a = Integer.parseInt(ajtf.getText());
         int b = Integer.parseInt(bjtf.getText());
         if (b == 0) {
            throw new ArithmeticException("B should be non-zero!");
         }
         int ans = a / b;

         alab.setText("\nDividend (A) = " + a);
         blab.setText("\nDivisor (B) = " + b);
         anslab.setText("\nResult = " + ans);
         err.setText("");
      } catch (NumberFormatException e) {
         err.setText("Enter Only Integers!");
         alab.setText("");
         blab.setText("");
         anslab.setText("");
      } catch (ArithmeticException e) {
         err.setText("B should be non-zero!");
```

```
                alab.setText("");
                blab.setText("");
                anslab.setText("");
            }
        }
    });

    // 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();
      }
    });
  }
}
```

Q10.a) Demonstrate Deadlock

```
class A {

synchronized void foo(B b) {

String name = Thread.currentThread().getName();
System.out.println("Sneha Prasanna 1BM22CS284");
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); // 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();

}

}
```

Q10.b)Demonstrate IPC

Demostrate Inter Process Communication.

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

    synchronized int get() {
        while (!valueSet) {
            try {
                System.out.println("\nConsumer waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("\nIntimate Producer\n");
        notify();
        return n;
    }

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

class Producer implements Runnable {
    Q q;
```

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

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

class Consumer implements Runnable {
    Q q;

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

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

public 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.");
    }
```

}

Q4. Develop a Java Program to create an abstract class nammed Shape that contains two integers and an empty method nammed printArea(). Provide three classes nammed 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;

abstract class Shape {
   double a;
   double b;

   abstract void getInput();
   abstract void printArea();
}

class Rectangle extends Shape {
   Scanner s = new Scanner(System.in);

   void getInput() {
      System.out.println("Enter length and breadth: ");
      a = s.nextDouble();
      b = s.nextDouble();
   }

   void printArea() {
      System.out.println("Area of Rectangle: " + (a * b));
   }
}

class Triangle extends Shape {
   Scanner s = new Scanner(System.in);

   void getInput() {
      System.out.println("Enter base and height: ");
      a = s.nextDouble();
      b = s.nextDouble();
   }
```

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

class Circle extends Shape {
    Scanner s = new Scanner(System.in);

    void getInput() {
        System.out.println("Enter radius: ");
        a = s.nextDouble();
    }

    void printArea() {
        System.out.println("Area of Circle: " + (3.14 * a * a));
    }
}

public class Main {
    public static void main(String[] args) {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();

        r.getInput();
        r.printArea();

        t.getInput();
        t.printArea();

        c.getInput();
        c.printArea();
    }
}
```

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

```
Import java.util.Scanner;
class Subject{
Int subjectMarks;
Int credits;
Int grade;
}
Class Student{
Subject subject[];
String name;
String usn;
Double SGPA;
Scanner s;
Student(){
Int i;
subject=new Subject[9];
for(i=0;i<9;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(){
for(int i=0;i<9;i++)
{
System.out.println(“Enter marks for subjects”+(i+1)+”:”);
subject[i].subjectMarks=s.nextInt();
System.out.println(“Enter credits for subjects”+(i+1)+”:”);
subject[i].credits=s.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 compute_SGPA()
{
int effectiveScore=0;
Int totalCredits=0;
for(int i=0;i<9;i++)
{
effectiveScore+=(subject[i].grade+subject[i].marks);
totalCredits+=subject[i].credits;
}
SGPA=(Double)effectiveScore/(double)totalCredits;
}
}
class Main
{ public static void main(String args[]){
Student s1=new Student();
s1.getStudentDetail();
s1.getMarks();
s1.computeSGPA();
System.out.println(“Name:”+s1.name);
System.out.println(“USN:”+s1.usn);
System.out.println(“SGPA:”+s1.SGPA);
}
}
```

Q6. 1. Demonstrate various string constructor with various java programs.

Using String Literal:

java

Copy code

```
// Using String literal
String str1 = "Hello, World!";
System.out.println("String 1: " + str1);
```

Using new keyword and a character array:

java

Copy code

```
// Using new keyword and character array
char[] charArray = {'H', 'e', 'l', 'l', 'o'};
String str2 = new String(charArray);
```

System.out.println("String 2: " + str2);
Using new keyword and byte array with a specific character encoding (e.g., UTF-8):

java
Copy code
// Using new keyword and byte array with character encoding
byte[] byteArray = {72, 101, 108, 108, 111};
String str3 = new String(byteArray, java.nio.charset.StandardCharsets.UTF_8);
System.out.println("String 3: " + str3);
Using StringBuffer or StringBuilder:

java
Copy code
// Using StringBuffer
StringBuffer stringBuffer = new StringBuffer("Hello");
stringBuffer.append(", Java!");
String str4 = stringBuffer.toString();
System.out.println("String 4: " + str4);

// Using StringBuilder
StringBuilder stringBuilder = new StringBuilder("Hello");
stringBuilder.append(", Java!");
String str5 = stringBuilder.toString();
System.out.println("String 5: " + str5);
Using String constructor with another String object:

java
Copy code
// Using String constructor with another String object
String originalStr = "Original String";
String str6 = new String(originalStr);
System.out.println("String 6: " + str6);
Write a Java program to create an abstract class Bird with abstract methods fly() and makeSound(). Create subclasses Eagle and Hawk that extend the Bird class and implement the
respective methods to describe how each bird flies and makes a sound.
// Abstract class Bird
abstract class Bird {
    // Abstract methods

```
    public abstract void fly();
    public abstract void makeSound();
}

// Subclass Eagle
class Eagle extends Bird {
    // Implementation of fly method for Eagle
    @Override
    public void fly() {
        System.out.println("Eagle soars high in the sky.");
    }

    // Implementation of makeSound method for Eagle
    @Override
    public void makeSound() {
        System.out.println("Eagle screeches loudly.");
    }
}

// Subclass Hawk
class Hawk extends Bird {
    // Implementation of fly method for Hawk
    @Override
    public void fly() {
        System.out.println("Hawk glides gracefully in the air.");
    }

    // Implementation of makeSound method for Hawk
    @Override
    public void makeSound() {
        System.out.println("Hawk emits a sharp cry.");
    }
}

// Main class to test the Bird subclasses
public class BirdTest {
    public static void main(String[] args) {
        // Create an instance of Eagle
        Eagle eagle = new Eagle();
```

```
        System.out.println("Eagle:");
        eagle.fly();
        eagle.makeSound();

        System.out.println();

        // Create an instance of Hawk
        Hawk hawk = new Hawk();
        System.out.println("Hawk:");
        hawk.fly();
        hawk.makeSound();
    }
}
```

Generics.

```
import java.util.Stack;

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

        // Pushing 5 integers into the stack
        System.out.println("Pushing 5 integers into the stack:");
        for (int i = 1; i <= 5; i++) {
            intStack.push(i);
        }

        // Popping and displaying integers from the stack
        System.out.println("Popping and displaying integers from the stack:");
        while (!intStack.isEmpty()) {
            System.out.println(intStack.pop());
        }

        // Pushing 5 doubles into the stack
        System.out.println("Pushing 5 doubles into the stack:");
        for (double i = 1.1; i <= 5.5; i += 1.1) {
            doubleStack.push(i);
```

```
}

// Popping and displaying doubles from the stack
System.out.println("Popping and displaying doubles from the stack:");
while (!doubleStack.isEmpty()) {
    System.out.println(doubleStack.pop());
}}}
```

Date:

Name : Sneha Prasanna

Class : 3E

USN → 1BM22CS284

Java Record

Index Page.

Sl.No	Title	Date
01	Quadratic Equation	12/12/23
02	SGPA of Student	19/12/23
03	Book - class object Program	26/12/23
04	Abstract class - Shape	2/1/24
05	Bank Account	9/1/24
06	Strings and abstract class	16/1/24
07	Package - CIE, SEE	23/01/2024
08	Exception Handling, - Father and Son, Threads	30/1/2024
09	Applets	28/02/24
10	IPC, Deadlock	13/02/2024

Program → 1) Develop a java program that prints all real solutions upload to the quadratic equations $ax^2+bx+c=0$. 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.

12/12/23

Lab Program 1 :- 12-12-23

1) Quadratic formula

```
import java.util.Scanner;
class Quadratic{
    int a,b,c;
    double r1, r2, d;
    void getdata(){

    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 nonzero 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();
    }
}
```

Sneha Prabha
1BM22CS284

Output

① Enter the coefficients of a, b, c
1 2 3
Roots are imaginary
Root1 = -1.0 + i 1.4142135623730951
Root1 = -1.0 + -i 1.4142135623730951

② Enter the coefficients of a, b, c
1 -4 4
Roots are real and equal
Root1 = Root2 = 2.0

③ Enter the coefficients of a, b, c
5 6 1
Roots are real and distinct
Root1 = -0.2 Root2 = -1.0

19-12-23

papergrid

Date: / /

19/12/23

[Lab Program - 2]

Main Program

a) 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.

$$SGPA = \frac{\sum[(\text{Course Credits})(\text{Grade Points})]}{\sum[[\text{Course Credits}]]}$$

considering all courses registered in that semester (including those with F grade).

Cumulative grade point Average → [CGPA]

Code

⇒

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

class Student
{
    Subject subject[];
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Student()
    {
        int i;
        subject = new Subject[9];
```

LAB Program

```
for(i=0; i<9; i++)
        subject[i] = new Subject();
    s=new Scanner(System.in);
}

void getStudentDetails()
{
    System.out.print("Enter your Name:");
    name = s.next();
    System.out.print("Enter your USN:");
    usn = s.next();
}

void getMarks(){
        for(int i=0; i<9; i++)
    {
        System.out.print("Enter marks for subject " + (i+1) + ":");
        subject[i].subjectMarks = s.nextInt();
        System.out.print("Enter credits for subject " + (i+1) + ":");
        subject[i].credits = s.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<9;(i++)
      {
       effectiveScore+=(subject[i].grade*subject[i].marks);
       totalCredits+=subject[i].credits;
      }
     SGPA=(Double)effectiveScore/(double)totalCredits;
    }
   }
   class main
   {
    public static void main(String args[])
    {
     Student S1=new Student;
      S1.getStudentDetail();
     S1.getMarks();
     S1.computeSGPA();
    System.out.println("Name:"+S1.name);
   System.out.println("USN:"+S1.usn);
  System.out.println("SGPA"+S1.SGPA);
     }
    }
```

Output

Enter your name sneha
Enter your USN:
1BM22CS284
Enter marks for subject1:
93

Enter your credits for subject1: 07

Enter marks for subject2: 47

enter your credits for subject2: 6

Enter marks for subject3: 67

enter your credits for subject3: 8

enter marks for subject4: 78

enter your credits for subject4: 5

enter marks for subject5: 56

Enter your credits for subject5: 07

Enter marks for subject6: 77

enter your credits for subject6: 9

Enter marks for subject 7: 89

enter your credits for subject7: 89

Enter marks for subject8: 78

enter your credits for subject8: 08

Name: Sneha

USN: 1BM22CS284

SGPA: 7.631578947368421

Lab Program 3

Tuesday

26/12/25

1. Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the value for the members. Include methods to set and get the details of the objects. Include a string() 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;
  num = "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;
}
// end of class

class Main
{
public static void main(String args[])
{
Scanner s = new Scanner(System.in);
int n;
String name;
String author;
int price;
int numPages;
System.out.println("Enter no. of books");
n = s.nextInt();
Books b[];
b = new Books[n];
for (int i=0; i<n; i++)
{
System.out.println("Enter details of Book " + (i+1) + ":");
System.out.println("Enter name of book:");
name = Scan.nextLine();
System.out.println("Enter author name:");
author = Scan.nextLine();
System.out.println("Enter price of book:");
price = Scan.nextInt();
System.out.println("Enter no. of pages:");
numPages = Scan.nextInt();
b[i] = new Books(name, author, price, numPages);
}

```
System.out.println("Book details:");
System.out.println("Book Name \t Author \t Price \t No of pages");
for(int i=0; i<n; i++)
    {
        System.out.println(b[i].name + "\t" + b[i].author +
                "\t" + b[i].price + "\t" +
                b[i].numPages);
    }
}
System.out.println("Name --- Sneha");
System.out.println("1BM22CS284");
}
```

Output

```
Enter no. of books
3
Enter details of Book1
Enter name of book
Once
Enter author name: John Brown
Enter price of book
320
Enter no. of pages
415
Enter details of Book2:
Enter name of book
New
Enter author name:
Vivek
Enter price of book
300
Enter no. of pages.
210
```

Enter details of Book 3:
Enter name of book
start
Enter author name:
carol
Enter price of book
200
Enter no of pages
213
Book Details:

Book Name	Author	Price	No of pages
~~one~~	again	810	415
New	Vivek	200	210
Start	carol	200	315

Name - Sneha
USN 1BM22CS284

URA 2-1-24

2/01/24 Tuesday

Lab Program-4

1) 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.

Steps
1. Add class InputScanner.
2. Add class Shape extends InputScanner
3. Add class Rectangle extends Shape
4. Add class Triangle extends Shape
5. Add class Circle extends Shape.

Sol:-

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

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

~~class Shape extend Input Scanner~~
~~{~~
~~}~~

```
class Rectangle extends Shape {
@Override
void getInput() {
    Scanner scanner = new Scanner (System.in);
    System.out.println("Enter length of rectangle:");
    a = scanner.nextDouble();
    System.out.println("Enter width of rectangle:");
    b = scanner.nextDouble();
}
@Override
void displayArea(){
  System.out.println("Area of Rectangle:" + (a*b));
}
}
class Triangle extends Shape{
@Override
void getInput(){
Scanner scanner = new Scanner(System.in);
System.out.println("Enter base of ~~rectangle~~ a triangle:");
a = scanner.nextDouble();
System.out.println("Enter height of triangle:");
b = scanner.nextDouble();
}
@Override
void displayArea(){
   System.out.println("Area of Triangle:"+(0.5 *a *b));
}
}
```

papergrid

Date: / /

```
class Circle extends Shape{
    @Override
    void getInput(){
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter radius of circle:");
    a = Scanner.nextDouble();
    // b is not needed for circle, but we kept it for consistency
       with the Shape class
    b = 0;
}
    @Override
    void displayArea(){
        System.out.println("Area of circle:" + (3.14 * a * a));
    }
}

public class main {
    public static void main (String[] args){
        Rectangle rectangle = new Rectangle();
        rectangle.getInput();
        rectangle.displayArea();

        Triangle triangle = new Triangle();
        triangle.getInput();
        triangle.displayArea();
        Circle class
        Circle circle = new Circle();
        circle.getInput();
        circle.displayArea();
    }
}
System.out.println("Name - Sneha");
System.out.println("USN - 1BM22CS284");
}
```

Output

Enter length of rectangle:
5
Enter width of rectangle:
3
Area of Rectangle: 15.0
Enter base of triangle:
6
Enter height of triangle:
2
Area of Triangle: 6.0
Enter radius of circle:
4
Area of circle: 50.265482457436691

Name -- Sneha

USN -- 1BM22CS289

9/01/2024 Tuesday

Lab Program - 5

1) Develop a Java Program ~~that~~ to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other called current account. The saving account provides compound interest and withdrawal facility 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 ~~for~~ penalty if necessary and update the balance.

```
import java.util.Scanner;
class Bank {
String cust_name;
String acc_no;
String acc_types;
Double min_balance;
Double Service_charge;
public static void main (String[] args) {
    Sav_acc SavingsAccount = new Sav_acc();
    SavingsAccount.displayBalance();
    SavingsAccount.CompoundInterest();
    SavingsAccount.Withdraw();
```

→

```
Curr_acct currentAccount = new Curr_acct();
currentAccount.displayBalance();
currentAccount.depositBalance();
```

```
class Account {
String cust_name;
String acc_no;
String acc_types;
double min_balance;
double Service_charge;
Scanner sc = new Scanner (System.in);
Account() {
System.out.println("Enter customer name\n");
    cust_name = sc.nextline();
System.out.println("Enter account number:");
    acc_no = sc.nextline();
System.out.println("Enter the minimum balance\n");
    min_balance = sc.nextDouble();
}

void deposit(double amount) {
    balance += amount;
    System.out.println("Deposit of $" + amount + " Successful.");
}
```

```
void displayBalance(){
    System.out.println("Balance: $" + balance);
}
void withdraw(double amount){
    if (balance - amount < min_balance){
        System.out.println("Insufficient funds. Service charge of $"
                + service_charge + "imposed.");
        balance -= service_charge;
    } else{
        balance -= amount;
    }
    System.out.println("Withdrawal of $" + amount +
                "successful.");
}
}
class SavAcc extends Account{
    double interest_rate;
    System.out.println("Enter balance:");
    balance = sc.nextDouble();
    interest rate =
    System.out.println("Enter Interest Rate:");
    interest_rate = sc.nextDouble();
}
void CompoundInterest(){
    double interest = balance * interest_rate / 100;
    deposit(interest);
    System.out.println("Interest of $" + interest + "credited.");
}
```

```
void withdraw(){
    System.out.println("Enter withdrawal amount:");
    double amount = sc.nextDouble();
    super.withdraw(amount);
    }
}
class Curr_acct extends Account{
    Curr_acct(){
        System.out.println("Enter balance:");
        balance = sc.nextDouble();
        service_charge = 10.0; // Example service charge
                               for current account
    }
    void chequeBook(){
        System.out.println("Cheque book issued successfully");
    } System.out.println("Name -- Sneha");
    System.out.println("USN -- 1BM22CS284");
    }
```

Output

Enter customer name:
Sneha
Enter account number:
123ab
Enter the minimum balance:
350000
Enter balance:
3600000
Enter interest rate:
0.1
Balance : $3600000.0
Deposit of $3600.0 successful.
Interest of $3600.0 credited.

9/11/24

Enter withdrawal amount:
2000
Withdrawal of $2000.0 Successful
Enter customer name:
raj
Enter account number:
12299
Enter the minimum balance:
20000
Enter balance:
7800
Balance: $7800.0
Cheque Book Issued Successfully:
Name - Sneha
USN - 1BM22CS284

Lab 6

16/1/24

1) Demonstrate various string constructor with proper java program

```
import java.util.Scanner;
// Construct string of subset of char array.
class SubStringCons {
public static void main(String args[]) {
byte ascii[] = {65, 66, 67, 68, 69, 70};
String s1 = new String(ascii);
System.out.println(s1);
String s2 = new String(ascii, 2, 3);
System.out.println(s2);
    }
}
```

Output

ABCDEF

CDE

2) Demonstrate toString().

```
import java.util.Scanner;
// override toString() for Box class.
class Box {
    double width;
    double height;
    double depth;
    Box(double w, double h, double d) {
        width = w;
        height = h;
        depth = d;
    }
```

```
public String toString(){
    return "Dimensions are " + width + " by " +
        depth + " by " + height + ".";
}
}
class toStringDemo{
public static void main(String args[]){
Box b = new Box(10, 12, 14);
String s = "Box b: " + b; // Concatenate box object.
sbi
System.out.println(b); // Convert Box to string.
System.out.println(s);
}
}
```

Output

Dimensions are 10.0 by 14.0 by 12.0
Box b: Dimensions are 10.0 by 14.0 by 12.0.

20) Write a Java program using Generics. Show the stack class for ~~5 integers and 5 double.~~ 5 integers & 5 double integer and double type parameters.

Ans:-

```
public class Stack<E>{
    E stck[];
    int top;
    final int SIZE = 10;
    Stack()
    {
    stck = (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 underflow");
  return null;
  }
else
return stack[top--];
}
}
```

21) ~~Write a Java program using Generics. Show the~~

```
import java.util.Scanner;
public class TestStack {
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 elements into the integer
                      stack.");
  for (int i=0 ; i<5; i++)
  int n = s.nextInt();
  mystack1.push(n);
  }
```

```
System.out.println("Enter elements into the Double stack");
for (int i = 0; i<5; i++)
    {
        Double m = s.nextDouble();
        mystack2.push(m);
    }
System.out.println("Elements of stack 1");
for (int i=0; i<5; i++)
System.out.println(mystack1.pop());
System.out.println("Elements of stack 2");
for (int i=0; i<5; i++)
System.out.println(mystack2.pop());
s.close();
    }
}
```

Output:

Enter elements into the Integer stack
1
2
3
4
5
Enter elements into the Double stack
1.1
2.2
3.3
4.4
5.5

Elements of stack 1
5
4
3
2
1

Elements of stack 2
5.5
4.4
3.3
2.2
1.1

19) Write a Java program to create an abstract class Bird with abstract methods fly() and makeSound(). Create subclasses Eagle and Hawk that extend the Bird class and implement the respective methods to describe how each bird flies & makes sound.

Ans:-

```
// Abstract class Bird
abstract class Bird {
// Abstract methods
    public abstract void fly();
    public abstract void makeSound();
}
//Subclass Eagle
class Eagle extends Bird {
    // Implementation of fly method for Eagle
    @Override
    public void fly() {
        System.out.println("Eagle soars high in the sky.");
    }
// Implementation of makeSound method for Eagle
    @Override
    public void makeSound() {
        System.out.println("Eagle screeches loudly.");
    }
}
// Subclass Hawk
class Hawk extends Bird {
    // Implementation of fly method for Hawk
    @Override
    public void fly() {
```

```
System.out.println("Hawk glides gracefully in the air.");
}
// Implementation of makeSound method for Hawk
@Override
public void makeSound() {
    System.out.println("Hawk emits a sharp cry.");
}
}
// Main class to test the Bird subclasses
public class BirdTest {
    public static void main(String[] args) {
        Eagle eagle = new Eagle();
        System.out.println("Eagle:");
        eagle.fly();
        eagle.makeSound();
        System.out.println();
        // Create an instance of Hawk
        Hawk hawk = new Hawk();
        System.out.println("Hawk:");
        hawk.fly();
        hawk.makeSound();
    }
}
```

16/1/24

Output

```
Eagle:
Eagle soars high in the sky.
Eagle screeches loudly.

Hawk:
Hawk glides gracefully in the air.
Hawk emits a sharp cry.
```

29/01/24

Lab program - 7

1) Create a package CIE which has two classes - Student and Internals. The class Student has members like name, usn, 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 stored 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 Student{
protected String usn = new String();
protected String name = new String();
protected int sem;
public void inputStudentDetails(){
Scanner sc = new Scanner(System.in);
System.out.println("Enter the usn");
usn = sc.next();
System.out.println("Enter the name");
name = sc.next();
System.out.println("Enter the semester");
sem = sc.nextInt(); }
public void displayStudentDetails(){
System.out.println("usn -- "+ usn);
System.out.println("Name -- "+ name);
System.out.println("Semester -- "+sem);
}
```

```
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);
    for(int i = 0; i < 5; i++)
    {
      System.out.println("enter the marks of subject");
      marks = s.next();
    }

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);
for(int i=0; i<5; i++){
          System.out.println("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 ;
class Main{
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();
  }// end of for loop
```

} // end of public main
} end of class main

Output

Enter the usn
1AS123
Enter the name
John Doe
Enter the semester
3
Enter CIE marks
Enter the marks of subject 1 : 80
Enter the marks of subject 2: 75
Enter the marks of subject 3: 90
Enter the marks of subject 4: 85
Enter the marks of subject 5: 88
Enter SEE marks
Subject 1 marks: 85
Subject 2 marks: 82
Subject 3 marks: 78
Subject 4 marks: 90
Subject 5 marks: 88

Enter the usn
2XYZ456
Enter the name
Jane Doe
Enter the semester
3

Enter CIE marks
Enter the marks of subject1 : 75
Enter the marks of subject2 : 80
Enter the marks of subject-3: 92
Enter the marks of subject4 : 88
Enter the marks of subject-5 : 85
Enter SEE marks
Subject 1 marks: 88
Subject 2 marks: 90
Subject-3 marks : 75
Subject 4 marks: 82
Subject 5marks : 78
Displaying data:
USn - -1ABC123
Name- - John Doe
Semester - -3
Subject 1 : 102
Subject 2 : 108
Subject 3 : 120
Subject 4 : 121
Subject 5 : 108
USn - - 2XYZ456
Name - - Jane Doe
Semester - -3
Subject 1: 109
Subject 2: 116
Subject 3: 167
Subject 4: 170
Subject 5 : 163

30/11/24

30/1/24 — Lab Program-8 — Tuesday

Q4) 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 inputage<0. In Son class, implement a constructor that cases both father and son's age and throws an exception if ~~father age >= son~~ sons age >= fathers age.

Ans:

```
import java.util.Scanner;
class WrongAge extends Exception{
    WrongAge(String message){
        super(message);
    }
}
class Father{
    int fatherAge;
    Father() throws WrongAge{
        Scanner s = new Scanner(System.in);
        System.out.println("Enter fathers age");
        fatherAge = s.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{
  super();
Scanner s = new Scanner (System.in);
System.out.println("Enter the 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");
  }
 }
  void display(){
    super.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("Error:" + e.getMessage());
       }
  }
}
```

30/1/24

Output

Enter father's age
38
Enter the son's age
39
Error: Son's age cannot be greater than father's age.

30 | 1 | 24

6/02/24 Tuesday

Lab Program-8

1. Extract single thread program
2. Threads with constructors.
3. Multiple threads.
4. Lab Program-8.
5. Threads using Priority (create 1 thread with max priority, 2nd thread with min priority and 3rd thread with norm priority)
6. Synchronize method.
7. Lab Program 10. (Demonstrate IPC and Deadlock)

Lab Program-8

1) Write a Program to create two threads, one thread displaying "BNS college of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

→

```
class NewThread implements Runnable
{
    Thread t;
    NewThread;
    {
        t = new Thread(this, "Nthread");
        System.out.println("CT: " + t);
        t.start();
    }
    public void run()
    {
        try {
```

```
for(int n=5; n>0; n--)
{
    System.out.println("CSE");
    Thread.sleep(2000);
}
catch (InterruptedException (e) {
System.out.println("CSE thread interrupted");
}
System.out.println("CSE thread quitting");
}
}

Class Thread2
{
public static void main(String args[])
{
new Newthread();
System.out.println("Back in main");
try
{
for(int n=5; n>0; n--)
{
System.out.println("BMS College of Engineering");
Thread.sleep(10000);
}
}
catch(InterruptedException ie)
{
System.out.println("BMS thread interrupted");
}
System.out.println("BMS thread quitting.");
```

```
System.out.println("Sneha Prasanna ---- 1BM22CS284");
}
}
```

Output

CT: Thread[#30,Nthread,5,main]
Entering main
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE Thread quitting
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS thread quitting
Sneha Prasanna ---- 1BM22CS284

13/02/24
10.1) Q)

Lab Program - 10

Tuesday

Demonstrate ~~Interpo~~ Deadlock.

```
class A{
Synchronized void foo(B b){
 String name = Thread.currentThread().getName();
 System.out.println("Sneha Prasanna 1BM22CS284");
 System.out.println(name + " entered A.foo");
 try{
    Thread.sleep(10000);
 } 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

SnehaPrasanna 1BM22CS287

MainThread entered A.foo

RacingThread entered A.bar

MainThread trying to call A.last()

Inside A.last

Back in main thread

RacingThread trying to call A.last()

Inside A.last

Back in ~~other thread~~

10.a) Demonstrate IPC ~~Output~~

```
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 Intimate Producer\n");
        notify();
        return n;
    }
```

```
synchronized void put(int n){
    while (valueSet){
        try {
            System.out.println("\n 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("\n Intimate 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<4){
        q.put(i++);
    }
  }
}
```

Date: / /

```
public 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:

```
Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate Producer
Producer waiting
Consumed: 0
Got: 1
Intimate Producer
consumed: 1
Put: 2
Intimate consumer
Producer waiting
Got: 2
Intimate Producer
Consumed: 2
Put: 3
Intimate consumer
Producer waiting
Got: 3
Intimate Producer
Consumed: 3
```

Put: 4
Intimate Consumer
Got: 4
Intimate Producer
Consumed: 4

21/02/24 13.02.24

Lab Program-9

7) 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 and 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.

Ans

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

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

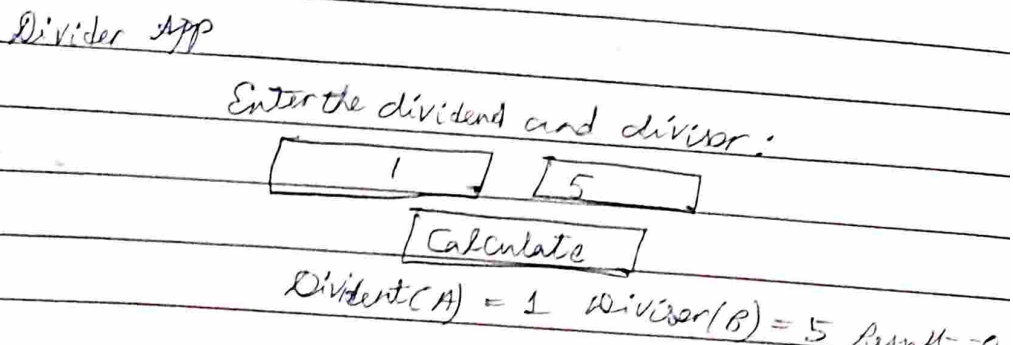
Date: / /

```
// text label
JLabel jlab = new JLabel("Enter the divident and divisor: ");
// 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(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(err);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

button.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            try {
                int a = Integer.parseInt(ajtf.getText());
                int b = Integer.parseInt(bjtf.getText());
                if (b == 0) {
                    throw new ArithmeticException
                    ("B should be non-zero!");
                }
```

```
int ans = a/b;
alab.setText("\n Dividend(A) = "+a);
blab.setText("\n Divisor(B) = "+b);
anslab.setText("\n Result = "+ans);
err.setText("");
}
catch (NumberFormatException e) {
    err.setText("Enter only Integers !");
    alab.setText("");
    blab.setText("");
    anslab.setText("");
}
catch(ArithmeticException e) {
    err.setText("B should be non-zero!");
    alab.setText("");
    blab.setText("");
    anslab.setText("");
}
});
// 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 SingleEvent();
    }
});
}
```

Date: / /

Output



10 functions used in the provided JavaSwing program.

1. JFrame() - This is a constructor for a creating a new JFrame which is a top-level container used to represent a Window in Swing applications.

2. setSize(int width, int height) - This function sets the size of the JFrame to the specified width and height in pixels.

3. setLayout(LayoutManager mgr) - This function sets the layout manager for the JFrame. In this case, it sets it to FlowLayout, which arranges components from left to right then top to bottom.

4. setDefaultCloseOperation(int operation) - This function sets the default close operation for the JFrame. In this case, it's set to 'EXIT_ON_CLOSE', which exits the application when the JFrame is closed.

5. JLabel(String text) = This is a constructor for creating a JButton with the specified text.

6. JTextField(int columns) - This is a constructor for creating a JTextField with the specified number of columns (width in characters).

7. JButton(String text) – This is constructor for creating a JButton with the specified text.

8. addActionListener(ActionListener listener) – This function adds an ActionListner to the JButton, so that an action (like clicking the button) can be detected and handled.

9. setText(String Text) – This function sets the text of a JLabel or a JTextField to the specified text

10. parseInt(String s) – This is a static method of a Integer class used to parse the string argument as a signed decimal integer. It throws a NumberformatException if the string dosn't have a parsable integer.

20/2/24