B.M.S COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

OBJECT ORIENTED JAVA PROGRAMMING

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

SHASHANK PATEL C J
1BM22CS255

Department of Computer Science and Engineering
B.M.S College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019

Sr.No.	Title	Page No.	Sign./Remarks
1	Quadratic equation	10	12/12/23
2	SGPA calculator (19/12/2023)	10	19/12/23
3	Book detail (26/12/2023)	10	26/12/23
4	Abstract class (02/01/2024)	10	2/1/2024
5	Bank account (16/01/2024)	10	16/1/2024
6.	Packages (23/01/2024)	10	23/1/24
7.	Exception handling (30/01/2024)	10	30/1/2024
8.	Multi threading (06/02/2024)	10	6/2/2024
9.	Demonstrate IPC and Deadlock (13/02/2024)		13/2/2024
10.	User interface (Divide App) (20/02/2024)		20/2/2024

Lab

PAGE NO :
DATE : 12/12/2023

(1) * Develop a java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$ read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
class Quadratic
{
    int a,b,c;
    double r1,r2,d;
    void getd()
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the coefficients of a,b,c
        ");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    Void compute()
    {
        while(a==0)
        {
            System.out.println("Not a quadratic equation");
            System.out.println("Enter a non zero value for a:"
            );
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
    }
    d = b*b - 4*a*c;
```

```
if (d == 0)
{
    r1 = (-b)/(2*a);
    System.out.println("Roots are real and equal");
    System.out.println("Root1 = Root2 =" + r1);
}
else if (d > 0)
{
    r1 = ((-b) + (Math.Sqrt(d)))/(double)(2*a);
    r2 = ((-b) - (Math.Sqrt(d)))/(double)(2*a);
    System.out.println("Roots are real and distinct");
    System.out.println("Root1 =" + r1 + "Root2 =" + r2);
}
else if (d < 0)
{
    System.out.println("Roots are imaginary");
    r1 = (-b)/(2*a);
    r2 = Math.Sqrt(-d)/(2*a);
    System.out.println("Root1 =" + r1 + "+" + r2 + "i");
    System.out.println("Root1 =" + r1 + "-" + r2 + "i");
}
}
}

Class QuadraticMain
{
    public static void main(String args[])
    {
        Quadratic qr = new quadratic();
        qr.getd();
        qr.compute();
    }
System.out.println("Shashank Patel CJ IBM 22CS255");
```

PAGE NO :
DATE :

Output:

① enter the coefficients of a, b, c
1 2 1
roots are real and equal
root1 = root2 = -1.0

② enter the coefficients of a, b, c
2 4 5
roots are imaginary
root1 = -1.0 + i 1.2247448713915889
root1 = -1.0 - i 1.2247448713915889

③ enter the coefficients of a, b, c
1 4 1
roots are real and distinct
root1 = -0.2679491924311228
root2 = -3.7320508075688772

Shashank Patel CJ 1BM22CS255

12/12/2023

DATE :

9) 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.

$$SGPA = \frac{\sum [(\text{Credit(course)})(\text{Grade point})]}{\sum (\text{no. of credits})}$$

```
import java.util.*;
class Subject{
    int subjectMarks;
    int credits;
    int grade;
}
class student{
    String name;
    String USn;
    double sgpa;
    Scanner s;
    Subject subject[];
    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 the name:");
    name = s.nextLine();
    System.out.println("enter the USn:");
```

```
USN = S.nextLine();
}
void getMarks(){
    int i;
    for(i=0; i<9; i++)
    {
        System.out.println("enter the marks and
                credits "+i+" : ");
        System.out.println("enter the marks : ");
        int marks = S.nextInt();
        System.out.println("enter the credits : ");
        int credit = S.nextInt();
        Subject[i].SubjectMarks = marks;
        Subject[i].credits = credit;
        Subject[i].grade = (Subject[i].SubjectMarks/
                10)+1;
        if(Subject[i].SubjectMarks == 11)
        {
            Subject[i].SubjectMarks = 10;
        }
        if(Subject[i].SubjectMarks <= 4)
        {
            Subject[i].SubjectMarks = 0;
        }
    }
}
Void computeSGPA(){
    int i;
    int totalCredits = 0;
    int evaluateMarks = 0;
    for(i=0; i<9; i++)
    {
        totalCredits += subject[i].credits;
```

```
        evaluatemarks += subject[i].grade * subject[i].
                         credits;
    }
    sgpa = (float) evaluatemarks / total credits;
    System.out.println("Sgpa = " + sgpa);
  }
}
class main {
    public static void main (String[] args) {
        Student S1 = new Student();
        S1.getStudentDetails();
        S1.getMarks();
        S1.computeSGPA();
    }
}
```

Output:

```
enter the name:
Shashank
enter the USN:
1BM22CS255
enter the marks and credits:
enter the marks
95
enter the credits:
4
enter the marks and credits:
enter the marks
92
enter the credits:
4
enter the marks and credits:
```

PAGE NO :
DATE :

enter the marks:
88
enter the Credits:
3
enter the marks and Credits 3:
enter the marks:
88
enter the credits:
3
enter the marks and credits 4:
enter the marks:
86
enter the Credits:
1
enter the marks and credits 5:
Enter the marks:
88
enter the credits:
3
enter the marks and credits 6:
enter the marks:
78
enter the Credits:
3
enter the marks and Credits 7:
enter the marks:
99
enter the Credits:
1
enter the marks and Credits 8:
enter the marks:
0
enter the Credits:
0

Sgpa = 9.300001907734863
Shashank Patil CJ 1BM22CS255

19/12/2023

19/12/2023

25/10/23 3. Create a class Book which contains four members : name, author, price, num_pages. Include a constructor to set the values for the member. Include methods to set and get the details of the objects. Include a : toString() method that could display the complete details of the book. Develop a java program to create n book objects.

```
import java.util.*;
class 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, author, price, numPages;
        name = "Book name : " + this.name + "\n";
        author = "Author name : " + this.author + "\n";
```

```
        price = "Price: " + this.price + "\n";
        numPages = "Number of pages: " + this.numPages
                + "\n";
        return name + author + price + numPages;
    }
}
class Main3
{
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);
        int n;
        int i;
        int price;
        String name;
        String author;
        int numPages;
        System.out.println("Enter the no. of books");
        n = s.nextInt();
        Book b[];
        b = new Book[n];
        for(i=0; i<n; i++)
        {
        System.out.println("Enter the details of the book: "
                + i);
        System.out.println("Enter the name of the book:");
        name = s.nextLine();
        System.out.println("Enter the author of the book:");
        author = s.nextLine();
        System.out.println("Enter the price of the book:");
        price = s.nextInt();
        System.out.println("Enter the no. of pages of the book:");
        numPages = s.nextInt();
```

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

Output:

```
Enter the no. of books
4
Enter the details of the book:0
Enter the name of the book:
harry
Enter the author of the book:
potter
Enter the price of the book:
500
Enter the no. of pages of the book:
450
Enter the details of the book:1
Enter the name of the book:
Winter
Enter the author of the book:
dan
Enter the price of the book:
900
Enter the no. of pages of the book:
478
```

Enter the details of the book: 2
Enter the name of the book:
Malgudi
Enter the author of the book:
RK
Enter the price of the book:
599
Enter the no. of pages of the book:
567
Enter the details of the book: 3
Enter the name of the book:
Umang
Enter the author of the book:
rock
Enter the price of the book:
749
Enter the no. of pages of the book:
467
Book details:
Book name: harry
Author name: potter
Price : 500
Number of pages: 450

Book name: Winter
Author name: Dan
Price : 900
Number of pages: 478

Book name: Malgudi
Author name: RK
price : 599
Number of pages: 567

PAGE NO :
DATE :

Book name : Sumanji
Author name : Jack
Price : 799
Number of pages : 167
Shashank Patil C.J. 1BM22CS255

23/12/23

PAGE NO
DATE: 02/01/2024

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.

```
import java.util.*;
class InputScanner{
    protected Scanner s;
    public InputScanner()
    {
        s = new Scanner(System.in);
    }
    public double getDoubleInput(String message)
    {
        System.out.println(message);
        return s.nextDouble();
    }
    public void closeScanner()
    {
        s.close();
    }
}
abstract class Shape extends InputScanner
{
    protected double dim1;
    protected double dim2;
    Shape()
    {
```

```
        Super();
    }
    public abstract void printArea();
}
class Rectangle extends Shape
{
    public Rectangle()
    {
        Super();
    }
    public void printArea()
    {
        dim1 = getDoubleInput("enter length of the
                               rectangle");
        dim2 = getDoubleInput("enter breadth of the
                               rectangle");
        double area = dim1 * dim2;
        System.out.println("area of the rectangle:"
                           + area);
    }
}
class Triangle extends Shape
{
    public Triangle()
    {
        Super();
    }
    public void printArea()
    {
        dim1 = getDoubleInput("enter base of the
                               triangle");
        dim2 = getDoubleInput("enter height of the
                               triangle");
```

```
        double area = 0.5 * dim1 * dim2;
        System.out.println("area of the Triangle: "+area);
    }
}
class Circle extends Shape
{
    public Circle()
    {
        super();
    }
    public void printArea()
    {
        dim1 = getDoubleInput("enter radius of the circle");
        double area = Math.PI * dim1 * dim1;
        System.out.println("area of the circle: "+area);
    }
}
public class Area
{
    public static void main(String[] args)
    {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();
        r.printArea();
        t.printArea();
        c.printArea();
        r.closeScanner();
    }
}
```

PAGE NO:
DATE:

Output:-

```
enter length of the rectangle
5
enter breadth of the rectangle
4
area of the rectangle : 20.0
enter base of the triangle
3
enter height of the triangle
4
area of the Triangle : 6.0
enter radius of the circle
5
area of the circle: 78.53981633974483
```

Shashank Patel C J 1BM22CS255

2/1/2024

PAGE NO.
DATE 07/01/2021

5. Develop a java program to create a class Bank account that maintains two kinds of account for its customers, i.e., Savings and current using inheritance concept by defining display(), deposit() and withdraw() methods in the account class.

```
import java.util.*;
class account
{
    String name;
    int accno;
    String type;
    double balance;
    account(String name, int accno, String type, double
        balance)
    {
        this.name = name;
        this.accno = accno;
        this.type = type;
        this.balance = balance;
    }
    void deposit(double amount)
    {
        balance += amount;
    }
    void withdraw(double amount)
    {
        if((balance - amount) >= 0)
        {
            balance -= amount;
        }
```

```
        else
        {
            System.out.println("Insufficient balance to withdraw");
        }
    }
    Void display()
    {
        System.out.println("name:"+name+"accno:"
                +accno+"type:"+type+
                "balance"+balance);
    }
}

Class Savings extends account
{
    private static double rate=10;
    private static double time=1;
    Savings(String name, int accno, String type, double
            balance)
    {
        super(name, accno, "Savings", balance);
    }
    Void interest()
    {
        balance+=balance * Java.lang.Math.pow
                (1+rate/100), time);
        System.out.println("balance="+balance);
    }
}

Class current extends account
{
    private static double minbal=1000;
```

```
    private static double service_taxes = 50;
    current (String name, int accno; String type, double
        balance)
    {
        super (name, accno, "current", balance);
    }
    void checkMin()
    {
        if (balance < minbal)
        {
            System.out.println ("balance is low service
            taxes are added" + service_taxes);
            balance -= service_taxes;
            System.out.println("balance =" + balance);
        }
    }

public class accountmain
{
    public static void main (String[] args)
    {
        Scanner S = new Scanner (System.in);
        System.out.println ("enter the name :");
        String name = S.nextLine();
        System.out.println ("enter the type of account :");
        String type = S.nextLine();
        System.out.println ("enter the accno:");
        int accno = S.nextInt();
        System.out.println("enter the initial balance :");
        double balance = S.nextDouble();
        int ch;
        double amount1, amount2;
```

```
account ao = new account(name, accno, type, balance);
Savings so = new Savings(name, accno, type, balance);
current co = new current(name, accno, type, balance);
while(true)
{
    if(ao.type.equals("Savings"))
    {
        System.out.println("Enter 1. deposit 2. withdraw
                            3. interest 4. display");
        System.out.println("enter the choice");
        ch = s.nextint();
        switch(ch)
        {
            case 1: System.out.println("enter the
                    amount to be deposited");
                    amount1 = s.nextDouble();
                    so.deposit(amount1);
                    break;
            case 2: System.out.println("enter the
                    amount to be withdrawn");
                    amount2 = s.nextdouble();
                    so.withdraw(amount2);
                    break;
            case 3: so.interest();
                    break;
            case 4: so.Display();
                    break;
            case 5: System.exit(0);
            default: System.out.println("Invalid input");
                    break;
        }
    }
}
```

```
else
{
    System.out.println("enter 1. deposit 2. Withdraw
                        3. display");
    System.out.println("enter the choice");
    ch = S.nextInt();
    switch(ch)
    {
        case 1: System.out.println("enter the amount
                    to be deposited");
                amount1 = S.nextDouble();
                CO.deposit(amount1);
                break;
        case 2: System.out.println("enter the
                amount to be withdrawn");
                amount2 = S.nextdouble();
                CO.checkmin();
                CO.withdraw(amount2);
                break;
        case 3: CO.display();
                break;
        case 4: System.exit(0);
        default: System.out.println("invalid input");
                break;
    }
}
}
}
```

Output 1

savings

enter the name:
S
enter the type of account:
savings
enter the accno:
1
enter the initial balance:
500
enter 1. deposit 2. withdraw 3. interest 4. display
enter the choice
1
enter the amount to be deposited
500
enter 1. deposit 2. withdraw 3. interest 4. display
enter the choice
4
name: S accno: 1 type: savings balance 1000.0
enter 1. deposit 2. withdraw 3. interest 4. display
enter the choice
2
enter the amount to be withdrawn
500
enter 1. deposit 2. withdraw 3. interest 4. display
enter the choice
4
name: S accno: 1 type: savings balance 500.0
enter 1. deposit 2. withdraw 3. interest 4. display
enter the choice
3
balance = 950.0

enter 1: deposit 2. withdraw 3. Interest 4. display
enter the choice
4
name: S accno: 1 type: savings balance: 950.0
enter 1. deposit 2. withdraw 3. Interest 4. display
enter the choice
5
current a/c -
enter the name:
S
enter the type of account:
current
enter the accno:
1
enter the initial balance:
500
enter 1. deposit 2. withdraw 3. display
enter the choice
3
name: S accno: 1 type: current balance: 500.0
enter 1. deposit 2. withdraw 3. display
enter the choice
1
enter the amount to be deposited
500
enter 1. deposit 2. withdraw 3. display
enter the choice
2
enter the amount to be withdrawn
1000
enter 1. deposit 2. withdraw 3. display
enter the choice
3

PAGE NO :
DATE :

name: S acno: 1 type: current balance: 0.0
enter 1. deposit 2. withdraw 3. display
enter the choice
2
enter the amount to be withdrawn
500
balance in low sufficient are added 50.0
balance = -50.0
insufficient balance to withdraw
enter 1. deposit 2. withdraw 3. display
enter the choice
4

16/1/2024 Shashank Patel CJ 1BM22CS255

Lab-6

PAGE NO:
DATE 22/01/24

6) Develop a java program and create a CIE package consists of Student and Internal class and SEE package consists of Externals and main class and display the Students marks in Internals and externals.

```
//Student.Java

package CIE;
import java.util.Scanner;
public class Student
{
    protected String USN = new String();
    protected String name = new String();
    protected int Sem;
    public void Input Student details()
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("enter USN:");
        USN = scanner.nextLine();
        System.out.println("enter Name:");
        name = Scanner.nextLine();
        System.out.println("enter Semester:");
        Sem = Scanner.nextInt();
    }
    public void display Student Details()
    {
        System.out.println("USN:" + USN);
        System.out.println("Name:" + name);
        System.out.println("Sem:" + Sem);
    }
}
```

PAGE NO :
DATE :

```
// Internals.java
package CIE;
import java.util.Scanner;
public class Internals extends Student
{
    protected int marks[] = new int[5];
    public void inputCIEMarks()
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the internal marks
                            for " + name);
        for(int i=0; i<5; i++)
        {
            System.out.println("Subject " + (i+1) +
                               " marks: ");
            marks[i] = scanner.nextInt();
        }
    }
}

// External.java

package SEE;
import CIE.Internals;
import java.util.Scanner;
public class External extends Internals {
    protected int marks[];
    protected int final_marks[];
    public External()
    {
        marks = new int[5];
        final_marks = new int[5];
    }
```

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

// Main.java

import SEE.Externals;

public class Main {
    public static void main(String[] args){
        int num_of_students = 2;
        Externals finalmarks[] = new Externals
                                 [num_of_students];
        for(int i=0; i< num_of_students; i++){
            finalmarks[i] = new externals();
```

```
final mark[i].InputStudentDetails();
System.out.println("Enter CIE marks:");
final mark[i].InputCIEMarks();
System.out.println("Enter SEE marks:");
final mark[i].InputSEEmarks();
}

System.out.println("Display data:");
for(int i=0; i<num_of_students; i++){
        finalmark[i].CalculateFinalMarks();
        final mark[i].displayFinalMarks();
}
}
}
```

Output:

```
Enter USN:
1BM22CS255
enter name:
Shashank Patel C.J.
enter Sem
3
enter CIE marks:
enter external marks:
Subject 1: 45
Subject 2: 46
Subject 3: 42
Subject 4: 43
Subject 5: 44
enter SEE marks:
Enter SEE marks for Shashank Patel C.J.
Subject 1: 85
```

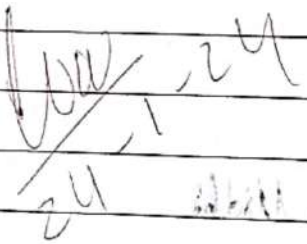
Subject 2 : 89
Subject 3 : 87
Subject 4 : 88
Subject 5 : 87

Enter USN : 1BM22CS485
Enter name : Nidhi
Enter Semester : 3
Enter CIE Marks
Enter Internal marks for Nidhi :
Subject 1 : 45
Subject 2 : 47
Subject 3 : 40
Subject 4 : 43
Subject 5 : 44
Enter SEE marks
Enter SEE marks for Nidhi
Subject 1 : 90
Subject 2 : 78
Subject 3 : 79
Subject 4 : 89
Subject 5 : 80

Display data :
USN : 1BM22CS255
Name : Shashank Patel C.J.
Sem : 3
Subject 1 : 87
Subject 2 : 90
Subject 3 : 85
Subject 4 : 87
Subject 5 : 87

USN : 1BM22CS485
Name : Nidhi
Sem : 3
Subject 1 : 90
Subject 2 : 86
Subject 3 : 76
Subject 4 : 86
Subject 5 : 75

Shashank Patil. C.J. 1BM22CS255



LAB-7

PAGE NO:
DATE: 30/07/24

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 calls both father and son's age and throws an exception if son's age is >= Father's age.

```
import java.util.Scanner;
class Wrongage extends Exception
{
    public WrongAge(String meltage)
    {
        super(meltage);
    }
}
class InputScanner
{
    protected Scanner s;
    public InputScanner()
    {
        s = new Scanner(System.in);
    }
}
class Father extends InputScanner
{
    protected int FatherAge;
    public Father() throws WrongAge
    {
        System.out.println("enter Father's age:");
```

```
        FatherAge = S.nextInt();
        if (FatherAge<0)
        {
            throw new WrongAge("Age cannot be
                                negative");
        }
    }
    public void display()
    {
        System.out.println("Father's age : "+FatherAge);
    }
}
class Son extends Father
{
    protected int SonAge;
    public Son() throws WrongAge
    {
        System.out.println("enter Son's age:");
        SonAge = S.nextInt();
        if(SonAge >= FatherAge)
        {
            throw new WrongAge("Son's age
            can't be greater than or equal
            to father's age");
        }
        if (SonAge<0)
        {
            throw new WrongAge("Age cannot be
                                negative");
        }
    }
    public void display()
    {
```

```
        super.display();
        System.out.println("Son's Age: "+SonAge);
    }
}

public class Main2
{
    public static void main(String args[])
    {
        try
        {
            Son son = new Son();
            son.display();
        }
        catch(WrongAge e)
        {
            System.out.println("Error: "+e.getMessage());
        }
    }
}
```

Output :-

```
Enter Father's age:
25
Enter Son's age:
25
Error: Son's age can't be greater than or equal to
       father's age.

Enter Father's age:
-35
Error: Age cannot be negative
```

PAGE NO :
DATE :

enter Father's Age :
34
enter Son's age :
-9
Error: Age cannot be negative

enter Father's age :
35
enter Son's age :
13
Father's age : 35
Son's age : 13

Shashank Patil C.J. 1BM22CS255

35 1/12/2024

PAGE NO :
DATE :

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 Thread1 implements Runnable
{
    public void run()
    {
        for(int i=0; i<5; i++)
        {
            try
            {
                System.out.println("BMS College of
                Engineering");
                Thread.sleep(10000);
            }
            catch(InterruptedException ie)
            {
                System.out.println("The sleping
                thread is woken up");
            }
        }
    }
}

class Thread2 implements Runnable
{
    public void run()
    {
        for (int j=0; j<5; j++)
        {
```

```
for(int i=0; i<5; i++)
{
    try
    {
        System.out.println("CSE");
        Thread.sleep(2000);
    }
    catch (InterruptedException ie)
    {
        System.out.println("The Sleeping thread
        is woken up");
    }
}
}

class Main21
{
    public static void main(String[] args)
    {
        Thread t1 = new Thread(new thread1());
        Thread t2 = new Thread(new thread2());
        t1.start();
        t2.start();
    }
}
```

Output:-

BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE

PAGE NO :
DATE :

BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE

6/2/2024

PAGE NO:
DATE: 20/02/2024

10) 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 the Result field when the divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

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

class SwingDemo{
    // Create a frame container
    JFrame jfrm = new JFrame("Divide App");
    private JFrame jfrm;
    private JTextField ajtf, bjtf;
    private JLabel alab, blab, anslab, err;

    public SwingDemo(){
        jfrm = new JFrame("Divide App");
        jfrm.setSize(275,200);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_
                ON_CLOSE);
        JLabel jlab = new JLabel("Enter the
```

```
divisor and dividend: ");
aftf = new JTextField(8);
bftf = new JTextField(8);
JButton button = new JButton("calculate");
err = new JLabel();
alab = new JLabel();
blab = new JLabel();
anslab = new JLabel();

frm.add(err);
frm.add(alab);
frm.add(aftf);
frm.add(bftf);
frm.add(button);
frm.add(alab);
frm.add(blab);
frm.add(anslab);

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt){
        calculateDivision();
    }
});
frm.setVisible(true);
}

private void calculateDivision() {
    try {
        int a = Integer.parseInt(aftf.getText());
        int b = Integer.parseInt(bftf.getText());
        if (b==0) {
            throw new ArithmeticException("Division by
                zero!");
        }
```

```
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("B should be non-zero!");
}
}
public static void main(String args[]){
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
}
```

Output:

Enter the divider and dividend

[2] [1]

[Calculate] A = 2 B = 1 Ans = 2.0

Functions :-

1) JFrame :- It is a top level container in Java String that represents a window with a title bar, border and optional menubar.

2) SetSize :- It is used to set size of the frame.

3) SetLayout :- This line sets the layout manager for the frame to FlowLayout. Which arranges components from left to right in a flow like manner.

4) add :- This line adds the JLabel to the frame.

5) SetVisible :- This line makes the frame visible.

20/2/2024

10. Demonstrate inter process communication and deadlock.

```
class Q {
int n;
boolean valueSet = false;
synchronized int get() {
while(!valueSet)
try {
System.out.println("\n Consumer Waiting\n");
wait();
} catch(InterruptedException e)
{
System.out.println("Interrupted Exception 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<6){
q.put(i++);
}
}
}

class Consumer implements Runnable {
Q q;
Consumer(Q q){
this.q = q;
new Thread(this, "Consumer").start();
}
public void run(){
        int i=0;
while(i<6){
int r = q.get();
System.out.println("Consumed:"+r);
i++;
}
}
}
```

```
class PCFixed {
public static void main(String args[])
{
    Q q = new Q();
    new Producer(q);
    new Consumer(q);
    System.out.println("Press Control-C to Stop.");
}
}
```

Output:

```
Put: 0
Intimate Consumer
Producer Waiting
Press Control-C to Stop
Got: 0
Intimate Producer
Consumed: 0
Put: 1
Intimate Consumer
Producer Waiting
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
Producer waiting
Got: 4
Intimate producer
Consumed: 4
Put: 5
Intimate Consumer
~~producer waiting~~
Got: 5
Intimate Producer
consumed: 5

Shashank Patel C.J., IBM22CS255

5)

```
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("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");
  }
  public static void main(String args[])
  {
    new Deadlock();
  }
}
```

Output:

Main Thread entered A.foo
Racing Thread entered B.bar
Main Thread trying to call B.last()
Inside A.last
Back in main thread
Racing Thread trying to call A.last()
Inside A.last
Back in other thread

Shashank Patil C.J. 1BM22CS255

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

```
import java.util.Scanner;

class Quadratic

{

   int a,b,c;

   double r1,r2,d;

   void getd()

   {

     Scanner s=new Scanner(System.in);

     System.out.println("enter the coefficients of a,b,c");

     a=s.nextInt();

     b=s.nextInt();

     c=s.nextInt();

   }

   void compute()

   {

     while(a==0)

     {

          System.out.println("not a quadratic equation");

          System.out.println("enter a non zero value for a:");

       Scanner s=new Scanner(System.in);

          a=s.nextInt();

        }

        d=b*b-4*a*c;

        if(d==0)
```

```
        {
          r1=(-b)/(2*a);
          System.out.println("roots are real and equal");
       System.out.println("root1=root2="+r1);
        }
        else if(d>0)
        {
          r1=((-b)+(Math.sqrt(d)))/(double)(2*a);
          r2=((-b)-(Math.sqrt(d)))/(double)(2*a);
          System.out.println("roots are real and distinct");
          System.out.println("root1="+r1+"root2="+r2);
        }
        else if(d<0)
        {
          System.out.println("roots are imaginary");
      r1=(-b)/(2*a);
          r2=Math.sqrt(-d)/(2*a);
          System.out.println("root1="+r1+"+i"+r2);
          System.out.println("root1="+r1+"-i"+r2);
        }
    }
}
class QuadraticMain
{
    public static void main(String args[])
        {
```

```
        Quadratic q=new Quadratic();
        q.getd();
        q.compute();
        System.out.println("Shashank Patel C J USN-1BM22CS255");
      }
}
```

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

```
import java.util.Scanner;
class Subject{
int subjectMarks;
int credits;
int grades;
}
class Student{
Subject subject[];
String name;
String usn;
double SGPA;
Scanner s;
Student(){
subject = new Subject[9];
for(int i = 0;i<9;i++){
subject[i] = new Subject();
}
```

```
s= new Scanner(System.in);
}
void getStudentDetails(){
System.out.println("Enter your name: ");
this.name = s.nextLine();
System.out.println("Enter your usn: ");
this.usn = s.next();
}
void getMarks(){
for(int i=0;i<8;i++){
System.out.println("Enter the marks of the "+(i+1)+" subject");
subject[i].subjectMarks = s.nextInt();
System.out.println("Enter the credits of the "+(i+1)+"subject");
subject[i].credits = s.nextInt();
subject[i].grades = (subject[i].subjectMarks/10)+1;
if(subject[i].grades >10)
subject[i].grades = 10;
if(subject[i].grades <4)
subject[i].grades = 0;
}
}
void computeSGPA(){
int sum=0;
int totalCredits=0;
for(int i=0;i<9;i++){
sum+=(subject[i].grades*subject[i].credits);
```

```
totalCredits+=subject[i].credits;
}
this.SGPA=(double)sum/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);
System.out.println("Shashank Patel C J USN-1BM22CS255");
}
}
```

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.*;
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,author,price,numPages;
        name="Book name:"+this.name+"\n";
        author="Author name:"+this.author+"\n";
        price="Price:"+this.price+"\n";
        numPages="Number of pages:"+this.numPages+"\n";
        return name+author+price+numPages;
    }

}
class Main3
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        int n;
        int i;
        String name;
```

```
String author;
int price;
int numPages;
System.out.println("Enter the no. of books\n");
n=s.nextInt();
Book b[];
b=new Book[n];
for(i=0;i<n;i++)
{

   System.out.println("Enter the details of the book:"+i);
   System.out.println("Enter the name of the book:");
   name=s.nextLine();
   System.out.println("Enter the author of the book:");
   author=s.nextLine();
   System.out.println("Enter the price of the book:");
   price=s.nextInt();
   System.out.println("Enter the no. of pages of the book:");
   numPages=s.nextInt();
   b[i]=new Book(name,author,price,numPages);
}
System.out.println("Book details:");
for(i=0;i<n;i++)
{
   System.out.println(b[i].toString());
}
```

```
        System.out.println("Shashank Patel C J USN-1BM22CS255");
    }
}
```

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.*;
class inputScanner{
    protected Scanner s;
    public inputScanner()
    {
        s=new Scanner(System.in);
    }
    public double getDoubleinput(String message)
    {
        System.out.println(message);
        return s.nextDouble();
    }
    public void closeScanner()
    {
        s.close();
    }
}
abstract class Shape extends inputScanner
{
    protected double dim1;
```

```
    protected double dim2;
    Shape()
    {
       super();
    }
    public abstract void printArea();
}
class Rectangle extends Shape
{
    public Rectangle()
    {
       super();
    }
    public void printArea()
    {
       dim1=getDoubleinput("enter length of the rectangle");
       dim2=getDoubleinput("enter breadth of the rectangle");
       double area=dim1*dim2;
       System.out.println("area of the rectangle:"+area);
    }
}
class Triangle extends Shape
{
    public Triangle()
    {
       super();
```

```
    }
    public void printArea()
    {
        dim1=getDoubleinput("enter base of the triangle");
        dim2=getDoubleinput("enter height of the triangle");
        double area=0.5*dim1*dim2;
        System.out.println("area of the Triangle:"+area);
    }
}
class Circle extends Shape
{
    public Circle()
    {
        super();
    }
    public void printArea()
    {
        dim1=getDoubleinput("enter radius of the circle");
        double area=Math.PI*dim1*dim1;
        System.out.println("area of the circle:"+area);
    }
}
public class Area
{
    public static void main(String[] args)
    {
```

```
        Rectangle r=new Rectangle();
        Triangle t=new Triangle();
        Circle c=new Circle();
        r.printArea();
        t.printArea();
        c.printArea();
        r.closeScanner();
        System.out.println("Shashank Patel C J USN-1BM22CS255");
    }
}
```

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

- Create a class Account that stores customer name, account number and type of account.From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- a)Accept deposit from customer and update the balance.
- b)Display the balance.
- c)Compute and deposit interest
- d)Permit withdrawal and update the balance
- Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.*;
class account
{
   String name;
```

```
int accno;
String type;
double balance;
account(String name,int accno,String type,double balance)
{
   this.name=name;
   this.accno=accno;
   this.type=type;
   this.balance=balance;
}
void deposit(double amount)
{
   balance+=amount;
}
void withdraw(double amount)
{
   if((balance-amount)>=0)
   {
      balance-=amount;
   }
   else
   {
      System.out.println("insufficient balance to withdraw");
   }
}
void display()
```

```
    {
        System.out.println("name:"+name+"accno:"+accno+"type:"+type+"balance"+balance);
    }
}
class savings extends account
{
    private static double rate=10;
    private static double time=1;
    savings(String name,int accno,String type,double balance)
    {
        super(name,accno,"savings",balance);
    }
    void interest()
    {
        balance+=balance*java.lang.Math.pow(1-(rate/100),time);
        System.out.println("balance="+balance);
    }
}
class current extends account
{
    private static double minbal=1000;
    private static double service_taxes=50;
    current(String name,int accno,String type,double balance)
    {
        super(name,accno,"current",balance);
    }
```

```
  void checkmin()
  {
    if(balance<minbal)
    {
      System.out.println("balance is low service taxes are added"+service_taxes);
      balance-=service_taxes;
      System.out.println("balance="+balance);
    }
  }
}
public class accountmain
{
  public static void main(String[] args)
  {
    Scanner s=new Scanner(System.in);
    System.out.println("enter the name:");
    String name=s.nextLine();
    System.out.println("enter the type of account:");
    String type=s.nextLine();
    System.out.println("enter the accno:");
    int accno=s.nextInt();
    System.out.println("enter the initial balance:");
    double balance=s.nextDouble();
    int ch;
    double amount1,amount2;
    account a0=new account(name,accno,type,balance);
```

```
savings s0=new savings(name,accno,type,balance);
current c0=new current(name,accno,type,balance);
while(true)
{
  if(a0.type.equals("savings"))
  {
    System.out.println("enter 1.deposit 2.withdraw 3.interest 4.display");
    System.out.println("enter the choice");
    ch=s.nextInt();
    switch(ch)
    {
      case 1:System.out.println("enter the amount to be deposited");
          amount1=s.nextDouble();
          s0.deposit(amount1);
          break;
      case 2:System.out.println("enter the amount to be withdrawn");
          amount2=s.nextDouble();
          s0.withdraw(amount2);
          break;
      case 3:s0.interest();
          break;
      case 4:s0.display();
          break;
      case 5:System.exit(0);
      default:System.out.println("invalid input");
          break;
```

```
    }
  }
  else
  {
    System.out.println("enter 1.deposit 2.withdraw 3.display");
    System.out.println("enter the choice");
    ch=s.nextInt();
    switch(ch)
    {
      case 1:System.out.println("enter the amount to be deposited");
          amount1=s.nextDouble();
          c0.deposit(amount1);
          break;
      case 2:System.out.println("enter the amount to be withdrawn");
          amount2=s.nextDouble();
          c0.checkmin();
          c0.withdraw(amount2);
          break;
      case 3:c0.display();
          break;
      case 4:System.exit(0);
      default:System.out.println("invalid input");
           break;
    }
  }
}
```

```
        System.out.println("Shashank Patel C J USN-1BM22CS255");
    }
}
```

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 {

protected String usn = new String();

protected String name = new String();

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 sem");

sem=s.nextInt();

}

public void displayStudentDetails()

{

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

```
    System.out.println("Name"+name);
    System.out.println("Sem"+sem);
}
}
package CIE;
import java.util.Scanner;
public class Internals extends Student {
protected int marks[] = new int[5];
public void inputCIEmarks()
{
    Scanner scanner=new Scanner(System.in);
    System.out.println("enter the internal marks for"+name);
    for(i=0;i<5;i++)
    {
        System.out.println("SUbject"(i+1)+"marks:");
        marks[i]=scanner.nextInt();
    }
}
}
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.print("Subject "+(i+1)+" marks: ");
marks[i] = s.nextInt();
 } }
public void calculateFinalMarks() {
for(int i=0;i<5;i++)
finalMarks[i] = marks[i]/2 + super.marks[i];
}
public void displayFinalMarks() {
displayStudentDetails();
for(int i=0;i<5;i++)
System.out.println("Subject " + (i+1) + ": " + finalMarks[i]);
}
}
import SEE.Externals;
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();
}
System.out.println("Shashank Patel C J USN-1BM22CS255");
}
}
```

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

```
import java.util.Scanner;
class WrongAge extends Exception
{
   public WrongAge(String message)
   {
      super(message);
```

```
    }
}
class InputScanner
{
    protected Scanner s;
    public InputScanner()
    {
        s=new Scanner(System.in);
    }
}
class Father extends InputScanner
{
    protected int FatherAge;
    public Father() throws WrongAge
    {
        System.out.println("enter Father's age:");
        FatherAge=s.nextInt();
        if(FatherAge<0)
        {
            throw new WrongAge("Age cannot be negative");
        }
    }
     public void display()
    {
          System.out.println("Father's age:"+FatherAge);
    }
```

```
}
class Son extends Father
{
   protected int SonAge;
   public Son() throws WrongAge
   {
      System.out.println("enter Son's age:");
      SonAge=s.nextInt();
      if(SonAge>FatherAge)
      {
         throw new WrongAge("Son's age can't be greater than father's age");
      }
      if(SonAge==FatherAge)
      {
         throw new WrongAge("Son's age and Father's age can't be same");
      }
      if(SonAge<0)
      {
         throw new WrongAge("Age cannot be negative");
      }
   }
   public void display()
   {
      super.display();
      System.out.println("Son's age:"+SonAge);
   }
```

```
}
public class Main2
{
   public static void main(String args[])
   {
      try
      {
         Son son=new Son();
         son.display();
      }
      catch(WrongAge e)
      {
         System.out.println("Error:"+e.getMessage());
      }
     System.out.println("Shashank Patel C J USN-1BM22CS255");
   }
}
```

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 thread1 implements Runnable
{
   public void run()
   {
      for(int i=0;i<5;i++){
      try
      {
```

```
            System.out.println("BMS College Of Engineering");
            Thread.sleep(10000);
        }
        catch(InterruptedException ie)
        {
            System.out.println("The sleeping thread is woken up");
        }
      }
    }
}
class thread2 implements Runnable
{
    public void run()
    {
        for(int j=0;j<5;j++){
        for(int i=0;i<5;i++){
        try
        {
            System.out.println("CSE");
            Thread.sleep(2000);
        }
        catch(InterruptedException ie)
        {
            System.out.println("The sleeping thread is woken up");
        }
    }
```

```
    }
    }
}
class Main21
{
    public static void main(String[] args)
    {
        Thread t1=new Thread(new thread1());
        Thread t2=new Thread(new thread2());
        t1.start();
        t2.start();
        System.out.println("Shashank Patel C J USN-1BM22CS255");
    }
}
```

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.*;

public class SwingDemo {
    private JFrame jfrm;
    private JTextField ajtf, bjtf;
    private JLabel alab, blab, anslab, err;
```

```
public SwingDemo() {
  jfrm = new JFrame("Divider App");
  jfrm.setSize(275, 200);
  jfrm.setLayout(new FlowLayout());
  jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  JLabel jlab = new JLabel("Enter the divider and dividend:");
  ajtf = new JTextField(8);
  bjtf = new JTextField(8);
  JButton button = new JButton("Calculate");
  err = new JLabel();
  alab = new JLabel();
  blab = new JLabel();
  anslab = new JLabel();
  jfrm.add(err);
  jfrm.add(jlab);
  jfrm.add(ajtf);
  jfrm.add(bjtf);
  jfrm.add(button);
  jfrm.add(alab);
  jfrm.add(blab);
  jfrm.add(anslab);
  button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
      calculateDivision();
    }
  });
```

```
    jfrm.setVisible(true);
}
private void calculateDivision() {
    try {
        int a = Integer.parseInt(ajtf.getText());
        int b = Integer.parseInt(bjtf.getText());
        if (b == 0) {
            throw new ArithmeticException("Division by zero!");
        }
        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!");
    } catch (ArithmeticException e) {
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("B should be non-zero!");
    }
}
```

```
    public static void main(String args[]) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new SwingDemo();
            }
        });
        System.out.println("Shashank Patel C J USN-1BM22CS255");
    }
}
```

10.a)Demonstrate 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++;
}
}
}
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.");
```

```
 System.out.println("Shashank Patel C J USN-1BM22CS255");
}
}
```

b).Demonstrate 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();
        System.out.println("Shashank Patel C J USN-1BM22CS255");
    }
}
```