

INDEX

Name K. Keerthi Reddy

Standard 9 Section C Roll No. 100

Subject Java

SL No.	Date	Title	Page No.	Teacher Sign / Remarks
1.	30/9/24	Quadratic Equation	1	
2.	7/10/24	Calculation of SGPA	3	8
3.	14/10/24	Book details using toString()	5	1.10
4)	21/10/24	shape and areas (Abstract classes)	6	8 21.10
5.	28/10/24	Bank class using inheritance	8	8 28.10
6.	11/11/24	Marks card of student using package	13)
7.	28/11/24	Father & Son's age using Exception Handling	16)
8.	28/11	Display college name & department using threads,	17	8 02.12
10 a)	28/11	Demonstration of Interprocess Communication	18.21	
10 b)	28/11	Demonstration of Deadlock	19	
9.	28/11	Creation of Divider class	20	
				Completed

30/9/24

• Type promotion
Date: 30 Page: 1Bajna Gold
Date: 30 Page: 1LAB PROGRAMS

mybox 2 dyed

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

```
import java.util.Scanner;
import java.lang.Math;
```

```
class Quadratic {
```

```
    int a, b, c;
```

```
    double d1, d2, d;
```

```
    void get() {
```

```
        while (a == 0) {
```

```
            System.out.println("Not a quadratic equation");
```

```
            System.out.println("Enter a: ");
```

```
            a = scanner.nextInt();
```

```
}
```

```
    d = (b * b) - (4 * a * c);
```

```
    if (d == 0) {
```

```
        d1 = (-b) / (2.0 * a);
```

```
        System.out.println("Roots are real and equal,  
the two roots are: " + d1);
```

```
}
```

```
    else if (d > 0) {
```

```
        d1 = ((-b) + (Math.sqrt(d))) / (2.0 * a);
```

```
        d2 = ((-b) - (Math.sqrt(d))) / (2.0 * a);
```

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

realme GT NEO 3T

```

        System.out.println("Root 1 is: " + r1);
        System.out.println("Root 2 is: " + r2);
    }
    else if(d < 0){
        r1 = (-b) / (2.0 * a);
        r2 = Math.sqrt(-d) / (2.0 * a);
        System.out.println("Roots are imaginary");
        System.out.println("Root 1 is: " + r1 + " + " + r2 + "i");
        System.out.println("Root 2 is: " + r1 + " - " + r2 + "i");
    }
}
}

```

```

public class QuadraticSolve {
    public static void main(String args[]){
        Scanner scanner = new Scanner(System.in);
        Quadratic obj = new Quadratic();
        System.out.println("Enter a,b and c: ");
        obj.a = scanner.nextInt();
        obj.b = scanner.nextInt();
        obj.c = scanner.nextInt();
        obj.getd();
        System.out.println("Name: K. Keerthi Reddy");
        System.out.println("USN: 1BMR3CS137");
        scanner.close();
    }
}

```

Output: Enter a, b, and C:

0

1

2

The given equation is not a quadratic equation

realme GT NEO 3T

Enter a:

2

Roots are imaginary

Root 1 is: $-0.25 + 0.9682i$

Root 2 is: $-0.25 - 0.9682i$

Name : R.Karthi Reddy

USN : 1BM23CS137

DIPM

~~Q10~~
~~Q-ii~~

7/10/24

array of object

- Q) Develop a Java program to create a class student with members id, 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;  
    public void calculateGrade() {  
        if (subjectMarks < 40) {  
            grade = 0;  
        } else if (subjectMarks >= 90 && subjectMarks <= 100) {  
            grade = 10;  
        } else if (subjectMarks >= 80) {  
            grade = 9;  
        } else if (subjectMarks >= 70) {  
            grade = 8;  
        } else if (subjectMarks >= 60) {  
            grade = 7;  
        } else if (subjectMarks >= 50) {  
            grade = 6;  
        } else if (subjectMarks >= 40) {  
            grade = 5;  
        } else {  
            grade = 0;  
        }  
    }  
}
```

```
class Student {
    String name;
```

```
    String uen;
```

```
    double CGPA;
```

```
    Subject[] subjects; }
```

```
    Scanner s;
```

```
    Student() { }
```

```
        subjects = new Subject[8];
```

```
        for (int i=0; i<8; i++) { }
```

```
            subjects[i] = new Subject();
```

```
}
```

```
    S = new Scanner(System.in); }
```

```
public void getStudentDetails() { }
```

```
    System.out.print("Enter Student name: ");
```

```
    this.name = s.nextLine();
```

```
    System.out.print("Enter student uen: ");
```

```
    this.uen = s.nextLine(); }
```

```
public void getMarks() { }
```

```
    for (int i=0; i<8; i++) { }
```

```
        System.out.print("Enter marks for subject " +
```

```
        (i+1) + ": ");
```

```
        subjects[i].subjectMarks = s.nextInt();
```

```
        System.out.print("Enter credits for subject " +
```

```
        (i+1) + ": ");
```

```
        subjects[i].credit = s.nextInt();
```

```
        subjects[i].calculateGrade(); }
```

```
}
```

~~s.nextLine();~~

realme GT NEO 3T

```
public void computeSGPA() {
    double totalPoints = 0;
    int totalCredits = 0;
    for (int i=0; i<8; i++) {
        totalPoints += subjects[i].grade *
            subjects[i].credits;
        totalCredits += subjects[i].credits;
    }
    SGPA = (totalCredits == 0) ? 0 : totalPoints / totalCredits;
}
```

```
public void displayResults() {
    System.out.println("Student Name: " + this.name);
    System.out.println("USN: " + this.usn);
    System.out.printf("SGPA: %.2f %n", SGPA);
}
```

```
public class Student2 {
    public static void main (String [] args) {
        Student sl = new Student1();
        sl.getStudentDetails();
        sl.getMarks();
        sl.computeSGPA();
        sl.displayResults();
    }
}
```

Output:

Enter student name : Keerthi Reddy

Enter student USN : 1BM23CS137

Enter marks for subject 1 : 90

Enter credits for subject 1 : 4

Enter marks for subject 2 : 90

Enter credits for subject 2 : 4

Enter marks for subject 3 : 85

Enter credits for subject 3 : 3

Enter marks for subject 4 : 82

Enter credits for subject 4 : 3

Enter marks for subject 5 : 87

Enter credits for subject 5 : 3

Enter marks for subject 6 : 95

Enter credits for subject 6 : 1

Enter marks for subject 7 : 72

Enter credits for subject 7 : 1

Enter marks for subject 8 : 91

Enter credits for subject 8 : 1

Student Name : Keerthi Reddy

USN : 1BM23CS137

SGPA : 9.45

14/10/24

Create a class Book which contains 4 members : name, author, numPages. Include a constructor to set the values for members. Include methods to set and get the details of the objects. Include a toString() method that could display 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() {  
    return "The name of book is: " + this.name + "\n" +  
           "Author name: " + this.author + "\n" +  
           "Price of book is: " + this.price + "\n" +  
           "Number of pages: " + this.numPages + "\n";  
}
```

```
public class Bookdetails{
```

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

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

realme GT NEO 3T

```
System.out.print("Enter number of books: ");
```

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

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

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

```
System.out.print("Enter name of book: ");
```

```
s.nextLine();
```

```
String name = s.nextLine();
```

```
System.out.print("Enter author of book: ");
```

```
s.nextLine();
```

```
String author = s.nextLine();
```

```
System.out.print("Enter price of book: ");
```

```
int price = s.nextInt();
```

```
System.out.print("Enter no. of pages of book: ");
```

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

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

```
}
```

```
System.out.println("Book Details: ");
```

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

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

```
}
```

```
System.out.println("Name : k. Keerthi Reddy");
```

```
System.out.println("USN: 1BM23CS137");
```

```
sc.close();
```

```
}
```

Output :

Enter number of books : 2

Enter name of the book : harry potter

Enter author of book : J.K. Rowling

Enter price of book : 5690

Enter pages of the book : 800

Enter name of the book : cursed child

Enter author of book : J.K. Rowling

Enter price of book : 7095

Enter pages of book : 600

Book Details:

The name of book is : harry potter

Author's name : J.K. Rowling

The price of book is : 5690

Number of pages : 800

The name of book is : Cursed child

Author's name : J.K. Rowling

The price of book is : 7095

Number of pages : 600

Name of student : R. Keerthi Reddy

USN of the student : 1BM23CS137

8
H.10

realme GT NEO 3T

LAB PROGRAM - 4

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named pointArea().

Provide 3 classes named rectangle, triangle, circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method pointArea() that prints area of the given shape.

```
import java.util.Scanner;  
abstract class Shape {  
    int a, b;  
    abstract double pointArea();  
    void value(){  
        Scanner s = new Scanner(System.in);  
        a = s.nextInt();  
        b = s.nextInt();  
    }  
}
```

```
class Rectangle extends Shape {  
    double pointArea(){  
        return a * b;  
    }  
}
```

```
class Triangle extends Shape {  
    double pointArea(){  
        return 0.5 * a * b;  
    }  
}
```

```
class Circle extends Shape {  
    double printArea () {  
        return 3.14 * a * a;  
    }  
}
```

```
class ShapeArea {  
    public static void main (String args []) {  
        Rectangle r = new Rectangle ();  
        System.out.println ("Enter values of length and breadth:");  
        r.value ();
```

```
Triangle t = new Triangle ();  
System.out.println ("Enter values of base and height:");  
t.value ();
```

```
Circle c = new Circle ();  
System.out.println ("Enter the value of radius:");  
c.valueRadius ();
```

```
System.out.println ("The area of rectangle is: " +  
    r.printArea ());
```

```
System.out.println ("The area of Triangle is: " +  
    t.printArea ());
```

```
System.out.println ("The area of circle is: " +  
    c.printArea ());
```

```
}
```

Output

Enter the values of length and breadth :

2

3

Enter the values of base and height :

5

2

Enter the value of radius :

2

The area of rectangle is : 6.0

The area of Triangle is : 5.0

The area of Circle is : 12.56

Name : K. Keerthi Reddy

USN : IBM23CS137

~~Rs. 21.10~~

28/10/74

LAB PROGRAM - 5

- 1) Develop a ^{class} Program to create a Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The saving 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.
- 2) Create a ^{class} Account that stores customer name, account no 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
 - e) Check for the minimum balance, impose penalty if necessary and update balance.

Bank

Savings Accts Current Accts

1) Cont

2) Withdrawal

1) No inter

2) Checkbook

deposit / withdraw
as per
type of acc

realme GT NEO 3T

import java.util.Scanner;

class Account {

String name;
String int number;

String type;

double balance; String

Account (String name, int number, String type,
double balance) {

this.name = name;

this.number = number;

this.type = type;

this.balance = balance;

}

void deposit (double amount) {

if (amount > 0) {

balance += amount;

System.out.println ("Deposit successful.

New balance: " + balance);

} else {

System.out.println ("Invalid deposit amount.");

}

}

final double MIN_BALANCE = 1000;

final double PENALTY = 100;

CurrentAccount (String name, int number, double balance) {

super (name, number, "Current", balance);

}

void withdraw (double amount) {

if (amount > balance) {

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

realme GT NEO 3T

}

```

        balance -= amount;
        if (balance < MIN_BALANCE) {
            System.out.println("Minimum balance requirement
                not met. Penalty applied.");
            balance -= PENALTY;
        }
    }

    void showBalance() {
    }

    void writeCheck(double amount) {
        if (amount >= balance) {
            balance -= amount;
            System.out.println("Check written for : " + amount);
        } else {
            System.out.println("Insufficient balance to
                write the check.");
        }
    }

}

class SavingsAccount extends Account {
    double interestRate = 0.04;
    SavingsAccount(String name, int number, double balance) {
        super(name, number, "Savings", balance);
    }

    void addInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added. New Balance : "
            + balance);
    }
}

```

realme GT NEO 3T

```
    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance.");
            return;
        }
    }
```

```
    balance -= amount;
```

```
    showBalance();
```

```
}
```

```
public class Bank {
```

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

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

```
        System.out.print("Enter account holder's name:");
```

```
        String name = s.nextLine();
```

```
        System.out.print("Enter account number:");
```

```
        int number = s.nextInt();
```

```
        System.out.print("Enter initial balance:");
```

```
        double balance = s.nextDouble();
```

```
        System.out.print("Enter account type (1 for savings  
2 for current): ");
```

```
        int accountType = s.nextInt();
```

```
        Account account = (accountType == 1) ? new
```

```
SavingsAccount(name, number, balance) : new
```

```
CurrentAccount(name, number, balance);
```

```
        while (true) {
```

~~System.out.println("\n---Action Menu---")~~~~System.out.println("1. Deposit 2. Withdraw")~~~~3. Check Balance 4. Write check (current only) 5. Exit")~~~~int action = s.nextInt();~~

realme GT NEO 3T

```

if (action == 5) {
    System.out.println("Exiting... ");
    s.close();
    return;
}

if (action == 1) {
    System.out.print("Amount to deposit: ");
    account.deposit(s.nextDouble());
} else if (action == 2) {
    System.out.print("Amount to withdraw: ");
    if (account instanceof SavingsAccount) {
        ((SavingsAccount) account).withdraw(s.nextDouble());
    } else {
        ((CurrentAccount) account).withdraw(s.nextDouble());
    }
}

else if (action == 3) {
    account.showBalance();
} else if (action == 4 & account instanceof CurrentAccount) {
    System.out.print("Amount for check: ");
    ((CurrentAccount) account).writeCheck(s.nextDouble());
} else {
    System.out.println("Invalid action. Try again.");
}
}

```

Output

Enter Account holder's name: Geethi.

Enter account number: 1BN23S137

Enter initial balance: 5000

Enter account type (1 for savings, 2 for current): 1

realme GT NEO 3T

---Action Menu---

1. Deposit 2. Withdraw 3. Check Balance 4. Write check
(Current only) 5. Exit

Select an action: 1

Amount to deposit: 50000

Deposit successful. New balance: 505000.0

---Action Menu---

1. Deposit 2. Withdraw 3. Check Balance 4. Write check
(Current only) 5. Exit

Select an action: 2

Amount to withdraw: 504900

Current balance: 100.0

---Action Menu---

1. Deposit 2. Withdraw 3. Check Balance 4. Write check
(Current only) 5. Exit

Select an action: 5

Exiting...

Enter account holder's name: Keerthi

Enter account number: 1BN23CS132

Enter initial balance: 5000

Enter account type (1 for savings, 2 for current): 2.

---Action Menu---

1. Deposit 2. Withdraw 3. Check Balance 4. Write check (Current only) 5. Exit

Select an action: 1

Amount to deposit: 40000

Deposit successful. New balance: 405000.0

--- Action menu ---

1. Deposit
2. withdraw
3. Check balance

Select an action : 2

Amount to withdraw : 404900

Minimum balance requirement not met. Penalty applied.

Current balance : 0.0

--- Action menu ---

1. Deposit
2. withdraw
3. Check Balance

Select an action : 4

Amount for check : 8000

Check ~~were~~ written for : 8000.0

Current balance : 198000.0

R
28.10

11/11/24

Lab Program-6

Bafna Gold
Date: _____
Page: _____

- a) 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 internals marks scored in 5 courses of the current semester of student. Create another package SIE which has the class External which is derived class of Student. This class has an array that stores SIE marks scored in 5 courses of the current semester of student. Import two packages in file that declares the final marks of n students in all 5 courses.

* In CIE folder ... CIE folder Student.java */

```
package CIE;
import java.util.Scanner;
public class Student {
    protected String usn;
    protected String name;
    protected int sem;
    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter USN:");
        usn = s.nextLine();
        System.out.println("Enter Name:");
        name = s.nextLine();
        System.out.println("Enter Semester:");
        sem = s.nextInt();
    }
}
```

realme GT NEO 3T

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

* Now in Internal.java file

```
package CT;
import java.util.Scanner;
public class Internal extends Student {
    protected int[] marks = new int[5];
```

```
public void inputInternalMarks() {
    Scanner s = new Scanner(System.in);
    System.out.println("Enter internal marks for 5 courses:");
    for (int i = 0; i < 5; i++) {
        System.out.println("Enter mark " + (i + 1) + ":");
        marks[i] = s.nextInt();
    }
}
```

```
public void displayInternalMarks() {
    System.out.println("Internal marks:");
    for (int i = 0; i < 5; i++) {
        System.out.println("Course " + (i + 1) + " mark: " + marks[i]);
    }
}
```

/* Now in SEE file External.java need to change */

```
package SEE;
```

```
import CT.Internal;
import java.util.Scanner;
```

realme GT NEO 3T

```

public class External extends Internals {
    protected int[] extMarks = new int[5];
    protected int[] finalMarks = new int[5];

    public void inputSEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter external Marks for Sasi:");
        for (int i=0; i<5; i++) {
            System.out.print("External Marks "+(i+1)+":");
            externalMarks[i] = s.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for (int i=0; i<5; i++) {
            finalMarks[i] = marks[i] + extMarks[i];
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        displayCourse();
        System.out.println("External Marks:");
        for (int i=0; i<5; i++) {
            System.out.println("Course "+(i+1)+": "+extMarks[i]);
        }
        System.out.println("final Marks:");
        for (int i=0; i<5; i++) {
            System.out.println("Course "+(i+1)+": "+finalMarks[i]);
        }
    }
}

```

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

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of Students: ");
        int n = sc.nextInt();
        sc.nextLine();
```

```
External[] students = new External[n];
```

```
for (int i=0; i<n; i++) {
    students[i] = new External();
    System.out.println("Enter details of student " + (i+1));
    students[i].inputStudentDetails();
    students[i].inputCIEmarks();
    students[i].inputSEEmarks();
}
```

```
for (int i=0; i<n; i++) {
    students[i].calculateFinalMarks();
    students[i].displayFinalMarks();
}
sc.close();
```

Output:

Enter number of students: 2

Enter details of student 1

realme GT NEO 3T

Enter USN:

137

Enter Name:

Keerthi

Enter Semester:

3

Enter Internal Marks for 5 subjects:

Enter marks 1:

90

Enter marks 2:

90

Enter marks 3:

90

Enter marks 4:

90

Enter marks 5:

90

Enter External Marks for 5 subjects:

Enter marks 1:

90

Enter marks 2:

90

Enter marks 3:

90

Enter marks 4:

90

Enter marks 5:

90

Enter details for student 2

Enter USN:

142

realme GT NEO 3T

Enter name:

Kashvi

Enter semester:

3

Enter internal marks for S courses:

Enter marks 1:

85

Enter marks 2:

85

Enter marks 3:

85

Enter marks 4:

85

Enter marks 5:

85

Enter external marks for Scusses:

Enter marks 1:

85

Enter marks 2:

85

Enter marks 3:

85

Enter marks 4:

85

Enter marks 5:

85

28/11/22

Lab Program - 7

- i) Write a program that demonstrates handling of exception 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 takes both father and son's age and throws our exception if son's age is \geq father's age.

```
import java.util.Scanner;  
class WrongAge extends Exception {  
    public WrongAge() {  
        super("Age Error");  
    }  
    public WrongAge(String message) {  
        super(message);  
    }  
}  
class InputScanner {  
    Scanner s = new Scanner(System.in);  
}  
class Father extends InputScanner {  
    int fatherAge;  
    public Father() throws WrongAge {  
        System.out.print("Enter father's age: ");  
        fatherAge = s.nextInt();  
        if (fatherAge < 0) {  
            throw new Wrong("Age cannot be negative");  
        }  
    }  
}
```

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

```
}
```

```
class Son extends Father {
    int sonAge;
```

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

```
        System.out.print("Enter son's age : ");
        sonAge = s.nextInt();
    }
```

```
    if (sonAge < 0) {
        throw new WrongAge("Age cannot be negative");
    } else if (sonAge > fatherAge) {
        throw new WrongAge("Son's age cannot be greater than
                            father's age");
    }
}
```

```
}
```

```
public void FatherSon {
    public static void main(String []args)
        System.out.println("Son's age : " + sonAge);
}
```

```
}
```

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

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

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

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

realme GT NEO 3T

realme GT NEO 3T Name: K. Keerthi Reddy)

3
System.out.println("USN: IBM23CS137")

Output:

Enter father's age : 50

Enter son's age : 18

Son's age : 18

Father's age : 50

Name : K.Keerthi Reddy

USN : IBM23CS137.

Enter father's age : 18

Enter son's age : 29

Son's age cannot be greater than or equal to father's age.

Name : K.Keerthi Reddy

USN : IBM23CS137

28/11/12

Lab Program - 8

Q) 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 DisplayMessage1 extends Thread {  
    public void run() {  
        for (int i = 0; i < 5; i++) {  
            try {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(1000);  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```
class DisplayMessage2 extends Thread {  
    public void run() {  
        for (int i = 0; i < 5; i++) {  
            try {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```
public class multithread {
```

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

```
        DisplayMessage1 thread1 = new DisplayMessage1();
```

```
        DisplayMessage2 thread2 = new DisplayMessage2();
```

realme GT NEO 3T

```
Date: _____ Page: / 8  
Thread1.start();  
Thread2.start();  
System.out.println("Name: k.Keerthi Reddy");  
System.out.println("USN: IBM23CS137");  
}  
g
```

Output:

Name : k.Keerthi Reddy

USN : IBM23CS137

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS College of Engineering

BMS College of Engineering

BMS College of Engineering

BMS College of Engineering

28/11/22

Lab Program 10A

- 3) An incorrect implementation of a producer & consumer.

```
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("InterruptedException caught");
            }
        }
        System.out.println("Got " + n);
        valueSet = false;
        System.out.println("Notify Producer (" + n + ")");
        notify();
    }
    synchronized void put(int n) {
        while (valueSet) {
            try {
                System.out.println("Producer waiting");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        }
        this.n = n;
        valueSet = true;
    }
}
```

realme GT NEO 3T
valueSet = false;

System.out.println("Put " + n)

System.out.println("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<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 s = q.get();

System.out.println("Consumed " + s);

}

realme GT NEO 3T

3

```
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

Producer waiting ...

Get: 0

put: 1

Producer waiting ...

Consumed: 0

Get: 1

Consumed: 1

Put 2

Producer waiting ...

Get: 2

Consumed: 2

put: 3

Producer Waiting ...

Get: 3

Consumed: 3

Put: 4

Producer Waiting ...

Get: 4

Consumed: 4

realme GT NEO 3T

Put: 5

Producer waiting ...

Got: 5

Put: 6

Producer waiting ...

Consumed: 5

Got: 6

Consumed: 6

Put: 7

Producer waiting ...

Got: 7

Consumed: 7

Put: 8

Producer waiting ...

Got: 8

Consumed: 8

Put: 9

Producer waiting ...

Got: 9

Put: 10

Consumed: 9

Producer waiting ...

Got: 10

Consumed: 10

Put: 11

Producer waiting ...

Got: 11

Consumed: 11

Put: 12

Producer waiting ...

Got: 11

Consumed: 12

Put: 13

realme GT NEO 3T

28/11/2024

Lecture 10b

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

realme GT NEO 3T

```
class Deadlock implements Runnable {  
    A a = new A();  
    B b = new B();  
  
    Deadlock() {  
        Thread currentThread = Thread.currentThread().setName("Main Thread");  
        Thread t = new Thread(this, "Racing Thread");  
        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");  
    }  
}
```

Output:

Main Thread entered A.foo
Racing Thread entered B.bar
Racing Thread trying to call A.last()
Main Thread trying to call B.last()

28/11/24

Lab Program - 9

Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in Result field when the Divide button is clicked.

If Num1 & Num2 were not an integer, program would throw a Number Format Exception. If Num2 was zero, the program would throw an Arithmetic Exception displaying the exception in message dialog box.

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

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

        JLabel jlab = new JLabel ("Enter Divides and divided");
        JTextField aJTF = new JTextField (8);
        JTextField bJTF = new JTextField (8);

        JButton button = new JButton ("Calculate");
    }
}
```

```
jLabel err = new JLabel();  
JLabel abt = new JLabel();  
JLabel blab = new JLabel();  
JLabel anslab = new JLabel();  
ifrom.add(err);  
ifrom.add(jlab);  
ifrom.add(atf);  
ifrom.add(btf);  
ifrom.add(button);  
ifrom.add(alab);  
ifrom.add(blab);  
ifrom.add(anslab)
```

```
button.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        try {  
            int a = Integer.parseInt(atf.getText());  
            int b = Integer.parseInt(btf.getText());  
            int ans = a / b;  
            alab.setText("A = " + a);  
            blab.setText("B = " + b);  
            anslab.setText("Ans = " + ans);  
            err.setText("");  
        } catch (NumberFormatException e) {  
            alab.setText("");  
            blab.setText("");  
            anslab.setText("");  
            err.setText("Enter only Integers");  
        }  
    }  
});
```

```

82
    catch (ArithmaticException e) {
        alab.setText(" ");
        blab.setText(" ");
        anelab.setText(" ");
        err.setText("B should be non zero."); }

    }
}

if (from.setVisible(true));

```

```

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

```

Output:

Divide App - □ X
 enter the divisor and dividend:
 52 14
 calculate A=52 B=14 - Ans = 3