

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
**on**

## **Object Oriented Java Programming** **(23CS3PCOOJ)**

*Submitted by*

**Ariz Ejaz Khan (1BM23CS051)**

*in partial fulfillment for the award of the degree of*  
**BACHELOR OF ENGINEERING**  
*in*  
**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**  
(Autonomous Institution under VTU)

**BENGALURU-560019**  
**Sep-2024 to Jan-2025**

**B.M.S. College of Engineering,**  
**Bull Temple Road, Bangalore 560019**  
(Affiliated To Visvesvaraya Technological University, Belgaum)  
**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **Ariz Ejaz Khan (1BM23CS051)**, who is Bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Dr. Prasad G R Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
---	---

## Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	09-10-24	Quadratic Equation	4-6
2	16-10-24	Student Class	7-11
3	16-10-24	Class Book	12-14
4	23-10-24	Abstract class Shape	15-17
5	30-10-24	Bank class	18-22
6	13-11-24	Packages CIE and SEE	23-27
7	20-11-24	Exceptions in inheritance tree	28-30
8	27-11-24	Threads	31-32
9	27-11-24	User Interface for integer divisions	33-37
10	27-11-24	IPC and Deadlock	38-48

Github Link:

[https://github.com/Arizbms/Java\\_Lab](https://github.com/Arizbms/Java_Lab)

### Program 1

Implement Quadratic Equation.

Algorithm:

The image shows two pages of handwritten code and output for a Java program. The left page contains the code for the 'quadratic' class, and the right page contains the output of the program.

```
9-10-24
Week-1

Q. 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 discriminate is negative, display a message stating that there are no real solutions.

import java.util.*;

class quadratic {
    public static void main (String args[]) {
        double a, b, c;
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter a, b, c:");
        a = s.nextDouble ();
        b = s.nextDouble ();
        c = s.nextDouble ();
        if (a == 0) {
            System.out.println ("Not Quadratic");
        }
        else {
            double d = b*b - 4*a*c;
            if (d == 0) {
                double x = -b / (2*a);
                System.out.println ("Equal roots. x1 = x2 = " + x);
            }
            else if (d > 0) {
                double x1 = (-b + (Math.sqrt (d))) / (2*a);
                double x2 = (-b - (Math.sqrt (d))) / (2*a);
                System.out.println ("x1 = " + x1 + " " + "x2 = " + x2);
            }
            else {
                System.out.println ("Roots are imaginary");
            }
        }
        System.out.println ("Name: Ariz Ejaz Khan USN: 1BM23CS051");
    }
}
```

O/p

```
Enter a, b, c:
1
3
2
x1 = -1.0, x2 = -2.0
Name: Ariz Ejaz Khan USN: 1BM23CS051
```

Code:

```
import java.util.*;

class quadratic{
    public static void main(String args[]){
        float a,b,c;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter a,b,c:");
        a=s.nextFloat();
        b=s.nextFloat();
```

```

c=s.nextFloat();
if(a==0){
System.out.println("Not quadratic");
}
else{
float d=b*b-4*a*c;
if(d==0){
float r=-b/(2*a);
System.out.println("Equal roots.r1=r2="+r);
}
else if(d>0){
float r1=(-b)+(Math.sqrt(d))/(2*a);
float r2=(-b)-(Math.sqrt(d))/(2*a);
System.out.println("r1="+r1+" "+"r2="+r2);
}
else{
System.out.println("Roots are imaginary");
}

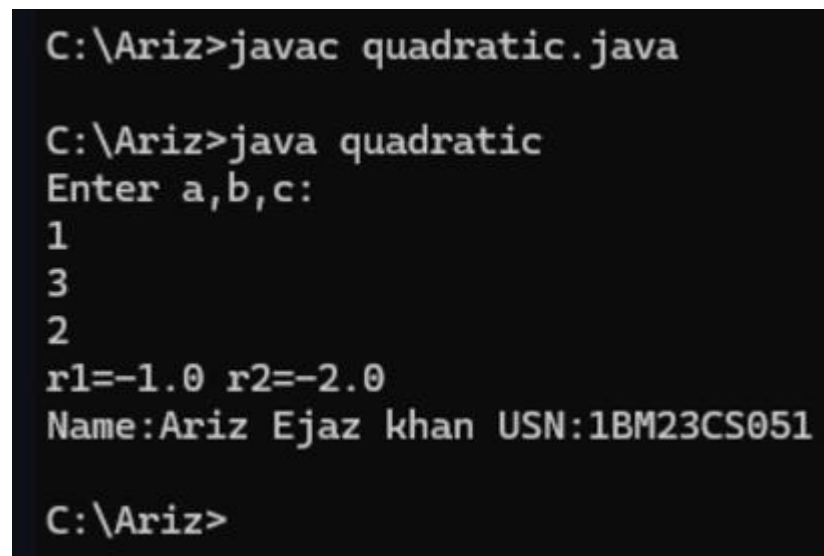
}

System.out.println("Name:Ariz Ejaz Khan USN:1BM23CS051");

}

}

```



```

C:\Ariz>javac quadratic.java

C:\Ariz>java quadratic
Enter a,b,c:
1
3
2
r1=-1.0 r2=-2.0
Name:Ariz Ejaz khan USN:1BM23CS051

C:\Ariz>

```

```
C:\Ariz>javac quadratic.java

C:\Ariz>java quadratic
Enter a,b,c:
4
-4
1
Equal roots.r1=r2=0.5
Name:Ariz Ejaz khan USN:1BM23CS051

C:\Ariz>
```

```
C:\Ariz>javac quadratic.java

C:\Ariz>java quadratic
Enter a,b,c:
-1
-10
-30
Roots are imaginary
Name:Ariz Ejaz khan USN:1BM23CS051

C:\Ariz>
```

## Program 2

Implement Student class with display details and SGPA function.

Algorithm:

```
16-10-24
Week 2
07 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 student.
import java.util.*;
class Std-details {
    int marks[] = new int[8];
    int credit[] = new int[8];
    String usn, name;
    Scanner sc = new Scanner(System.in);

    void getdetails() {
        System.out.println("Enter the usn and name");
        usn = sc.next();
        name = sc.next();
        System.out.println("Enter marks");
        for (int i = 0; i < 8; i++) {
            marks[i] = sc.nextInt();
        }
        System.out.println("Enter the credits");
        for (int i = 0; i < 8; i++) {
            credit[i] = sc.nextInt();
        }
    }

    void display() {
        System.out.println("usn: " + usn);
        System.out.println("name: " + name);
        for (int i = 0; i < 8; i++) {
            System.out.println("marks of " + (i+1) + " subject"
                               + " " + marks[i]);
        }
        System.out.println("SGPA: " + calculateSGPA());
    }

    double getgradePoint(int mark) {
        if (mark >= 90) return 10.0;
        else if (mark >= 80) return 9.0;
        else if (mark >= 70) return 8.0;
        else if (mark >= 60) return 7.0;
        else if (mark >= 50) return 6.0;
        else if (mark >= 40) return 5.0;
        else return 0.0;
    }

    double calculateSGPA() {
        int totalCredits = 0;
        double gradePoint = 0;
        for (int i = 0; i < 8; i++) {
            totalCredits += credit[i];
        }
        for (int i = 0; i < 8; i++) {
            gradePoint += getgradePoint(marks[i]) * credit[i];
        }
        return (gradePoint / totalCredits);
    }
}
```

```
return (gradePoint / totalCredits);
}
}

class student {
    public static void main (String args[]) {
        Std-details s1 = new Std-details();
        for (int i = 0; i < 3; i++) {
            s1[i] = new Std-details();
        }
        for (int j = 0; j < 3; j++) {
            System.out.println("Enter details of " + (j+1) +
                               " student");
            s1[j].getdetails();
        }
        for (int j = 0; j < 3; j++) {
            s1[j].display();
        }
        System.out.println("Name: Aniz Ejaz Khan
                           usn: 1BM23CS051");
    }
}

o/p
3 x 3 subg
Enter details of 1 student
Enter the usn and name
1BM23CS001

Anman
Enter the marks
65
75
85
Enter the credits
3
3
3
usn: 1BM23CS001
name: Anman
marks of 1 subject 65
marks of 2 subject 75
marks of 3 subject 85
SGPA: 8.0
Anme
Name: Aniz Ejaz Khan usn: 1BM23CS051
For 3 subjects
Enter details of 1 student
Enter the usn and name
1BM23CS001
Anman
Enter the marks
60
67
67
```



56  
78  
56  
77  
82

Enter the credits

3  
3  
3  
3  
3  
3  
3

USN = 18M23CJ002  
name = Asman

marks of 1 subject	60
marks of 2 subject	67
marks of 3 subject	67
marks of 4 subject	56
marks of 5 subject	78
marks of 6 subject	56
marks of 7 subject	77
marks of 8 subject	82

S GPA : 7.25

Name = Ariz Ezeq Khan USN: 18M23C5051

Code:

```
import java.util.*;
class Stud_details{
int marks[]=new int[8];
int credit[]=new int[8];
String usn,name;
Scanner sc = new Scanner(System.in);

void getdetails(){
System.out.println("Enter the usn and name");
usn=sc.next();
name=sc.next();
System.out.println("Enter the marks");
for(int i=0;i<8;i++){
marks[i]=sc.nextInt();
}
}
```



```

System.out.println("Enter the credits");
for(int i=0;i<8;i++){
    credit[i]=sc.nextInt();
}

}

void display(){
    System.out.println("usn:"+usn);
    System.out.println("name:"+name);
    for(int i=0;i<8;i++){
        System.out.println("marks of "+(i+1)+" subject"+" "+marks[i]);
    }
    System.out.println("SGPA:"+calculateSGPA());
}

double getgradepoint(int mark){
    if(mark>=90) return 10.0;
    else if(mark>=80) return 9.0;
    else if(mark>=70) return 8.0;
    else if(mark>=60) return 7.0;
    else if(mark>=50) return 6.0;
    else if(mark>=40) return 5.0;
    else return 0.0;
}

double calculateSGPA(){
    int totalcredits=0;
    double gradepoint=0;
    for(int i=0;i<8;i++){
        totalcredits+=credit[i];
    }
    for(int i=0;i<8;i++){
        gradepoint+=getgradepoint(marks[i])*credit[i];
    }
    return (gradepoint/totalcredits);
}

}

class student{
    public static void main(String args[]){
        Stud_details s1[]=new Stud_details[3];
        for(int j=0;j<3;j++){
            s1[j]=new Stud_details();

```

```

}
for(int j=0;j<3;j++){
System.out.println("Enter details of "+(j+1)+" student");
s1[j].getdetails();
}
for(int j=0;j<3;j++){
s1[j].display();
}
System.out.println("\nName:Ariz Ejaz Khan USN:1BM23CS051");
}
}

```

```

D:\java>javac student.java

D:\java>java student
Enter details of 1 student
Enter the usn and name
1BM23CS001
Arman
Enter the marks
60
67
67
56
78
56
77
82
Enter the credits
3
3
3
3
3
3
3
3
3
Enter details of 2 student
Enter the usn and name
1BM23CS002
Arya
Enter the marks
87
86
66
75
67
56
77
90
Enter the credits
3
3

```

```

3
3
3
3
3
3
3
3
Enter details of 3 student
Enter the usn and name
1BM23CS003
Akash
Enter the marks
77
88
87
84
56
78
98
92
Enter the credits
3
3
3
3
3
3
3
3
usn:1BM23CS001
name:Arman
marks of 1 subject 60
marks of 2 subject 67
marks of 3 subject 67
marks of 4 subject 56
marks of 5 subject 78
marks of 6 subject 56
marks of 7 subject 77
marks of 8 subject 82
SGPA:7.25

```

```

usn:1BM23CS002
name:Arya
marks of 1 subject 87
marks of 2 subject 86
marks of 3 subject 66
marks of 4 subject 75
marks of 5 subject 67
marks of 6 subject 56
marks of 7 subject 77
marks of 8 subject 90
SGPA:8.0
usn:1BM23CS003
name:akash
marks of 1 subject 77
marks of 2 subject 88
marks of 3 subject 87
marks of 4 subject 84
marks of 5 subject 56
marks of 6 subject 78
marks of 7 subject 98
marks of 8 subject 92
SGPA:8.625

Name:Ariz Ejaz Khan USN:1BM23CS051
D:\java>

```

### Program 3

Implement Book class with set and display details and toString function.

Algorithm:

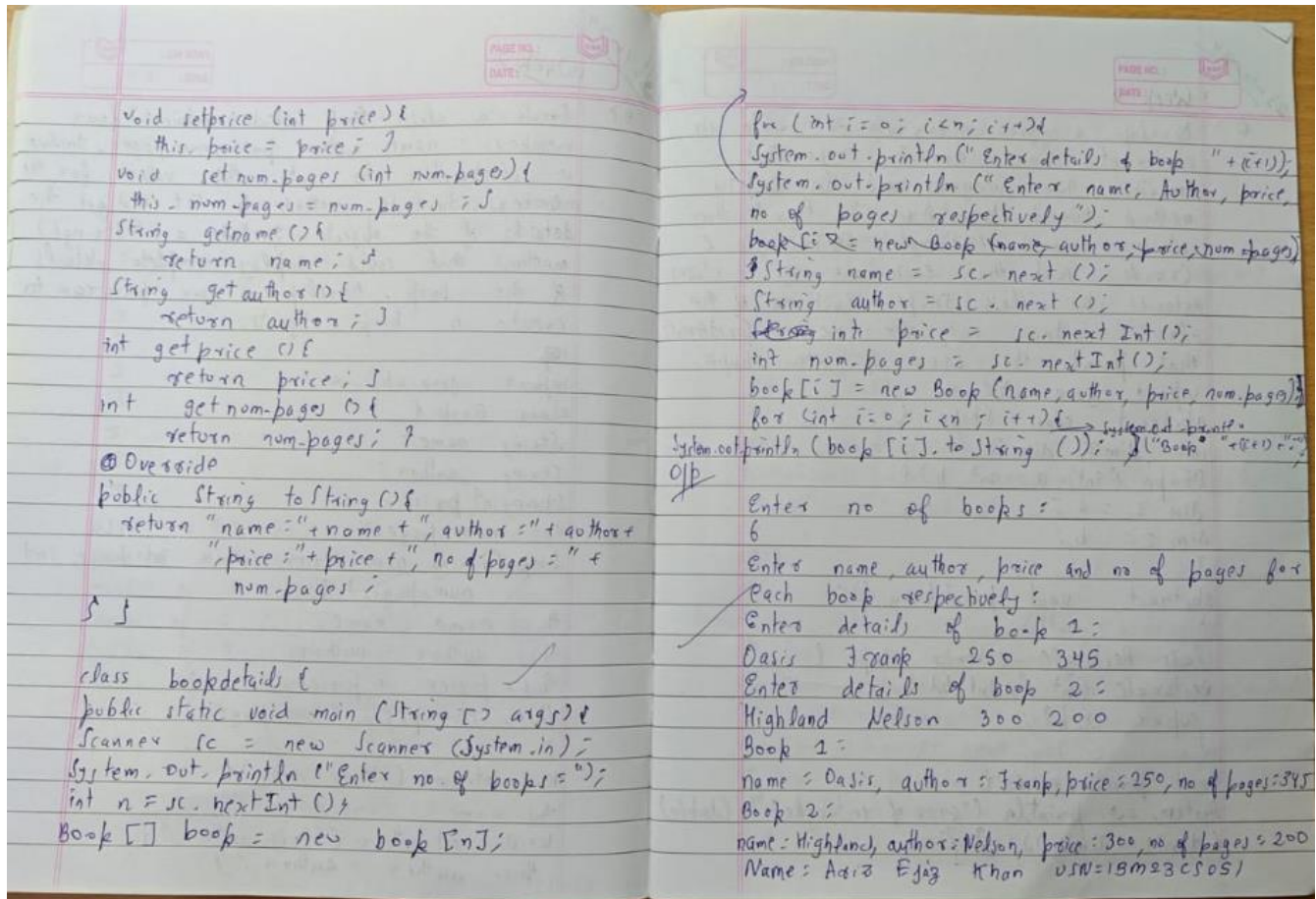
16-10-24

Week 3

PAGE NO. :  
DATE :

Q7 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 complete details of the book. Develop a java program to create n book objects.

```
import java.util.*;
class Book {
    String name;
    String author;
    String price;
    int num-pages;
    Book (String name, String author, int price, int num-pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num-pages = num-pages;
    }
    void setName (String name) {
        this.name = name;
    }
    void setAuthor (String author) {
        this.author = author;
    }
}
```



Code:

```

import java.util.*;
class Book {
    String name;
    String author;
    int price;
    int num_pages;
    Book(String name,String author,int price,int num_pages){
        this.name=name;
        this.author=author;
        this.price=price;
        this.num_pages=num_pages;
    }
    public String toString(){
        return "name:"+name+",author:"+author+",price:"+price+",no of pages:"+num_pages;
    }
}

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

```

```

Scanner sc=new Scanner(System.in);
System.out.println("Enter no of books:");
int n=sc.nextInt();
Book[] book=new Book[n];
System.out.println("Enter name,author,price and no of pages for each book respectively:");
for(int i=0;i<n;i++){
System.out.println("Enter details of book "+(i+1)+":");
String name=sc.next();
String author=sc.next();
int price=sc.nextInt();
int num_pages=sc.nextInt();
book[i]=new Book(name,author,price,num_pages);
}
for(int i=0;i<n;i++){
System.out.println("Book "+(i+1)+":");
System.out.println(book[i].toString());
}
}
}

```

```
E:\ariz\week3>javac BookDetails.java
```

```
E:\ariz\week3>java BookDetails
```

```
Enter no of books:
```

```
6
```

```
Enter name,author,price and no of pages for each book respectively:
```

```
Enter details of book 1:
```

```
Oasis Ariz 234 432
```

```
Enter details of book 2:
```

```
Hello Aman 432 550
```

```
Enter details of book 3:
```

```
Rise Rajesh 330 540
```

```
Enter details of book 4:
```

```
Interest Mahesh 343 543
```

```
Enter details of book 5:
```

```
LookUp Thomas 230 331
```

```
Enter details of book 6:
```

```
Forget Ravi 126 320
```

```
Book 1:
```

```
name:Oasis,author:Ariz,price:234,no of pages:432
```

```
Book 2:
```

```
name:Hello,author:Aman,price:432,no of pages:550
```

```
Book 3:
```

```
name:Rise,author:Rajesh,price:330,no of pages:540
```

```
Book 4:
```

```
name:Interest,author:Mahesh,price:343,no of pages:543
```

```
Book 5:
```

```
name:LookUp,author:Thomas,price:230,no of pages:331
```

```
Book 6:
```

```
name:Forget,author:Ravi,price:126,no of pages:320
```

```
E:\ariz\week3>Name:Ariz Ejaz Khan USN:1BM23CS051
```



## Program 4

Implement abstract class Shape.

Algorithm:

```
23-10-24
Week 4
0 > 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 &
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 given shape.
-> import java.util.*;
abstract class Shape {
    int dim1, dim2;
    Shape (int a, int b) {
        dim1 = a;
        dim2 = b;
    }
    abstract void printArea();
}
class Rectangle extends Shape {
    Rectangle (int l, int b) {
        super (l, b);
    }
    void printArea () {
        System.out.println ("area of rectangle: " + (double)
        (dim1 * dim2));
    }
}
class Triangle extends Shape {
    Triangle (int b, int h) {
        super (b, h);
    }
    void printArea () {
        System.out.println ("Area of triangle: " + (double)
        (dim1 * dim2 / 2));
    }
}
class Circle extends Shape {
    Circle (int r, int d) {
        super (r, d);
    }
    void printArea () {
        System.out.println ("area of circle: " + (double)
        (3.14 * dim1 * dim1));
    }
}
class shapearea {
    public static void main (String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter length & width of rectangle");
        int length = sc.nextInt();
        int breadth = sc.nextInt();
        System.out.println ("Enter base & height of
        triangle");
        int base = sc.nextInt();
        int height = sc.nextInt();
        System.out.println ("Enter radius of circle");
        int radius = sc.nextInt();
        Rectangle r = new Rectangle (length, breadth);
        Triangle t = new Triangle (base, height);
        Circle c = new Circle (radius, 0);
        r.printArea();
        t.printArea();
        c.printArea();
    }
}

o/p
Enter length and width of rectangle :
10 20
Enter base and height of triangle :
10 20
Enter radius of circle :
12
area of rectangle : 200.0
area of triangle : 100.0
area of circle : 452.15999999999997

Name = Aziz Ejaz Khan
USN = 18M23CS05
```

Code:



```

import java.util.*;
abstract class Shape{
int dim1,dim2;
Shape(int a,int b){
dim1=a;
dim2=b;
}
abstract void printArea();
}

class Rectangle extends Shape{
Rectangle(int l,int b){
super(l,b);
}
void printArea(){
System.out.println("area of rectangle:"+(double)(dim1*dim2));
}
}

class Triangle extends Shape{
Triangle(int b,int h){
super(b,h);
}
void printArea(){
System.out.println("area of tringle:"+(double)(dim1*dim2/2));
}
}

class Circle extends Shape{
Circle(int r,int d){
super(r,d);
}
void printArea(){
System.out.println("area of circle:"+(double)(3.14*dim1*dim1));
}
}

class shapearea{
public static void main(String[] args){
Scanner sc=new Scanner(System.in);
System.out.println("Enter length and width of retangle:");
int length=sc.nextInt();
int breadth=sc.nextInt();
System.out.println("Enter base and height of triangle:");
int base=sc.nextInt();
int height=sc.nextInt();
}
}

```

```

System.out.println("Enter radius of circle:");
int radius=sc.nextInt();
Rectangle r=new Rectangle(length,breadth);
Triangle t=new Triangle(base,height);
Circle c=new Circle(radius,0);
r.printArea();
t.printArea();
c.printArea();
System.out.println("\nName:Ariz Ejaz Khan USN:1BM23CS051");
}
}

```

```
E:\ariz\week4>javac shapearea.java
```

```

E:\ariz\week4>java shapearea
Enter length and width of retangle:
10 20
Enter base and height of triangle:
10 20
Enter radius of circle:
12
area of rectangle:200.0
area of tringle:100.0
area of circle:314.0

```

```
E:\ariz\week4>javac shapearea.java
```

```

E:\ariz\week4>java shapearea
Enter length and width of retangle:
10 20
Enter base and height of triangle:
10 20
Enter radius of circle:
12
area of rectangle:200.0
area of tringle:100.0
area of circle:452.15999999999997

```

```
E:\ariz\week4>Name:Ariz Ejaz Khan USN:1BM23CS051|
```

### Program 5

Implement class Bank, Account, Cur-acct, Sav-acct.

Algorithm:

30-10-24

Week 5

PAGE NO. :  
DATE : / /

Q. Develop a java program to create a class Bank ~~that~~ that maintains ~~two~~ savings & current account. The savings account provides compound interest & withdrawal facilities but no cheque book facility. The current account provides cheque book facility but with no interest. Current account holder should ~~also~~ maintain a min. balance else, a fine is created.

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

- a) Accept deposit from customer and update the balance
- ii) Display balance
- iii) Compute & deposit interest
- iv) Permit withdrawal & update balance
- v) Check for min balance, impose penalty if ~~if necessary~~ necessary, update the balance.

class Account {  
    String cname;  
    int accno;

```

String accType;
double balance;
int accNo;
String cName;
Account (String cName, int accNo, String accType, double initialBalance) {
    cName = cName;
    accNo = accNo;
    accType = accType;
    balance = initialBalance;
}

void deposit (double amount) {
    if (amount > 0) {
        balance += amount;
        System.out.println ("Deposited: " + amount);
    } else {
        System.out.println ("Invalid Deposit");
    }
}

void displayBalance () {
    System.out.println ("Current balance: " + balance);
}

class SaveAcct extends Account {
    final double ix = 0.05;
    SaveAcct (String cName, int accNo, double initialBalance) {
        super (cName, accNo, "Savings", initialBalance);
    }
}

```

```

void getInterest () {
    double interest = balance * ix;
    balance += interest;
    System.out.println ("Interest added: " + interest);
}

void withdraw (double amount) {
    if (amount > balance) {
        System.out.println ("Not enough balance");
    } else {
        balance -= amount;
        System.out.println ("Withdrawn: " + amount);
    }
}

class CurAcct (String cName, int accNo, double initialBalance) {
    final
    class CurAcct extends Account {
        final double minBalance = 5000.0;
        final double serviceCharge = 50.0;
        CurAcct (String cName, int accNo, double initialBalance) {
            super (cName, accNo, "Current", initialBalance);
        }
        void checkMinimumBalance () {

```

```

if (balance < minBalance) {
    balance -= serviceCharge;
    System.out.println ("Service charge imposed: " + serviceCharge);
}

void withdraw (double amount) {
    if (amount > balance) {
        System.out.println ("Insufficient balance");
    } else {
        balance -= amount;
        checkMinimumBalance();
        System.out.println ("Amount withdrawn: " + amount);
    }
}

```

```

public class Bank {
    public static void main (String args[]) {
        SaveAcct sa = new SaveAcct ("Aziz", 101, 2000);
        CurAcct ca = new CurAcct ("Sajid", 102, 1000);
        System.out.println ("Savings Account:");
        sa.deposit (500);
        sa.getInterest ();
        sa.withdraw (200);
        sa.displayBalance ();
    }
}

```

```

System.out.println ("In Current Account:");
ca.deposit (200);
ca.withdraw (800);
ca.displayBalance ();
}
}

```

O/P

```

Savings account:
Deposited: 500.0
Interest added: 125.0
Withdrawn: 200.0
Current Balance: 2425.0

Current account:
Deposited: 200.0
Service charge imposed: 50.0
Amount withdrawn: 800.0
Current Balance: 350

```

Name: Aziz Ejaz Khan  
 UIN: 18M23CS051

Code:

```

class Account{

```

```

String cname;
int accno;
String acctype;
double balance;
Account(String cName,int accNo,String accType,double initialbalance){
    cname=cName;
    accno=accNo;
    acctype=accType;
    balance=initialbalance;
}
void deposit(double amount){
    if(amount>0){
        balance+=amount;
        System.out.println("Deposited:"+amount);
    }
    else{
        System.out.println("Invalid Deposit");
    }
}
void displayBalance(){
    System.out.println("Current Balance:"+balance);
}
}

```

```

class SavAcct extends Account{
    final double ir=0.05;
    SavAcct(String cName,int accNo,double initialbalance){
        super( cName, accNo,"Savings",initialbalance);
    }
    void getInterest(){
        double interest=balance*ir;
        balance+=interest;
        System.out.println("Interest added:"+interest);
    }
    void withdraw(double amount){
        if(amount>balance){
            System.out.println("Not enough balance");
        }
        else{
            balance-=amount;
            System.out.println("Withdrawn:"+amount);
        }
    }
}

```

```

class CurAcct extends Account{
    final double minbalance=500.0;

```

```

final double serviceCharge=50.0;
CurAcct(String cName,int accNo,double initialbalance){
    super( cName, accNo,"Current",initialbalance);
}
void checkMinimumBalance(){
    if(balance<minbalance){
        balance-=serviceCharge;
        System.out.println("Service charge imposed:"+serviceCharge);
    }
}
void withdraw(double amount){
    if(amount>balance){
        System.out.println("Insuffecient balance");
    }
    else{
        balance-=amount;
        checkMinimumBalance();
        System.out.println("Amount withdrawn:"+amount);
    }
}
}

public class Bank{
    public static void main(String args[]){
        SavAcct sa=new SavAcct("Ariz",101,2000);
        CurAcct ca=new CurAcct("Sejil",102,1000);
        System.out.println("Savings account:");
        sa.deposit(500);
        sa.getInterest();
        sa.withdraw(200);
        sa.displayBalance();
        System.out.println("\nCurrent account:");
        ca.deposit(200);
        ca.withdraw(800);
        ca.displayBalance();
    }
}

```



```
D:\java>javac bank.java
```

```
D:\java>java Bank
```

```
Savings account:
```

```
Deposited:500.0
```

```
Interest added:125.0
```

```
Withdrawn:200.0
```

```
Current Balance:2425.0
```

```
Current account:
```

```
Deposited:200.0
```

```
Service charge imposed:50.0
```

```
Amount withdrawn:800.0
```

```
Current Balance:350.0
```

```
D:\java> Name:Ariz Ejaz khan   USN:1BM23CS051|
```



## Program 6

Implement packages CIE and SEE.

Algorithm:

```
13-11-24
Week - 6

02 Create a package CIE which has two
classes - student & internal. The class
Student has members like usn, name, sem.
The class internal 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 student.
Export the two packages in a file that
declares the final marks of a student
in all five courses.

package CIE;
public class Student {
    public String usn;
    public String name;
    public int sem;
    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

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

public class Internal {
    public int[] marks;
    public Internal(int[] marks) {
        if (marks.length != 5) {
            System.out.println("Enter 5 marks!");
        }
        else {
            this.marks = marks;
        }
    }
}

public void displaymarks() {
    System.out.println("Respective marks:");
    for (int i = 0; i < 5; i++) {
        System.out.print(marks[i] + " ");
    }
    System.out.println();
}

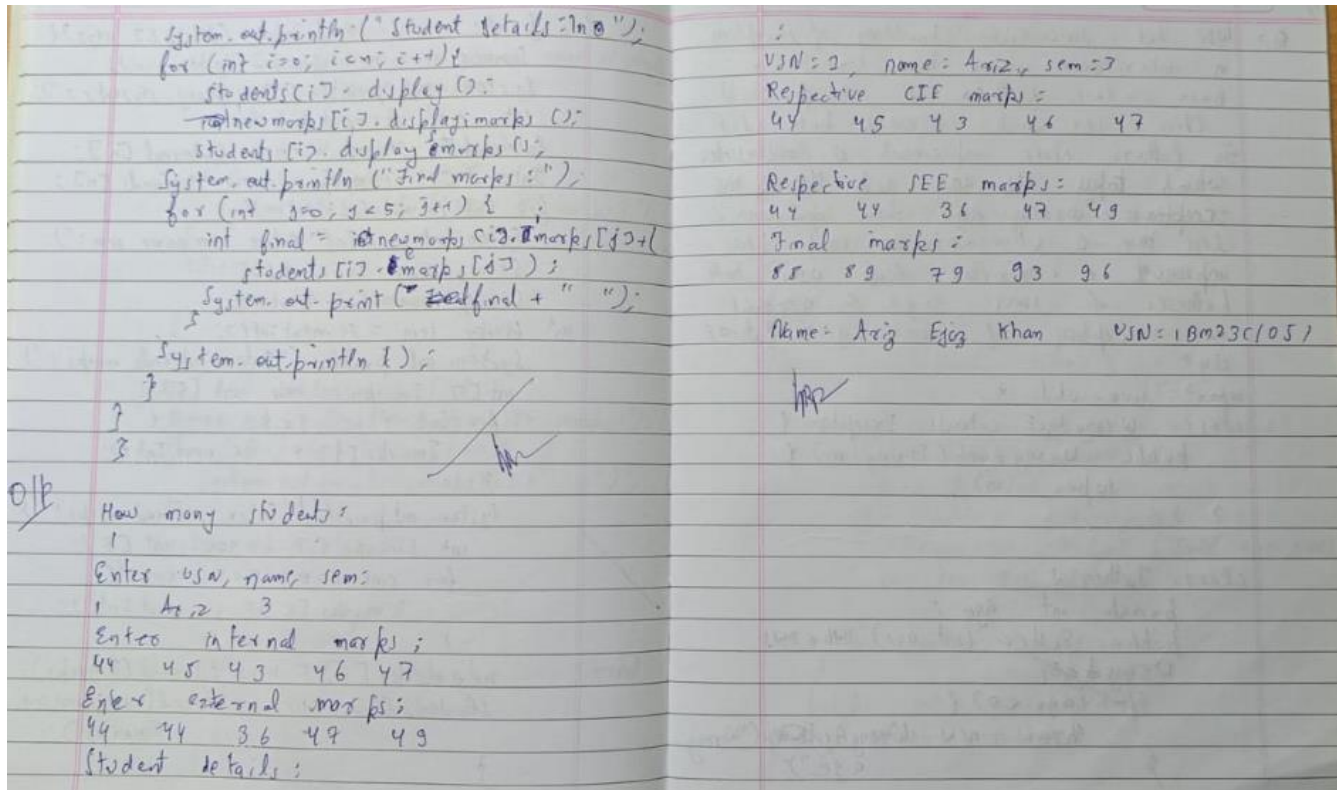
package SEE;
import CIE.*;
public class External extends Student {
```

```
public int[] marks;
public External(String usn, String name, int sem) {
    super(usn, name, sem);
    if (marks.length != 5) {
        System.out.println("Need 5 marks!");
    }
    else {
        marks = marks;
    }
}

public void displaymarks() {
    System.out.println("Respective SEE marks:");
    for (int i = 0; i < 5; i++) {
        System.out.print(marks[i] + " ");
    }
    System.out.println();
}

import CIE.*;
import SEE.*;
import java.util.*;
public class Main {

    public static void main (String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println("How many students:");
        int n = sc.nextInt();
        External[] students = new External[n];
        Internal[] newmarks = new Internal[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter usn, name, sem:");
            String usn = sc.next();
            String name = sc.next();
            int sem = sc.nextInt();
            System.out.println("Enter internal marks:");
            int[] Imarks = new int[5];
            for (int j = 0; j < 5; j++) {
                Imarks[j] = sc.nextInt();
            }
            System.out.println("Enter external marks:");
            int[] Emarks = new int[5];
            for (int k = 0; k < 5; k++) {
                Emarks[k] = sc.nextInt();
            }
            newmarks[i] = new Internal(Imarks);
            students[i] = new External(usn, name, sem, Emarks);
        }
    }
}
```



Code:

//INTERNAL

package CIE;

```

public class Internal{
    public int imarks[];
    public Internal(int[] marks){
        if(marks.length!=5) System.out.println("Need 5 marks!");
        else this.imarks=marks;
    }
    public void displayimarks(){
        System.out.println("Respective CIE marks:");
        for(int i=0;i<5;i++){
            System.out.println(imarks[i]+" ");
        }
        System.out.println();
    }
}

```

//EXTERNAL

package SEE;

```

import CIE.*;
public class External extends Student{
public int[] smarks;
public External(String usn,String name,int sem,int[] marks){
super(usn,name,sem);
if(marks.length!=5) System.out.println("Need 5 marks!");
else this.smarks=marks;
}
public void displaysmarks(){
System.out.println("Respective SEE marks:");
for(int i=0;i<5;i++){
System.out.println(smarks[i]+" ");
}
System.out.println();
}
}

```

```

//STUDENT
package CIE;
public class Student{
public String usn;
public String name;
public int sem;
public Student(String usn,String name,int sem){
this.usn=usn;
this.name=name;
this.sem=sem;
}
public void display(){
System.out.println("USN:"+usn+",name:"+name+",sem:"+sem);
}
}

```

```

//MAIN
import CIE.Student;
import CIE.Internal;
import SEE.External;
import java.util.*;
class MAIN {
    public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
System.out.println("How many students:");
int n=sc.nextInt();
External[] students=new External[n];

```

```

Internal[] newmarks=new Internal[n];
for(int i=0;i<n;i++){
System.out.println("Enter usn,name,sem:");
String usn=sc.next();
String name=sc.next();
int sem=sc.nextInt();
System.out.println("Enter internal marks:");
int[] Imarks=new int[5];
for(int j=0;j<5;j++){
Imarks[j]=sc.nextInt();
}
System.out.println("Enter external marks:");
int[] Emarks=new int[5];
for(int j=0;j<5;j++){
Emarks[j]=sc.nextInt();
}
newmarks[i]=new Internal(Imarks);
students[i]=new External(usn,name,sem,Emarks);
}
System.out.println("Student details:\n:");
for(int i=0;i<n;i++){
students[i].display();
newmarks[i].displayimarks();
students[i].displaysmarks();
System.out.println("Final marks:");
for(int j=0;j<5;j++){
int finalm=newmarks[i].imarks[j]+students[i].smarks[j];
System.out.println(finalm+" ");
}
}
System.out.println();

}

}

}

```

```
D:\Ariz>java MAIN
How many students:
2
Enter usn,name,sem:
1 Ariz 3
Enter internal marks:
44 45 43 46 47
Enter external marks:
44 44 36 47 49
Enter usn,name,sem:
2 Aman 3
Enter internal marks:
34 34 45 41 40
Enter external marks:
23 45 43 42 44
Student details:
:
USN:1,name:Ariz,sem:3
Respective CIE marks:
44
45
43
46
47

Respective SEE marks:
44
44
36
47
49

Final marks:
88
89
79
93
96

USN:2,name:Aman,sem:3
Respective CIE marks:
34
34
45
41
41
```

```
USN:2,name:Aman,sem:3
Respective CIE marks:
34
34
45
41
40

Respective SEE marks:
23
45
43
42
44

Final marks:
57
79
88
83
84
Name-Ariz Ejaz Khan, USN-1BM23CS051
D:\Ariz>
```



## Program 7

Implement handling of exceptions in inheritance tree using class Father and class Son.

Algorithm:

```
20-11-24 week-7
Q -> WAP that demonstrates handling of exception in inheritance tree. (create a base class father & a child class son which extends base class). In father's class implement a constructor which takes the age and throws the exception "wrong age" when input age is less than 0 (age < 0). In son's class implement a constructor that uses both father's & son's age & throws an exception if son's age > father's age.
import java.util.*;
class WrongAgeE extends Exception {
    public WrongAgeE(String m) {
        super(m);
    }
}
class Father {
    private int age;
    public Father(int age) throws WrongAgeE {
        if (age < 0) {
            throw new WrongAgeE("Wrong age");
        }
    }
}
class SonAgeE extends Exception {
    public SonAgeE(String m) {
        super(m);
    }
}
class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeE, SonAgeE {
        super(fatherAge);
        if (sonAge > fatherAge) {
            throw new SonAgeE("Son's age should be less than father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}
```

```
public class FatherSon {
    public static void main(String[] args) {
        while (true) {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter Father's age:");
            int fage = sc.nextInt();
            System.out.println("Enter son's age:");
            int sage = sc.nextInt();
            try {
                Son son = new Son(fage, sage);
                System.out.println("Accepted");
            } catch (WrongAgeE e) {
                System.out.println(e.getMessage());
            } catch (SonAgeE e) {
                System.out.println(e.getMessage());
            }
            System.out.println("RE-enter details again (Y/n) = ");
            String input = sc.next();
            if (input.equals("n")) {
                break;
            }
        }
    }
}
```

o/p

```
Enter Father's Age : 34
Enter Son's Age : 4
Accepted
Enter details again: (Y/n)
Y
Enter Father's Age : 54
Enter Son's Age : 21
Accepted
Enter details again: (Y/n)
Y
Enter Father's Age : 45
Enter Son's age : 46
Son's age should be less than father's age
Enter details again: (Y/n)
n
Name: Aiz Ejaz Khan USN: 18M23CS051
```

Code:

```
import java.util.Scanner;
class WrongAgeE extends Exception {
    public WrongAgeE(String m) {
        super(m);
    }
}

class SonAgeE extends Exception {
    public SonAgeE(String m) {
        super(m);
    }
}

class Father {
    private int age;
    public Father(int age) throws WrongAgeE {
        if (age < 0) {
            throw new WrongAgeE("Wrong age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeE, SonAgeE {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeE("Son's age should be less than father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}

public class FatherSon{
    public static void main(String[] args) {
        while(true){
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter Father's Age: ");
```



```

int fage = sc.nextInt();
System.out.print("Enter Son's Age: ");
int sage = sc.nextInt();
try {
    Son son = new Son(fage, sage);
    System.out.println("Accepted");
}
catch (WrongAgeE e) {
    System.out.println(e.getMessage());
}
catch (SonAgeE e) {
    System.out.println(e.getMessage());
}
System.out.println("Enter details again:(Y/n)");
String input = sc.next();
if (input.equalsIgnoreCase("n")) {
    break;
}
}
}
}

```

```
D:\>cd Ariz
```

```
D:\Ariz>javac FatherSon.java
```

```
D:\Ariz>java FatherSon
```

```
Enter Father's Age:
```

```
34
```

```
Enter Son's Age: 4
```

```
Accepted
```

```
Enter details again:(Y/n)
```

```
y
```

```
Enter Father's Age:
```

```
54
```

```
Enter Son's Age: 21
```

```
Accepted
```

```
Enter details again:(Y/n)
```

```
y
```

```
Enter Father's Age:
```

```
45
```

```
Enter Son's Age: 46
```

```
Son's age should be less than father's age
```

```
Enter details again:(Y/n)
```

```
n
```

```
D:\Ariz>Name:Ariz Ejaz Khan USN:1BM23CS051|
```

## Program 8

Implement two Threads.

Algorithm:

27-11-24 week 8

Q2 WAP 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 extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10 * 1000);
            }
        } catch (InterruptedException e) {}
    }
}

class CSE extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {}
    }
}
```

o/p

```
public class Multithread {
    public static void main (String[] args) {
        BMS bms = new BMS();
        CSE cse = new CSE();
        bms.start();
        cse.start();
    }
}
```

BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
AC

Name: Aziz Ejaz Khan UIN: 18M23C5051

Code:

```
class BMS extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10*1000);
            }
        } catch (InterruptedException e) {}
    }
}

class CSE extends Thread {
    public void run() {
        try {
```

```

        while (true) {
            System.out.println("CSE");
            Thread.sleep(2000);
        }
    } catch (InterruptedException e) {}
}
}

public class Multithread{
    public static void main(String[] args) {
        BMS bms = new BMS();
        CSE cse = new CSE();
        bms.start();
        cse.start();
    }
}

```

```

D:\ariz2>java Multithread
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
^C
D:\ariz2>Name:Ariz Ejaz Khan   USN:1BM23CS051|

```

### Program 9

Implement the creation of a User Interface for integer divisions with exceptions handling.

Algorithm:

27-11-24

Week 9

Code (UI creation)

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class Swing Demo {
    Swing Demo () {
        JFrame jfrm = new JFrame ("Dividers App");
        jfrm.setSize ( 275, 150 );
        jfrm.setLayout ( new FlowLayout () );
        jfrm.setDefaultCloseOperation ( JFrame.EXIT_ON_CLOSE );

        JLabel jlab = new JLabel ("Enter divider & dividend");
        JTextField atjf = new JTextField ( 8 );
        JTextField bjlf = new JTextField ( 8 );

        JButton button = new JButton ("Calculate")

        JLabel err = new JLabel ();
        JLabel alab = new JLabel ();
        JLabel blab = new JLabel ();
        JLabel ans lab = new JLabel ();

        jfrm.add (err);
```



```

jform.add (g1ab);
jform.add (a1tf);
jform.add (b1tf);
jform.button.add (button);
jform.add (a1ab);
jform.add (b1ab);
jform.add (ans1ab);

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

a1tf.add (ActionListener (I));
b1tf.add (ActionListener (I));

button.add (ActionListener (new ActionListener () {
    public void actionPerformed (ActionEvent evt) {
        try {
            int a = Integer.parseInt (a1tf.getText ());
            int b = Integer.parseInt (b1tf.getText ());
            int ans = a/b;

            a1ab.setText ("A = " + a);

```

```

        b1ab.setText ("B = " + b);
        ans1ab.setText ("Ans = " + ans);
    }
    catch (NumberFormatException e) {
        a1ab.setText ("");
        b1ab.setText ("");
        ans1ab.setText ("");
        evt.setText ("Enter Integer only!");
    }
    catch (ArithmeticException e) {
        a1ab.setText ("");
        b1ab.setText ("");
        ans1ab.setText ("");
        evt.setText ("B should be Non Zero!");
    }
});

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

```

27-11-24

## Open Ended Questions

PAGE NO.:  
DATE: / /

- Q > 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 where the Divide button is pressed. 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.

Dividers APP
□ ×

Enter the divider and dividend:-

Name = Aris Ejaz Khan  
USN = 1BM23CS051

Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
SwingDemo(){
// create jframe container
JFrame jfrm = new JFrame("Divider App");
jfrm.setSize(275, 150);
jfrm.setLayout(new FlowLayout());
// to terminate on close
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// text label
JLabel jlab = new JLabel("Enter the divider and dividend:");
// add text field for both numbers
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
// calc button
JButton button = new JButton("Calculate");
// labels
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();

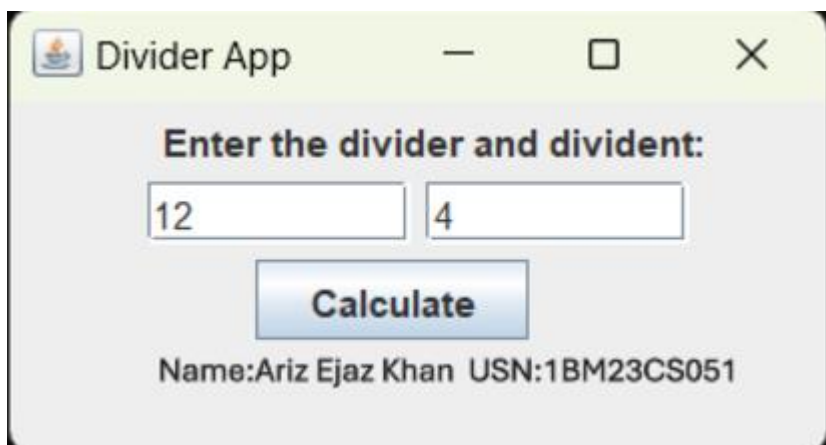
JLabel anslab = new JLabel();
// add in order :)
jfrm.add(err); // to display error boi
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
ActionListener l = new ActionListener() {
public void actionPerformed(ActionEvent evt) {
System.out.println("Action 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();
}
});
}
}

```





Divider App

Enter the divider and dividend:

12 4

Calculate A = 12 B = 4 Ans = 3

Name:Ariz Ejaz Khan USN:1BM23CS051

Divider App

B should be NON zero!

Enter the divider and dividend:

12 0

Calculate

Name:Ariz Ejaz Khan USN:1BM23CS051

## Program 10

Demonstrate IPC and Deadlock.

Algorithm:

```
27-11-24
Week 10
IPC code

class B {
    int n;
    boolean valueset = false;
    synchronized int get() {
        while (!valueset)
            try {
                System.out.println("In consumer waiting\n");
                wait();
            } catch (InterruptedException e) {}
        System.out.println("InterruptedException caught");
        System.out.println("B: " + n);
        valueset = false;
        synchronized void put(int n) {
            while (valueset)
                try {
                    System.out.println("In 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("In Inside consumer\n");
        }
    }
}

notify();
}

class Producer implements Runnable {
    B q;
    Producer(B 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 {
    B q;
    Consumer(B 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++;
        }
    }
}
```

```
27-11-24
Deadlock code

class PCFixed {
    public static void main(String args[]) {
        B q = new B(2);
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop");
    }
}

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 ();
```

```
}
```

27-11-24

Q2 Demonstrate inter process communication and deadlock.

D/P ~~Interprocess communication~~  
Deadlock

Racing Thread entered B.bar  
Main Thread entered A.foo  
Main Thread trying to call B.bar()  
Racing Thread trying to call A.bar()  
Inside A.bar  
Back in other thread  
Inside A.bar  
Back in main thread  
Name: Aziz Ejaz Khan USN: 18M23CS051

Interprocess communication

2/p Press control-C to stop  
Put = 0  
Animate consumer  
Producer waiting  
Get = 0  
Animate producer  
Put = 1  
Animate consumer  
Producer waiting  
Consumed = 0

Get = 1  
Animate producer  
consumed = 1  
Put = 2  
Animate consumer  
Producer waiting  
Get = 2  
Animate producer  
consumed = 2  
Put = 3  
Animate consumer  
Producer waiting  
Get = 3  
Animate producer  
consumed = 3  
Put = 4  
Animate consumer  
Producer waiting  
Get = 4  
Animate consumer  
consumed = 4  
Put = 5  
Name: Aziz  
Name: Aziz Ejaz Khan USN: 18M23CS051

Seen  
gl  
attitude

Code:  
(IPC)

```
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.");

}

}

```

```
D:\java>javac PCFixed.java

D:\java>java PCFixed
Press Control-C to stop.
Put: 0

Intimate Consumer

Producer waiting
Got: 0
Intimate Producer
Put: 1
Intimate Consumer

Producer waiting
consumed:0
Got: 1
Intimate Producer
consumed:1
Put: 2
Intimate Consumer

Producer waiting
Got: 2
Intimate Producer
consumed:2
Put: 3
Intimate Consumer

Producer waiting
Got: 3
Intimate Producer
consumed:3
Put: 4
Intimate Consumer

Producer waiting
Got: 4
Intimate Producer
consumed:4
Put: 5
Intimate Consumer

Producer waiting
Got: 5
Intimate Producer
consumed:5
Put: 6
Intimate Consumer

Producer waiting
Got: 6
Intimate Producer
consumed:6
Put: 7
Intimate Consumer

Producer waiting
```

```
Producer waiting
Get: 7
Intimate Producer
consumed:7
Put: 8
Intimate Consumer

Producer waiting
Get: 8
Intimate Producer
consumed:8
Put: 9
Intimate Consumer

Producer waiting
Get: 9
Intimate Producer
consumed:9
Put: 10
Intimate Consumer

Producer waiting
Get: 10
Intimate Producer
consumed:10
Put: 11
Intimate Consumer

Producer waiting
Get: 11
Intimate Producer
consumed:11
Put: 12
Intimate Consumer

Producer waiting
Get: 12
Intimate Producer
consumed:12
Put: 13
Intimate Consumer

Producer waiting
Get: 13
Intimate Producer
consumed:13
Put: 14
Intimate Consumer

Get: 14
Intimate Producer
consumed:14
D:\java>Name:Ariz Ejaz Khan  USN:18M23CS051
```

(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);

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();  
}  
}
```

```
D:\java>javac Deadlock.java
```

```
D:\java>java Deadlock  
RacingThread entered B.bar  
MainThread entered A.foo  
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
```

```
D:\java>Name:Ariz Ejaz Khan   USN:1BM23CS051|
```