__/__/____

INDEX

SR NO	Date	Content	Pg NO	Remarks
1	12-12-24	Lab 1 : Quadratic Roots	1-3	
2	19-12-24	Lab 2 : SGPA	7-10	
3	26-12-24	Lab 3 : Books	11-13	
4	2-1-24	Lab 4 : Abstract Class	14-16	
5	9-1-24	Lab 5(a) : Bank Account	17-22	
		(b) : generics	26-28	
6	23-1-24	Lab 6 : Packages	29-32	
7	31-1-24	Lab 7 : Exceptions	33-35	
8	6-2-24	Lab 8 : Threads	36-37	
9	13-2-24	Lab 10 : IPC & Deadlock	38-42	
10	20-2-24	Lab 9 : Interface for Int. divisions	43-45	

LAB

cd desktop
cd cs256001 [Folder name]
javac — java [class name]
java — [class name]
12/12/23

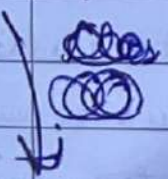
①

LAB 1: Quadratic Roots: Develop a Java program that prints all real solutions to the quadratic eq $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
class Quadratic
{
    int a, b, c;            //coeffs
    double r1, r2, d;       //roots
    void getcl()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficient 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 of a");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
```

(2)

```
d = b*b - 4*a*c;
if (d==0)
{
    r1 = (-b) / (2*a);
    System.out.println("Roots are real & equal");
    System.out.println("Root 1 = Root 2 = " + r1);
}
else if (d>0)
{
    r1 = ((-b) + (Math.sqrt(d))) / (double)(2*a);
    r2 = ((-b) - (Math.sqrt(d))) / (double)(2*a);
    System.out.println("Roots are real & distinct");
    System.out.println("Root1 = " + r1 + "Root 2 = " + r2);
}
else if (d<0)
{
    System.out.println("Roots are imaginary");
    r1 = (-b) / (2*a);
    r2 = Math.sqrt(-d) / (2*a);
    System.out.println("Root 1 = " + r1 + " + i " + r2);
    System.out.println("Root 2 = " + r1 + " - i " + r2);
}
}
}
```



__/__/____

③

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

O/P:

① Enter the coefficients of a, b, c
2 3 4
Roots are imaginary
Root1 = 0.0 + i1.19895 7
Root2 = 0.0 - i1.19895 7

② Enter the coefficients of a, b, c
5 0 0
Roots are real and equal
Root1 = Root2 = 0.0

③ Enter the coefficients a, b, c
0 0 0
Not a quadratic equation
Enter a non zero value of a:
6
Roots are ~~equal~~ real and equal
Root1 = Root2 = 0.0

10

12/12/23

NAME = SHREYA MITAWA
USN = 1BM22CS266

__/__/____

• Additional Programs:

Q1 To write a program in Java to find the area of a rectangle & verify the same with various inputs (length, breadth) [using ParseInt() method]

```
class RectangleArea {
{
    public static void main (String args [])
    {
        int length, breadth;
        length = Integer.parseInt (args [0]);
        breadth = Integer.parseInt (args [1]);
        int area = length * breadth;
        System.out.println ("length of rectangle = " + length);
        System.out.println ("breadth of rectangle = " + breadth);
        System.out.println ("area of rectangle = " + area);
    }
}
```

(5)

Q2 Illustrate an example for Scanner.

```
import java.util.Scanner;
class HelloWorld
{
    public static void main (String args [])
    {
        int a; float b; String s;
        Scanner in = new Scanner (System.in);
        System.out.println ("enter a string");
        s = in.nextLine ();
        System.out.println ("you entered string "+s);
        System.out.println ("Enter an Integer");
        a = in.nextInt ();
        System.out.println ("you entered integer "+a);
        System.out.println ("Enter a float");
        b = in.nextFloat ();
        System.out.println ("you entered float "+b);
    }
}
```

O/P

```
Enter a string
Java
You entered string java
Enter a integer
67
You entered integer 67
Enter a float
7.6
You entered float 7.6
```

Q3 Illustrate an example for 1D Array.

```
class AutoArray
{
    public static void main (String args[])
    {
        int month_days = new int [12];
        int month_days [] = {31,28,31,30,31,30,31,31,30,31,30,31};
        System.out.println ("April has " + month_days [3]
                            + " days");
    }
}
```

O/P

April has 31 days.

Name: Shreya Mitaiva
USN: 1BM22CS266

19/12/23

LAB 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

Formula: $SGPA = \frac{\sum [(\text{course credits})(\text{Grade Points})]}{\sum (\text{course credits})}$

```
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);
```

(8)

```
void getStudentDetails()
{
    System.out.println("Enter your Name:");
    name = s.nextLine();
    System.out.println("Enter your USN:");
    usn = s.next();
}
void getMarks()
{
    for(int i=0; i<9; i++)
    {
        System.out.println("Enter Marks for subject "
                + (i+1)+":");
        subject[i].subjectMarks = s.nextInt();
        System.out.println("Enter credits for subject "
                + (i+1)+":");
        subject[i].credits = s.nextInt();
        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;
```

(9)

```
        for (int i=0; i<9; 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);
    }
}
```

(12)

O/P

```
Enter your name Shreya
Enter your USN:
1BM22CS266
Enter marks for subject 1: 88
Enter your credits for subject 1: 4
Enter marks for subject 2: 89
Enter your credits for subject 2: 4
Enter marks for subject 3: 92
Enter your credits for subject 3: 3
Enter marks for subject 4: 90
Enter your credits for subject 4: 3
Enter marks for subject 5: 87
Enter your credits for subject 5: 3
Enter marks for subject 6: 86
Enter your credits for subject 6: 2
Enter marks for subject 7: 89
Enter your credits for subject 7: 2
Enter marks for subject 8: 95
Enter your credits for subject 8: 1
Name: Shreya
USN: 1BM22CS266
SGPA: 9.318181818
```

19/12/23

Name: Shreya Nitawa
USN: 1BM22CS266

LAB 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 & get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a Java program to create n book objects

```
import java.util.Scanner;

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

    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 name = "Book name:" + this.name + "\n";
        String author = "Author name:" + this.author + "\n";
        String price = "Price:" + this.price + "\n";
        String numPages = "No of Pages:" + this.numPages + "\n";
        return name + author + price + numPages;
    }
}
```

(12)

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

        System.out.print ("Enter the no of books : ");
        int n = s.nextInt();

        Book b[] = new Book [n];

        for (int i = 0; i < n; i++) {
            System.out.print ("Enter name of the Book : ");
            String name = s.next ();

            System.out.print ("Enter author of the Book : ");
            String author = s.next ();

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

            System.out.print ("Enter No. of pages of the Book : ");
            int numPage = s.nextInt();

            b[i] = new Book (name, author, price, numPages);
        }
        System.out.println ("\n Book Details : ");
        for (int i = 0; i < n; i++) {
            System.out.println ("Book" + (i+1) + ": \n" + b[i]);
        }
    }
}
```

(13)

O/P

Enter the nuber of books : 2
Enter name of the book : Java
Enter autter of the book : John
Enter the price of the book : 250
Enter the nuber of pages of the book : 100
Enter Name of the book : dbms
Enter auther of the book : George
Enter the peice of the book : 300
Enter the number of pages of the book : 150

Book details :
Book 1 :
Book name : java
Author name : John
Price : 250
Numbers of pages : 100

Book 2 :
Book name : dbms
Author name : george
Price : 300
Number of pages : 150

26/12/23

Name : Shreya Mitawa
USN : 1BM22CS266

(14)

LAB 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.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 Rectangle extends Shape {
    void getInput() {
        System.out.print("Enter length & width of rec:");
        a = s.nextDouble();
        b = s.nextDouble();
    }
    void displayArea() {
        System.out.println("Area of Rec:" + (a*b));
    }
}
```

(15)

```
class Triangle extends Shape() {
    void getInput() {
        System.out.print("Enter base & height of tri!");
        a = s.nextDouble();
        b = s.nextDouble();
    }
    void displayArea() {
        System.out.println("Area of tri: " + (0.5*a*b));
    }
}
class Circle extends Shape() {
    void getInput() {
        System.out.print("Enter radius of Circle: ");
        a = s.nextDouble();
    }
    void displayArea() {
        System.out.println("Area of Circle: " +
                (Math.PI * a * a));
    }
}
public class Main {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle();
        Triangle triangle = new Triangle();
        Circle circle = new Circle();

        rectangle.getInput();
        rectangle.displayArea();
        triangle.getInput();
        Triangle.displayArea();
        circle.getInput();
        circle.displayArea();
    }
}
```

(16)

O/P Enter length and width of rectangle: 10 20
Area of Rectangle: 200
Enter base and height of triangle: 10 30
Area of Triangle: 150
Enter radius of Circle: 15
Area of Circle: 706.8583

02/01/24

Name: Shreya Nitawa
USN: 1BM22CS266

(17)

9/11/24

Lab 5 (9) Develop a Java program to create Bank that ..... savings acc & current acc.
Include necessary methods
(a) Accept deposit & update the balance.
(b) Display
(c) Compute & deposite interest
(d) Permit withdrawal & update the balance.
(e) Check for min balance, impose penalty if needed & updt

```
import java.util.Scanner;
class Account
{
    String customerName;
    String accountNumber;
    String accountType;
    double balance;

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

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

```
void displayBalance()
{
    System.out.println("Account Type: " + AccountType);
    System.out.println("Customer Name: " + customerName);
    System.out.println("Account Number: " + accountNumber);
    System.out.println("Balance: " + balance);
}
}

class SavAct extends Account
{
    SavAct(String customerName, String accountNumber,
           double balance)
    {
        super(customerName, accountNumber, "Savings", bal);
    }
    void computeInterest(double interestRate)
    {
        double interest = balance * interestRate;
        deposit(interest);
        System.out.println("Interest computed &
            deposited. Updated balance: " + balance);
    }
    void withdraw(double amount)
    {
        if (amount <= balance)
            balance -= amount;
            System.out.println("withdrawn successfully.
                updated balance: " + balance);
        else
            System.out.println("Insufficient funds.
                Withdrawal failed");
    }
}
```

(19)

```
class CurAcct extends Account
{
    double serviceCharge;
    CurAcct(String customerName, String accountNumber, double balance,
                                                    double serviceCharge)
    {
        super(customerName, accountNumber, "Current", balance);
        this.serviceCharge = serviceCharge;
    }
    void withdraw(double amount)
    {
        if (amount <= balance)
            balance -= amount;
            System.out.println("withdrawal successful. Updated
                balance: " + balance);
        else
            System.out.println("Insufficient funds, Withdrawal failed");
    }
}
public class Main
{
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Customer Name: ");
        String customerName = s.nextLine();
        System.out.println("Enter account number: ");
        String accountNumber = s.nextLine();
        System.out.println("Enter initial balance: ");
        double initialBalance = s.nextLine();

        SavAcct savingsAccount = new SavAcct(customerName,
                                    accountNumber, initialBalance);
        CurAcct currentAcct = new CurAcct(customerName,
                                    accountNumber, initialBalance
                                    ,50);

        int choice;
        double amount;
        double interestRate = 0.05;
    }
```

```
do {
    Sys.o.pln("\nMenu:"\n 1. Deposit.\n, 2. displayBalance
        \n 3. Compute Interest (Savings Acc)\n
        4. Withdraw\n 5. Exit\n 6 'your Choice:');
    choice = s.nextInt();
switch(choice){
    case 1:
        Sys.o.p("Enter the amount to deposit:");
        amount = s.nextDouble();
        Sys.o.p("Choose account type (1. Savings / 2. Current):");
        int accountTypeChoice = s.nextInt();
        switch (acrountTypeChoice)
        {   case 1:
                savingssAccount.deposit(amount);
                break;
            case 2:
                currentAccount.deposit(amount);
                break;
        }   default: Sys.o.p("Invalid Input");
        break;

    case 2:
        Sys.o.p("Choose account type (1. Savings (2. Current):");
        int displayChoice = s.nextInt();
        switch (displayChoice)
        {   case 1: savingsAccount.displayBalance();
                break;
            case 2: currentAccount.displayBalance();
                break;
            default:
                Sys.o.p("Invalid Input");
        }
        break;

    case 3:
        Sys.o.println("Enter the interestRate:");
        interestRate = s.nextDouble();
        SavingsAccount.computeInterest(interestRate);
        break;
```

```
case 4:
    sy.out.p("enter the amount to withdraw: ");
    amount = s.nextDouble();
    s.o.p("Choose account type (1.savings 2.current)");
    int withdrawChoice = s.nextInt();
    switch (withdraw)
    {
        case 1:
            SavingsAccount.withdraw(amount);
            break;
        case 2:
            CurrentAccount.withdraw(amount);
            break;
        default:
            S.O.P("Invalid Input");
    }
    break;

case 5:
    system.out.println("Exiting the program....");
    break.
default:
    system.out.println("Invalid choice. Please try again");
}
}while(choice != 5);
s.close();
}
}
```

o/p →

O/P

Enter customer name: Shreya.
Enter account number: 12345789
Enter initial balance: 155879

Menu:
1. Deposit
2. display Balance
3. Compute Interest (Savings Account)
4. withdraw
5. Exit
Enter your choice: 1
Enter the amount to deposit: 11000
Choose account Type (1. savings / 2. current): 2
deposit successful. Updated balance: 166879

Menu:
1. deposit
2. display Balance
3. Compute Interest (Savings Account)
4. withdraw
5. Exit
Enter your choice: 3
Enter the interest rate: 0.2
deposit successful. Updated balance: 187054.8
Interest computed and deposited. Upload balance: 187054.8

10/10/24

NAME: Shreya Mitawa
USN: 1BMERCS266

~~Lab-6~~ Sample Programs

16/1/24

1. Demonstrate various String constructor with proper java program

```
class String {
    public static void main (String args[]) {
        char c[] = {'J', 'a', 'v', 'a'};
        String s1 = new String(c);
        String s2 = new String(s1);
        System.out.println(s1);
        System.out.println(s2);
    }}
```

O/P: Java
Java

8. Demonstrate startswith() to give output true or false.

```
public class Main {
    public static void main (String args[]) {
        String test = "testString";
        String pattern = "te";
        System.out.println(test.startswith(pattern));
        pattern = "est";
        System.out.println(test.startswith(pattern));
    }
}
```

O/P:
True
False

19. Write a Java program to create an abstract class Bird with abstract method fly() and makeSound(). Create subclass 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 void fly();
    abstract void makeSound();
}

class Eagle extends Bird {
    void fly() {
        System.out.println("Eagle can fly very high");
    }
    void makeSound() {
        System.out.println("Eagle makes a screech sound");
    }
}

class Hawk extends Bird {
    void fly() {
        System.out.println("Hawk can fly moderatlly high");
    }
    void makeSound() {
        System.out.println("Hawk makes a shrill sound");
    }
}
```

```
public class Main {
    public static void main(String args[]) {
        Bird bird1 = new Eagle();
        bird1.fly();
        bird1.makeSound();

        Bird bird2 = new Hawk();
        bird2.fly();
        bird2.makeSound();
    }
}
```

O/P:

Eagle can fly very high
Eagle makes a screech sound
Hawk can fly moderatlly high
Hawk makes a shrill sound

NAME: Shreya Nitawa
USN: 1BM22CS266

26 Lab 5(b)

Generic

Write a Java program to create a generic class Stack which holds 5 integers and 5 double values.

```
import java.util.EmptyStackException;
public class Stack<T>{
  private int maxSize;
  private int top;
  private Object[] stackArray;

public Stack(int size){
      maxSize = size;
      stackArray = new Object[maxSize];
      top = -1;
   }

public void push(T value){
   if(top<maxSize-1){
         top++;
        StackArray[top] = value;
  }else{
   throw new RuntimeException
        ("Stack is full");
  }
}

@Suppress Warnings("unchecked")
 public T pop(){
   if(!isEmpty()){
      return (T)stackArray[top--];
```

```
}else {
  thrownew EmptyStackException();
 }
}

public boolean isEmpty(){
    return (top==-1);
 }

public int size(){
    return top+1;
}

public static void main(String args[]){
   Stack<Integer> intStack = new Stack<X(5);
   Stack<Double> doubleStack = new Stack<>(5);

 for(int i=0; i<5; i++){
     intStack.push(i);
     doubleStack.push(i);
  }

  System.out.println("Integer Stack");
   for(int i=0; i<5; i++){
      System.out.println(intstack.pop());
    }
  System.out.println("Double Stack");
   for(int i=0; i<5; i++){
      System.out.println(doubleStack.pop());}
}}
```

O/P:

Integer Stack:

4
3
2
1
0

Double Stack:

4.0
3.0
2.0
1.0
0.0

17/01/24

NAME: Shreya Mitawa

USN: 1BM22CS296

23/1/24.

Lab 6

Q Create a package CIE has 2 classes Student & Int... ----- n students in all five courses ---

```
package CIE
import java.util.Scanner;
public class Student {
        public String usn, name;
        public int sem;

    public void inputStudentDetails() {
            Scanner sc=new Scanner(System.in);
            S.o.p("Enter the student usn:");
            usn = sc.nextLine();
            S.o.p("Enter the student name:");
            name = sc.nextLine();
            S.o.p=("Enter student semester:");
            sem = sc.nextInt();
            }
        public void display() {
            S.o.p("Student usn"+usn);
            S.o.p("Student name"+name);
            S.o.p("Student sem"+sem);
            }
    }

package CIE;
import java.util.Scanner;
public class Internals extends Student {
        public int marks[] = new int[5];
        public void inputCIEmarks() {
            Scanner sc= new Scanner(System.in);
            for...
```

(30)

Scan

```
for (int i=0; i<5; i++) {
    S.o.p("Enter marks of subject"
        + (++i) + ":");
    marks[i] = sc.nextInt();
}
}
}
```

```
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends CIE.Internals {
    public int marks[] = new int[5];
    public int finalMarks[] = new int[5];

    public void getSEEMarks() {
        Scanner sc = new Scanner(System.in);
        for (int i=0; i<5; i++) {
            S.o.p("Enter" + (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];
        }
    }
}
```

```
import SEE.Externals;
class PackageMain {
    public static void main(String args[]) {
        int numOfStudent = 2;
        Externals finalMarks[] = new External[numOfStudent];
```

```
for(int i=0; i<numOfStudent; i++) {
    finalMarks[i] = new Externals();
    finalMarks[i].getStudentInfo();
    S.o.p("Enter CIE Marks:");
    finalMarks[i].getCIEMarks();
    S.o.p("Enter SEE Marks:");
    finalMarks[i].getSEEMarks();
}
System.out.println("Display data:");
for(int i=0; i<5; i++) {
    finalMarks[i].display();
    finalMarks[i].calculateFinalMarks();
    finalMarks[i].getFinalMarks();
}
}
```

O/P

```
Enter Student USN: 1BM22CS264
Enter Student name: Shreya
Enter student semester: 3
Enter CIE Marks:
Enter 1 marks: 45
Enter 2 Marks: 44
Enter 3 Marks: 46
Enter SEE marks:
Enter 1 Marks: 98
Enter 2 Marks: 95
Enter 3 Marks: 96
```

32

Enter student USN: 1BM22CS265
Enter student name: Shreya
Enter Student semester: 3
Enter CIE Marks:
Enter 1 Marks: 45
Enter 2 Marks: 44
Enter 3 Marks: 43
Enter SEE Marks:
Enter 1 Marks: 97
Enter 2 Marks: 98
Enter 3 Marks: 96

displaydata:
Student Name: Shreya
Student USN: 1BM22CS266
Semester: 3
Marks of 1 subject: 94
Marks of 2 subject: 91
Marks of 3 subject: 94:

Student Name: Shreya S
student USN: 1BM22CS265
Semester: 3
Marks of 1 subject: 91
Marks of 2 subject: 93
Marks of 3 subject: 92

NAME: Shreya Mutanal
USN: 1BM22CS266

(33) 23/1/24

Lab 7

Write A program for handling of exceptions ..... if son's age is >= father's age.

```
class WrongAge extends Exception {
        WrongAge(String msg) {
            super(msg);
        }
}
import java.util.Scanner;
class InputScanner {
        Scanner sc;
        InputScanner() {
            sc = new Scanner(System.in);
        }
}
class father extends InputScanner {
    int fatherAge;
    father() throws WrongAge {
        S.o.p("Enter father's age:");
        fatherAge = sc.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negetive");
        }
    void display() {
            S.o.p("father's age:" + fatherAge);
    }
}

class Son extends father {
    int sonAge;
    Son() throws WrongAge {
        S.o.p("Enter Son's age:");
        sonAge = sc.nextInt();
```

(34)

```
        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();
        S.o.p("Son's Age: " + sonAge);
    }

    public static void main(String args[]) {
        try {
            Father son = new Son();
            son.display();
        }
        catch(WrongAge e) {
            S.o.p(e);
        }
    }
}
```

```
Enter father's age:
-12
Wrong Age: Age cannot be negative
Enter father's age:
25
Enter son's age:
45
Wrong Age: Son's age cannot be greater than father's age
```

(35)

enter father's age:
45
enter son's age:
22
father's age: 45
son's age: 22

NAME: Shreya Mittal
USN: 1BM22CS266

(26) 6/2/24

Lab 8

→ WAP to create 2 threads, BMSCE → once every 10 sec & "CSE" → once every 2 sec

```
class BMSCE implements Runnable {
    public void run() {
        while (true) {
            try {
                S.O.P ("BMS College of Engineering");
                Thread.sleep (10000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
class CSE implements Runnable {
    public void run() {
        while (true) {
            try {
                S.O.P ("CSE");
                Thread.sleep (2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
public class Main {
    public static void main (String[] args) {
        Thread t1 = new Thread (new BMSCE());
        Thread t2 = new Thread (new CSE());
        t1.start();
        t2.start();
    }
}
```

32

__/__/__

o/p

BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE

NAME: Shreya Nitturu
USN: 1BM22CS266

6/2/24

Lab 10

Demonstrate IPC & deadlock.

IPC

```
class Q {
int n;
boolean valueSet = false;
synchronized int get() {
while (!valueSet)
try {
S.o.p ("\n Consumer waiting\n");
wait();
} catch (InterruptedException e) {
S.o.p ("InterruptedException caught");
}
S.o.p ("Got: " + n);
valueSet = false;
S.o.p ("\n Intimate Producer\n");
notify();
return n;
}
synchronized void put(int n) {
while(valueSet)
try {
S.o.p ("\n Producer waiting\n");
wait();
} catch (InterruptedException e) {
S.o.p ("InterruptedException caught");
}
this.n = n;
valueSet = true;
```

```
s.o.p ("put:"+n);
s.o.p ("\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<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();
s.o.p ("consumed:" +r);
i++;
}
```

(20)

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

O/P

```
Put : 1
Got : 1
Put : 2
Got : 2
Put : 3
Got : 3
Put : 4
Got : 4
Put : 5
Got : 5
```

NAME: Shreya Mitawa
USN: 1BM22CS266

(4)

DEADLOCK

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

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

```
void last() {
S.O.P ("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);
S.O.P("Back in main thread");
}
public void run() {
b.bar(a);
S.O.P("Back in other thread");
}
public static void main(String args[]) {
new Deadlock();
}
}
```

O/P

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

8.2.24

NAME: Shreya Nibaua
USN: 1BM22CS266

__/__/__

Lab 9:

Write a program that creates a user interface to perform integer division ..... 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 labels
        JLabel jlab = new JLabel("Enter the divider & 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 orders

```
jfrm.add(err);          // to display borrar 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) {
        S.o.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 aus = a/b;

            alab.setText("\n A=" + a);
            blab.setText("\n B=" + b);
            anslab.setText("\n Ans=" + aus);
        }
        catch (NumberFormatException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter only Integers!");
        }
    }
});
```

```
catch (ArithmeticException e) {
    alab.setText("");
    blab.setText("");
    ansLab.setText("");
    err.setText("B should be NON zero!");
}
});
// 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();
    }
});
}
```

O/P

(1)

Enter the divider and dividend:

80 | 2

Calculate A=80 B=2 Ans=40

(2)

B should be NON zero!

Enter the divider and dividend:

80 | 0

Calculate

20.02.24

(16)

__/__/____

Event Functions:

- `ajtf.addActionListener(l);`
  = Associates the ActionListener with the first text field to handle action events.

- `bjtf.addActionListener(l);`
  = Associates the ActionListener with the second text field to handle action events.

- `button.addActionListener(... ) {`
  = performs decision based on user i/p & updates labels accordingly.

- `ActionListener l = e -> s.o.p("....");`
  = prints a message to the console when an action event occurs in a text field.

- `Integer.parseInt(String s)`:
  = it parses the specified string as an integer, it is used to convert the text entered into integers.

- `JFrame.setDefaultCloseOperation(int Operation);`
  = it is set to exit the application when the user closes JFrame

- `JFrame.add(Component comp)`
  = this method adds the specified component to this container.

- `JFrame.setVisible(boolean b)`: This method makes the JFrame visible or invisible.

30.02.24

NAME: Shreya Nutana
USN: 1BM22CS288

LAB: 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a 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("Roo1 = 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 + " + 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();
	}
}
```

LAB: 2

Develop a Java program to create a class Student with members usn, name, an array credits and an array mark. Include methods to accept and display details and a method to calculate SGPA of a student.
//Develop a java program to create a class Student with members usn,name, an array creadits 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 Student{
   String name;
   String usn;
   double SGPA;
   Subject subject[];
   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 subject "+(i+1)+":");
         subject[i].subjectMarks=s.nextInt();
         System.out.println("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 totalCreadits=0;
```

```
      for(int i=0;i<9;i++){
         effectiveScore += (subject[i].grade*subject[i].credits);
         totalCreadits += subject[i].credits;
      }

      SGPA=(double)effectiveScore/(double)totalCreadits;
   }
}
class Subject{
         int subjectMarks;
         int credits;
         int grade;

}
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);
         }
}
```

LAB: 3

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;
class Book{
   String name, author;
   int price,page_num;

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

   String toStrings(){
      String name,author,price,page_num;
      name="Book name: "+this.name+"\n";
      author="Author name: "+this.author+"\n";
      price="Price: "+this.price+"\n";
      page_num="Number of pages: "+this.page_num+"\n";

      return (name+author+price+page_num);
   }

   public static void main(String args[]){
      int n;
      String name,author;
      int price,page_num;

      Scanner sc=new Scanner(System.in);

      System.out.println("Enter the no. of books:");
      n=sc.nextInt();

      Book b[]=new Book[n];

      for (int i=0;i<n;i++){
         System.out.println("Enter name of book:");
         sc.nextLine();
         name=sc.nextLine();

         System.out.println("Enter author of a book:");
         author=sc.nextLine();

         System.out.println("Enter the price of book:");
         price=sc.nextInt();

         System.out.println("Enter the no. of pages of book:");
         page_num=sc.nextInt();
```

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

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

LAB: 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.Scanner;
public class InputScanner {
	Scanner s;
	InputScanner(){
		s=new Scanner(System.in);
	}
}
abstract public class Shape extends InputScanner{
	double a,b;
	abstract void getInput();
	abstract void displayArea();
}
public class Rectangle extends Shape {
	void getInput() {
		System.out.println("Enter the dimension of rectangle:");
		a=s.nextDouble();
		b=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of reactangle:"+(a*b));
	}
}
public class Circle extends Shape{
	void getInput() {
		System.out.println("Enter the dimension of circle:");
		a=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of circle:"+(3.14*a*a));
	}
}
public class Triangle extends Shape {
	void getInput() {
		System.out.println("Enter the dimension of triangle:");
		a=s.nextDouble();
		b=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of triangle:"+((a*b)/2));
	}
}
public class MainMethod {

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

```
                Rectangle R=new Rectangle();
                R.getInput();
                R.displayArea();

                Triangle T=new Triangle();
                T.getInput();
                T.displayArea();

                Circle C=new Circle();
                C.getInput();
                C.displayArea();
        }

}
```

LAB: 5

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

```
import java.util.Scanner;
public class Account {
	String customer_name;
	int account_no;
	String type_acc;
	double balance=0;
	Scanner sc;

	Account(String customer_name,int account_no,String type_acc){
		this.customer_name=customer_name;
		this.account_no=account_no;
		this.type_acc=type_acc;
		sc=new Scanner(System.in);
	}

	void diposite() {
		System.out.println("Enter the deposit amount:");
		int dipo=sc.nextInt();
		balance+=dipo;
	}

	void withdrawal() {
		System.out.println("Enter the withdrawl amount:");
		int with=sc.nextInt();
		if(with>balance) {
			System.out.println("Insufficient Balance");
		}
		else{
			balance-=with;
		}
	}

	void display() {
		System.out.println("Customer name:"+customer_name);
		System.out.println("Account number:"+account_no);
		System.out.println("Type of Account:"+type_acc);
		System.out.println("Balance:"+balance);
	}
	void applyinterest() {}

}

public class Cur_acct extends Account {

	Cur_acct(String customer_name,int account_no,String type_acc){
		super(customer_name,account_no,type_acc);}
```

```
	void withdrawal() {
		System.out.println("Enter the withdrawl amount:");
		int with=sc.nextInt();
		if (balance<=2000) {
			double pen=balance/(0.06);
			System.out.println("Insuffiecient balance penalty to be paid:"+pen);
			balance+=pen;
		}
		else{
			balance-=with;
		}
	}
}

public class Sav_acct extends Account{
Sav_acct(String customer_name,int account_no,String type_acc){
super(customer_name,account_no,type_acc);
}

void applyinterest() {
System.out.println("Enter the interest rate:");
int rate=sc.nextInt();
double interest=balance*rate;
balance+=interest;
System.out.println("Balance after interest:"+balance);
}
}

import java.util.Scanner;
public class Bank {
	public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter customer name:");
		String cus_name=sc.nextLine();
		System.out.println("Enter account number:");
		int acc_no=sc.nextInt();

		Account ca=new Cur_acct(cus_name,acc_no,"Current Account");
		Account sa=new Sav_acct(cus_name,acc_no,"Saving Account");

		int choice;
		while(true){
			System.out.println("----MENU----");
			System.out.println("1.Deposite\n2.Withdrawl\n3.Compute intereset for Saving
			account\n4.Display account details\n5.Exit");
			choice=sc.nextInt();

			switch(choice) {
			case 1:
			System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
			int acc=sc.nextInt();
			if(acc==1) {
```

```
                    sa.diposite();
                    }
                    else {
                    ca.diposite();
                    }
                    break;
                    case 2:
                    System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
                    int acc1=sc.nextInt();
                    if(acc1==1) {
                    sa.withdrawal();
                    }
                    else {
                    ca.withdrawal();
                    }
                    break;
                    case 3:
                    sa.applyinterest();
                    break;
                    case 4:
                    System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
                    int acc2=sc.nextInt();
                    if(acc2==1) {
                    sa.display();
                    }
                    else {
                    ca.display();
                    }
                    break;
                    case 5:
                    break;
                    default:
                    System.out.println("Invalid Choice");
                    break;
                    }

                    if(choice==5) {
                    break;
                    }
                }
        }
}
```

LAB: 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 Student{
   public String usn,name;
   public int sem;
   public void inputStudentDetails(){
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the student usn:");
      usn=sc.nextLine();
      System.out.println("Enter the student name:");
      name=sc.nextLine();
      System.out.println("Enter student semester:");
      sem=sc.nextInt();
   }

   public void disply(){
      System.out.println("Student USN:"+usn);
      System.out.println("Student Name:"+name);
      System.out.println("Student Sem:"+sem);
   }
}

package CIE;
import java.util.Scanner;
public class Internals extends Student{
   public int marks[]=new int[5];
   public void inputCIEmarks(){
      Scanner sc=new Scanner(System.in);
      for(int i=0;i<5;i++){
         System.out.println("Enter the marks for subject "+(i+1)+":");
         marks[i]=sc.nextInt();
      }
   }
}
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends CIE.Internals{
   public int marks[];
   public int finalMarks[];

   public Externals(){
      marks=new int[5];
      finalMarks=new int[5];
   }
```

```
    public void inputSEEMarks(){
        Scanner sc=new Scanner(System.in);
        for(int i=0;i<5;i++){
            System.out.println("Enter 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(){
        inputStudentDetails();
        for(int i=0;i<5;i++){
            System.out.println("Subject "+(i+1)+" final marks:"+finalMarks[i]);
        }
    }
}
import SEE.Externals;
class PackageMain{
    public static void main(String args[]){
        int numOfStudent=2;
        Externals finalMarks[]=new Externals[numOfStudent];
        for(int i=0;i<numOfStudent;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<numOfStudent;i++){
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].disply();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

LAB: 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.

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

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

   }

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

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

```
        try{
            Son son=new Son();
            son.display();
        }
        catch(WrongAge e){
            System.out.println(e);

        }
    }
}
```

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 BMS_College_of_Engineering implements Runnable {
   public void run() {
      while (true) {
         try {
            System.out.println("BMS College of Engineering");
            Thread.sleep(10000); // Sleep for 10000 milliseconds (10 seconds)
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }
}

class CSE implements Runnable {
   public void run() {
      while (true) {
         try {
            System.out.println("CSE");
            Thread.sleep(2000); // Sleep for 2000 milliseconds (2 seconds)
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }
}

public class ThreadMain {
   public static void main(String[] args) {
      Thread t1 = new Thread(new BMS_College_of_Engineering());
      Thread t2 = new Thread(new CSE());
      t1.start();
      t2.start();
   }
}
```

LAB: 9

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

```
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 divident:");

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

LAB: 10

Demonstrate Inter process Communication and deadlock.

```
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<5) {
			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<5) {
			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.");
	}
}
```

Deadlock:

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

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

class B {
	synchronized void bar(A a) {
		String name =Thread.currentThread().getName();
		System.out.println(name + " entered B.bar");
		try {
```

```
                Thread.sleep(1000);
            } catch(Exception e) {
                System.out.println("B Interrupted");
            }
            System.out.println(name + " trying to call A.last()");
            a.last();
        }
        void last() {
            System.out.println("Inside A.last");
        }
}

class Deadlock implements Runnable
{
        A a = new A();
        B b = new B();
        Deadlock() {
            Thread.currentThread().setName("MainThread");
            Thread t = new Thread(this,"RacingThread");
            t.start();
            a.foo(b); // 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();
        }

}
```