

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
**on**  
**Object Oriented Java Programming**  
**(23CS3PCOOJ)**

*Submitted by*

Kathi Abhilash(**1BM23CS143**)

*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 **Kathi Abhilash(1BM23CS143)**, 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. Surabhi S Assistant 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	30/09/2024	Implement Quadratic Equations	4
2	07/10/2024	Calculate SGPA	9
3	14/10/2024	Book Store	15
4	21/10/2024	Area of Shapes	20
5	28/10/2024	Bank	25
6	11/11/2024	Implement packages	39
7	28/11/2024	Exception Handling	44
8	28/11/2024	MultiThreading	48
9	28/11/2024	Creating Interfaces	52
10	28/11/2024	Demonstrate IPC and Deadlock	57

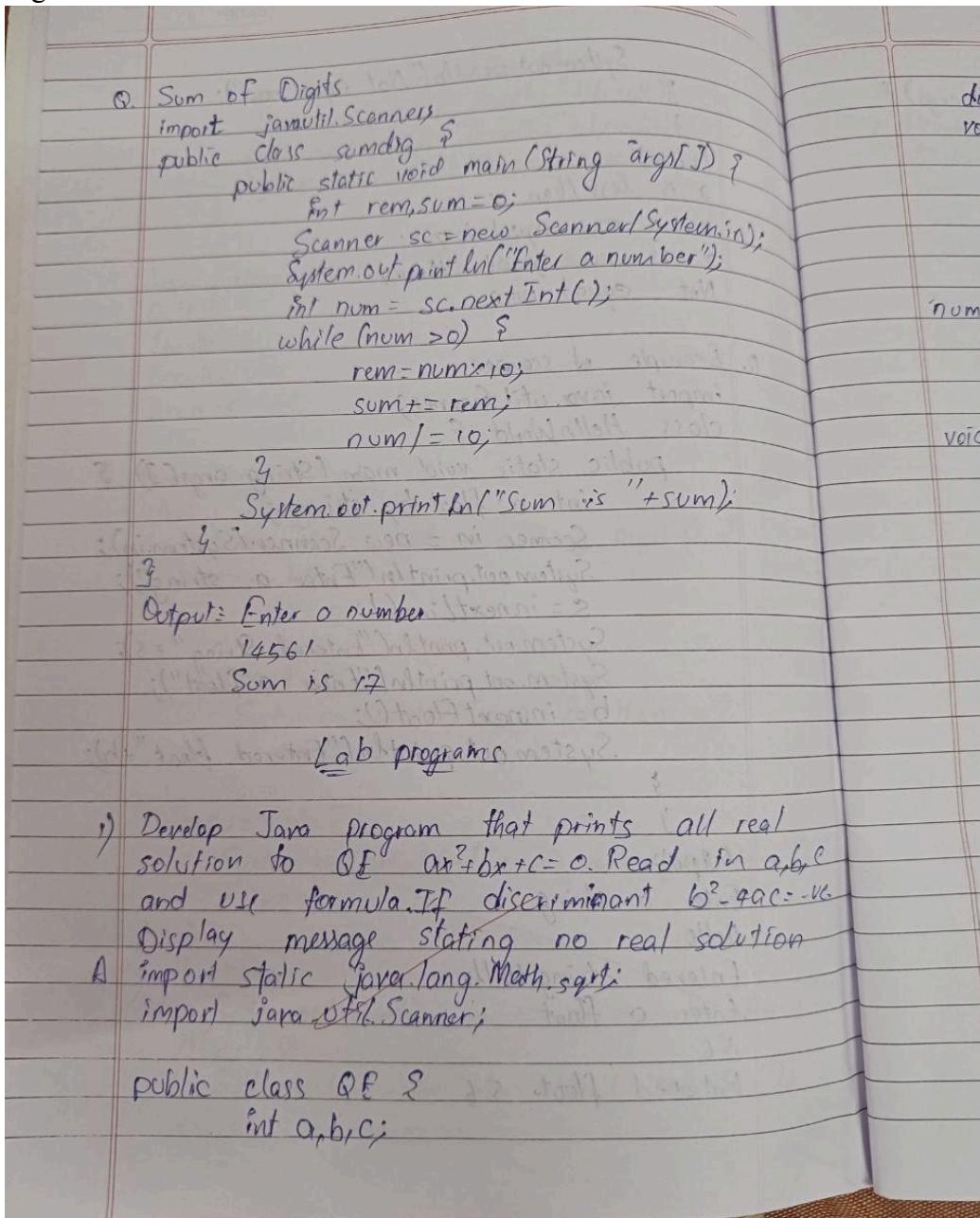
Github Link:

<https://github.com/KathiAbhilash/OOJ-Lab-Programs/tree/main>

### Program 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 discriminant  $b^2 - 4ac$  is negative, display a message stating that there are no real solutions.

Algorithm:



double r1, r2, d;  
void Input() {

Scanner sc = new Scanner(System.in)

System.out.println("Enter value of a:");  
a = sc.nextInt();

while (a == 0) {

System.out.println("Enter a non-zero  
number for a:");

a = sc.nextInt();

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

}

void display() {

if (d == 0) {

r1 = -b / (2.0 \* a);

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

System.out.println("Root: " + r1);

}

else if (d > 0) {

r1 = (-b + sqrt(d)) / (2.0 \* a);

r2 = (-b - sqrt(d)) / (2.0 \* a);

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

System.out.println("r1 = " + r1 + ", r2 = " + r2);

}

else {

r1 = -b / (2.0 \* a);

r2 = sqrt(-d) / (2.0 \* a);

System.out.println("Roots are imaginary");

System.out.println("r1 = " + r1 + " + " + r2 + "i");

System.out.println("r2 = " + r1 + " - " + r2 + "i");

?

```
public static void main (String [] args) {
    QE ge = new QE ();
    ge. input ();
    ge. display ();
}
```

3

**B**

Output:

Enter value of a: 1

Enter value of b: -4

Enter value of c: 4

Roots are real and equal

Root: 2.0

Enter value of a: 2

Enter value of b: 2

Enter value of c: 2

Roots are imaginary

$r_1 = -0.5 + 0.8660i$

$r_2 = -0.5 - 0.8660i$

Enter value of a: 1

Enter value of b: -9

Enter value of c: 10

Roots are real and different

$r_1 = 7.701, r_2 = 1.2984$

**B**  
S.D.

2. Define  
Stude  
and  
and  
calc

inp  
cls

Code:

```
import java.util.Scanner;
import java.lang.Math;
class Quadratic{
public static void main(String args[]){
int a, b, c;
double r1,r2;
Scanner scan = new Scanner(System.in);
System.out.println("Enter values of a:");
a = scan.nextInt();
System.out.println("Enter values of b:");
b = scan.nextInt();
System.out.println("Enter values of c:");
c = scan.nextInt();
scan.close();
double disc=b*b-4*a*c;
System.out.println("The value of discriminant:" + disc);
if(a == 0) {
System.out.println("Not Quadratic Equation");
}
else if(disc == 0) {
System.out.println("Roots are real and equal");
r1=(-b)/(2*a);
System.out.println("root1:" + r1 + " root2:" + r1);
}
else if(disc>0) {
r1=(((-b)+Math.sqrt(disc)))/(double)(2*a);
r2=(((-b)-Math.sqrt(disc)))/(double)(2*a);
System.out.println("root1:" + r1 + " root2:" + r2);
}
else if(disc < 0) {
System.out.println("Roots are Imaginary");
r1=(-b)/(2*a);
r2=Math.sqrt(-disc)/(4*a);
System.out.println("root1:" + r1 + " root2:" + r2);
}
}
}
```

## Output:

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window displays the following text:

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

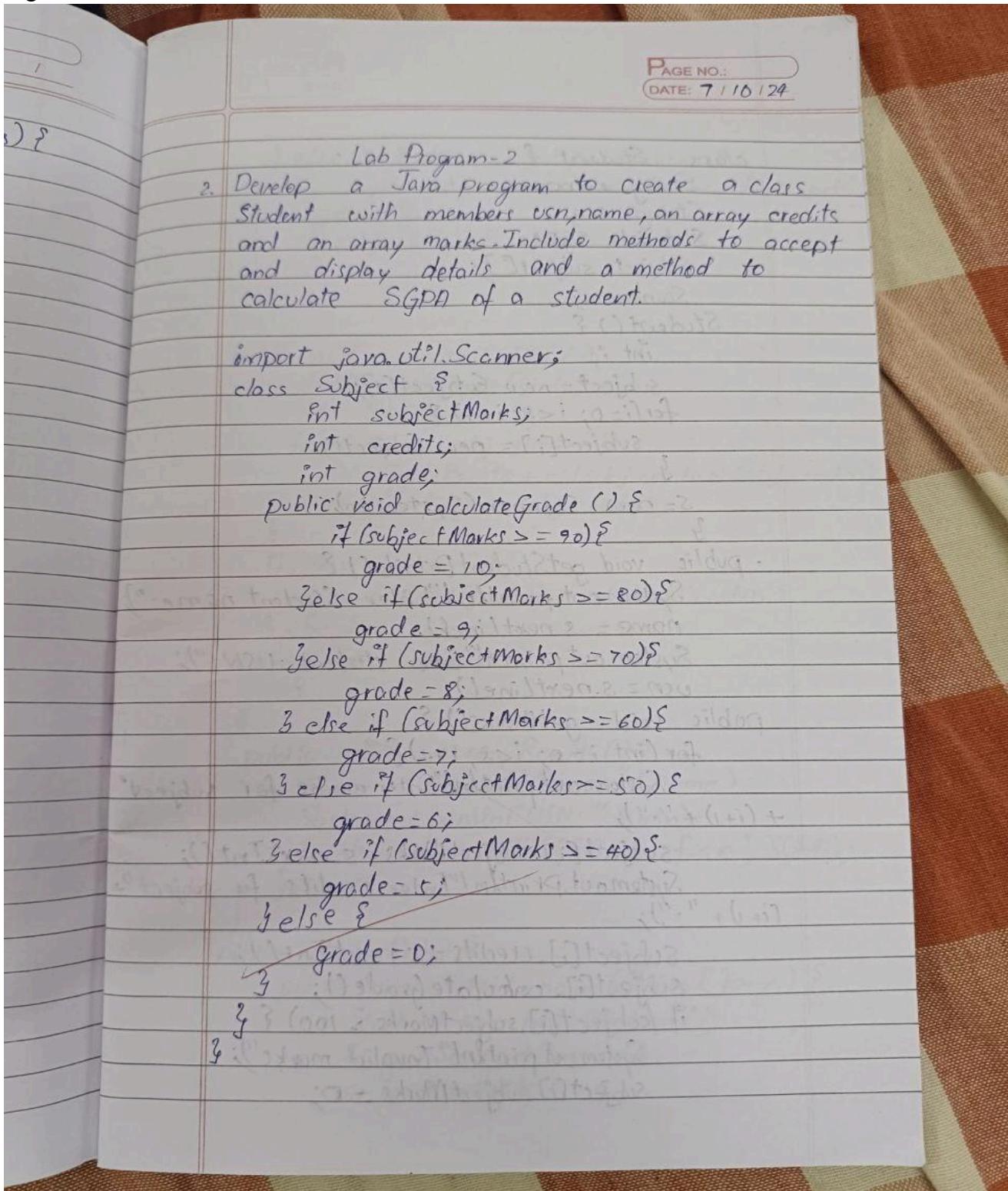
C:\Users\Abhilash Kathi>CD..
C:\Users>CD..
C:\>d:
D:\>cd/java
D:\java>javac Quadratic.java
D:\java>java Quadratic
Enter values of a:
0
Enter values of b:
2
Enter values of c:
3
The value of discriminant:4.0
Not Quadratic Equation
Name: Kathi Abhilash
USN: 1BM23CS143

D:\java>|
```

## Program 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.

Algorithm:



```
{ else if (subject[i].subjectMarks < 0) {  
    System.out.println("Invalid marks, should not  
be negative");  
    subject[i].subjectMarks = 0;  
}  
  
}  
  
public void computeSGPA() {  
    double totalGradePoints = 0;  
    int totalCredits = 0;  
    for (Subject subj : subjects) {  
        totalGradePoints += (subj.grade * subj.credits);  
        totalCredits += subj.credits;  
    }  
    if (totalCredits > 0) {  
        SGPA = totalGradePoints / totalCredits;  
    } else {  
        SGPA = 0.0;  
    }  
}  
  
public void displayResults() {  
    System.out.println("Name: " + name);  
    System.out.println("USN: " + usn);  
    System.out.println("SGPA: %.2f", SGPA);  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Student s1 = new Student();  
        s1.getStudentDetails();  
        s1.getMarks();  
    }  
}
```

st.computeSGPA();  
st.displayResults();

3

## Colgate

Enter Student name - Abhilash

Enter student USN: IBM23CS143

Enter marks for subject 1: 29

Enter credits for subject: 4

Enter remarks for subject 298

Enter credits for subject 4

Final marks for subject: 82

Enter marks for subject 3: 89  
Enter credits for subject 3:

Enter credits for subjects:

Enter marks for subject 4: >

Enter credits for subject4:

Enter marks for subjects : 6

inter credits for subjects:

Enter marks for subjects: 90

Enter credits for subject:

After credit for subjects.

Enter marks for subjects: q

Transfer credits for subjects: 1

riter marks for subject 8.99

Enter credits for subject 81

Home : Akash

Name : Abhishek  
S. No. : 1000-00

DSN: IBM23CS143

~~SGPA : 9.27~~

14.10

Code:

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

```
class Subject {  
    int subjectMarks;  
    int credits;  
    int grade;  
    public void calculateGrade() {
```

```

if (subjectMarks >= 90) {
    grade = 10;
} else if (subjectMarks >= 80) {
    grade = 9;
} else if (subjectMarks >= 70) {
    grade = 8;
} else if (subjectMarks >= 60) {
    grade = 7;
} else if (subjectMarks >= 50) {
    grade = 6;
} else if (subjectMarks >= 40) {
    grade = 5;
} else {
    grade = 0;
}
}

class Student {
    String name;
    String usn;
    double SGPA;
    Subject subject[];
    Scanner s;
    Student() {
        int i;
        subject = new Subject[8];
        for (i = 0; i < 8; i++) {
            subject[i] = new Subject();
        }
        s = new Scanner(System.in);
    }
    public void getStudentDetails() {
        System.out.print("Enter student name: ");
        name = s.nextLine();
        System.out.print("Enter student USN: ");
        usn = s.nextLine();
    }
}

public void getMarks() {
    for (int i = 0; i < 8; i++) {
        System.out.print("Enter marks for subject " + (i + 1) + ": ");
        subject[i].subjectMarks = s.nextInt();
        System.out.print("Enter credits for subject " + (i + 1) + ": ");
        subject[i].credits = s.nextInt();
        subject[i].calculateGrade();
    }
}

```

```

        if (subject[i].subjectMarks > 100) {
            System.out.println("Invalid marks, should not exceed 100.");
            subject[i].subjectMarks = 0;
        } else if (subject[i].subjectMarks < 0) {
            System.out.println("Invalid marks, should not be negative.");
            subject[i].subjectMarks = 0;
        }
    }

public void computeSGPA() {
    double totalGradePoints = 0;
    int totalCredits = 0;

    for (Subject subj : subject) {
        totalGradePoints += (subj.grade * subj.credits);
        totalCredits += subj.credits;
    }

    if (totalCredits > 0) {
        SGPA = totalGradePoints / totalCredits;
    } else {
        SGPA = 0.0;
    }
}

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

public class Main {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        s1.displayResults();
    }
}

```

## Output:

```
Command Prompt + X
error: invalid flag: Main.java
Usage: javac <options> <source files>
use --help for a list of possible options

D:\java>javac Main.java

D:\java>java Main
Enter student name: Abhilash
Enter student USN: 1BM23CS143
Enter marks for subject 1: 89
Enter credits for subject 1: 4
Enter marks for subject 2: 98
Enter credits for subject 2: 4
Enter marks for subject 3: 87
Enter credits for subject 3: 3
Enter marks for subject 4: 78
Enter credits for subject 4: 3
Enter marks for subject 5: 67
Enter credits for subject 5: 3
Enter marks for subject 6: 99
Enter credits for subject 6: 1
Enter marks for subject 7: 99
Enter credits for subject 7: 11
Enter marks for subject 8: 99
Enter credits for subject 8: 1
Name: Abhilash
USN: 1BM23CS143
SGPA: 9.27

D:\java>java Main
```



21:30  
ENG IN 13-10-2024

### **Program 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.

### **Algorithm:**

- 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 book. Develop a java program to create n book objects.

Code →

```
import java.util.Scanner;
class Book {
    private String name;
    private String author;
    private int price;
    private int numPages;

    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.price + "\n";
        numPages = "Number of pages" + this.numPages + "\n";
        return name + author + price + numPages;
    }
}
```

```

    public class Books {
        public static void main (String args []) {
            String name, author;
            int price, int numPages;
            Scanner sc = new Scanner (System.in);
            System.out.print ("Enter the no. of books:");
            int n = sc.nextInt();
            Book [] books = new Book [n];
            for (int i=0; i<n; i++) {
                System.out.println ("Enter the name of the " + (i+1) + " Book:");
                name = sc.nextLine();
                System.out.println ("Enter the author of the " + (i+1) + " Book:");
                author = sc.nextLine();
                System.out.println ("Enter the price of the " + (i+1) + " Book:");
                price = sc.nextInt();
                System.out.println ("Result");
                books[i] = new Book (name, author, price, numPages);
                System.out.println (book[i]);
            }
            System.out.println ("Abhilash Kathi");
            System.out.println ("IBM23(S143)");
        }
    }

```

Output  
 Enter  
 2  
 Enter  
 Abhi  
 Enter  
 87  
 Enter  
 11  
 Enter  
 Boo  
 Auth  
 Pric  
 num

(P)  
Q

Output:-

Enter the number of Books:

2

Enter the name of the book:

Song of Song of Ice and Fire

Enter the author of the book:

Babushka

Enter the price of the book:

879

Enter the number of pages in the book:

1134

Enter Result:

Book name = Song of Ice and Fire

Author name = Babushka

Price: 879

number of pages: 1134

Pages:

8  
21.10

Code:

```
import java.util.Scanner ;  
  
public class Main{  
    public static void main(String args[]){  
        int n ;  
        System.out.print("Enter the number of books:") ;  
        Scanner sc = new Scanner(System.in) ; n = sc.nextInt() ;  
        sc.nextLine() ;  
        Book books[] = new Book[n];  
        for(int i = 0 ; i<n ; i++){  
            System.out.print("Enter the book name: ") ;  
            String name = sc.nextLine() ;  
  
            System.out.print("Enter the author name:") ;  
            String author = sc.nextLine() ;  
            System.out.println("Enter the price of the book:") ;  
            int price = sc.nextInt() ;  
            System.out.println("Enter the number of pages in the book:") ;  
            int numPages = sc.nextInt() ;  
            sc.nextLine() ;  
  
            books[i] = new Book(name,author,price,numPages) ;  
        }  
  
        System.out.println("");  
        for(int i = 0 ; i<n ; i++){  
            System.out.println(books[i].toString()) ;  
        }  
        System.out.println("KATHI ABHILASH" ) ;  
        System.out.print("1BM23CS143") ;  
        sc.close();  
    }  
}  
  
class Book{  
    String name , author ;  
    int price , numPages ;  
  
    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 ;
}
}

```

## Output:

```

D:\java\LAB 3>javac Main.java
D:\java\LAB 3>java Main
Enter the number of books:3
Enter the book name: the art of war
Enter the author name:sun tzu
Enter the price of the book:
500
Enter the number of pages in the book:
256
Enter the book name: the subtle art of not giving a f*ck
Enter the author name:mark mason
Enter the price of the book:
300
Enter the number of pages in the book:
300
Enter the book name: harry potter
Enter the author name:jk rowling
Enter the price of the book:
600
Enter the number of pages in the book:
560
Book name: the art of war
Author name: sun tzu
Price: 500
Number of pages: 256
Book name: the subtle art of not giving a f*ck
Author name: mark mason
Price: 300
Number of pages: 300
Book name: harry potter
Author name: jk rowling
Price: 600
Number of pages: 560
KATHI ABHILASH
1BM23CS143

```

#### **Program 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.

#### **Algorithm:**

7) 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.

Code:-

```
import java.util.Scanner;  
abstract class Shapes {  
    int dimension1, dimension2;  
    abstract void printArea();  
}  
  
class Rectangle extends Shapes {  
    Rectangle(int length, int breadth) {  
        this.dimension1 = length;  
        this.dimension2 = breadth;  
    }  
  
    void printArea() {  
        int result = dimension1 * dimension2;  
        System.out.println("Area of the rectangle: " + result);  
    }  
}  
  
class Triangle extends Shapes {  
    Triangle(int base, int height) {  
        this.dimension1 = base;  
        this.dimension2 = height;  
    }  
  
    void printArea() {  
        double result = 0.5 * dimension1 * dimension2;  
        System.out.println("Area of the triangle: " + result);  
    }  
}
```

System  
{  
}  
class  
Circle  
this.  
{  
void  
doub  
Syste  
{  
}  
class  
publis  
Sc  
int  
Sy  
len  
bre  
Sy  
be  
he  
Sy  
()

abstract  
o integer  
Area(),  
> Triangle  
classes  
the  
Area()

```
System.out.println("Area of the Triangle: "+result);
}

class Circle extends Shapes {
    Circle(int radius) {
        this.dimension = radius;
    }

    void printArea() {
        double result = 3.14 * dimension * dimension;
        System.out.println("Area of Circle: " + result);
    }
}

class Area {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        int length, breadth, base, height, radius;
        System.out.println("Enter the dimensions of Rectangle:");
        length = s.nextInt();
        breadth = s.nextInt();
        System.out.println("Enter the dimensions of Triangle:");
        base = s.nextInt();
        height = s.nextInt();
        System.out.println("Enter the dimension of circle:");
        Circle cir = new Circle(radius);
        radius = s.nextInt();
        Rectangle rect = new Rectangle(length, breadth);
        Triangle tri = new Triangle(base, height);
        rect.printArea();
        tri.printArea();
        cir.printArea();
        s.close();
    }
}
```

```
System.out.println("Name: Abhilash Kathi");
```

```
System.out.println("USN: 1BM23(S143)");
```

8

Output:

Enter the dimensions of Rectangle:

2

8

Enter the dimensions of Triangle:

8

7

Enter the radius of Circle:

5

Area of the Rectangle: 16

Area of the Triangle: 28.0

Area of the Circle: 78.5

Name: Abhilash Kathi

USN: 1BM23(S143)

5) Develop

class

its

other

compon

no

prov

Curre

minin

below

imp

Code

imp

class

Stri

Stri

Stri

doc

Acc

th

th

thi

thi

3

vi

i

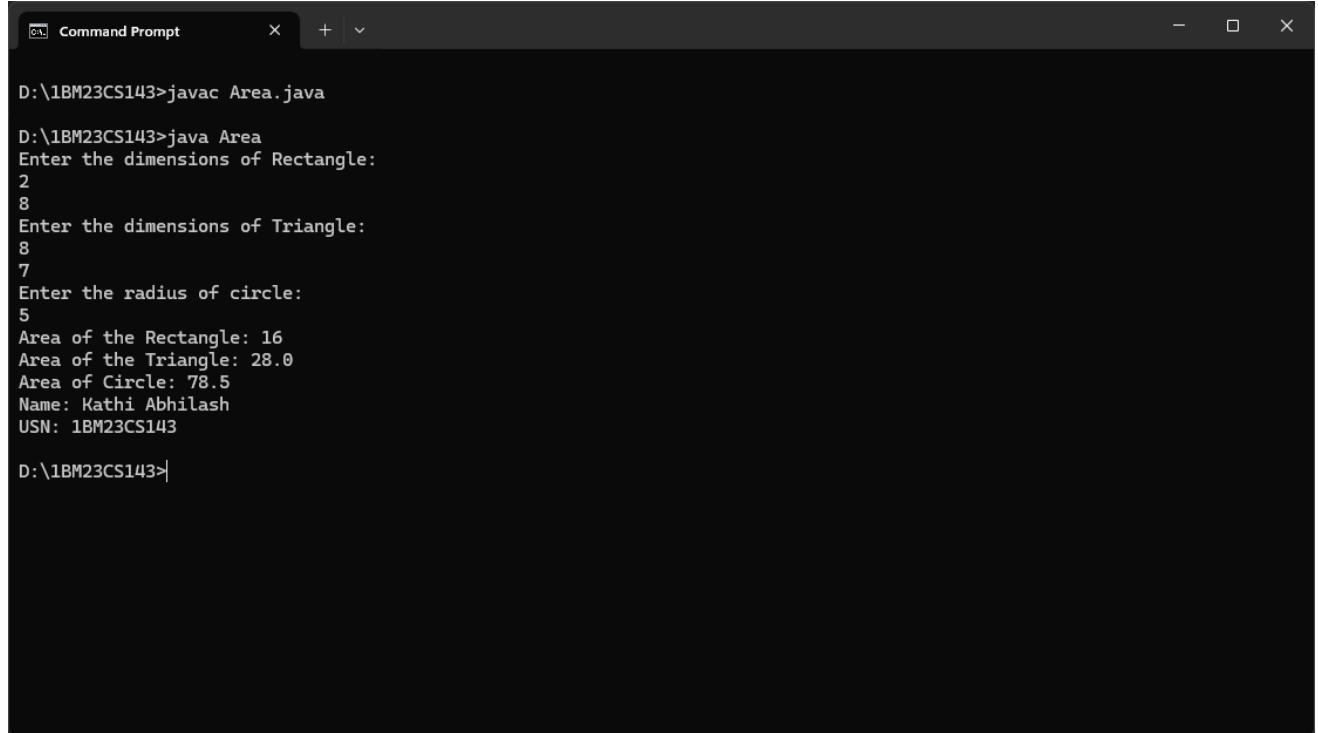
b

**Code:**

```
import java.util.Scanner;
abstract class Shapes {
int dimension1,dimension2;
abstract void printArea();
}
class Rectangle extends Shapes{
Rectangle(int length,int breadth) {
this.dimension1 = length;
this.dimension2 = breadth;
}
void printArea() {
int result = dimension1*dimension2;
System.out.println("Area of the Rectangle: " + result);
}
}
class Triangle extends Shapes{
Triangle(int base,int height) {
this.dimension1 = base;
this.dimension2 = height;
}
void printArea() {
double result = 0.5*dimension1*dimension2;
System.out.println("Area of the Triangle: " + result);
}
}
class Circle extends Shapes{
Circle(int radius) {
this.dimension1 = radius;
}
void printArea() {
double result = 3.14*dimension1*dimension1;
System.out.println("Area of Circle: " + result);
}
}
class Area {
public static void main(String args[]) {
Scanner s = new Scanner(System.in);
int length,breadth,base,height,radius;
System.out.println("Enter the dimensions of Rectangle: ");
length = s.nextInt();
breadth = s.nextInt();
Rectangle rect = new Rectangle(length,breadth);
System.out.println("Enter the dimensions of Triangle: ");
base = s.nextInt();
height = s.nextInt();
Triangle tri = new Triangle(base,height);
```

```
System.out.println("Enter the radius of circle: ");
radius = s.nextInt();
Circle cir = new Circle(radius);
rect.printArea();
tri.printArea();
cir.printArea();
s.close();
System.out.println("Name: Kathi Abhilash");
System.out.println("USN: 1BM23CS143");
}
}
```

### Output:



D:\1BM23CS143>javac Area.java

D:\1BM23CS143>java Area

Enter the dimensions of Rectangle:

2

8

Enter the dimensions of Triangle:

8

7

Enter the radius of circle:

5

Area of the Rectangle: 16

Area of the Triangle: 28.0

Area of Circle: 78.5

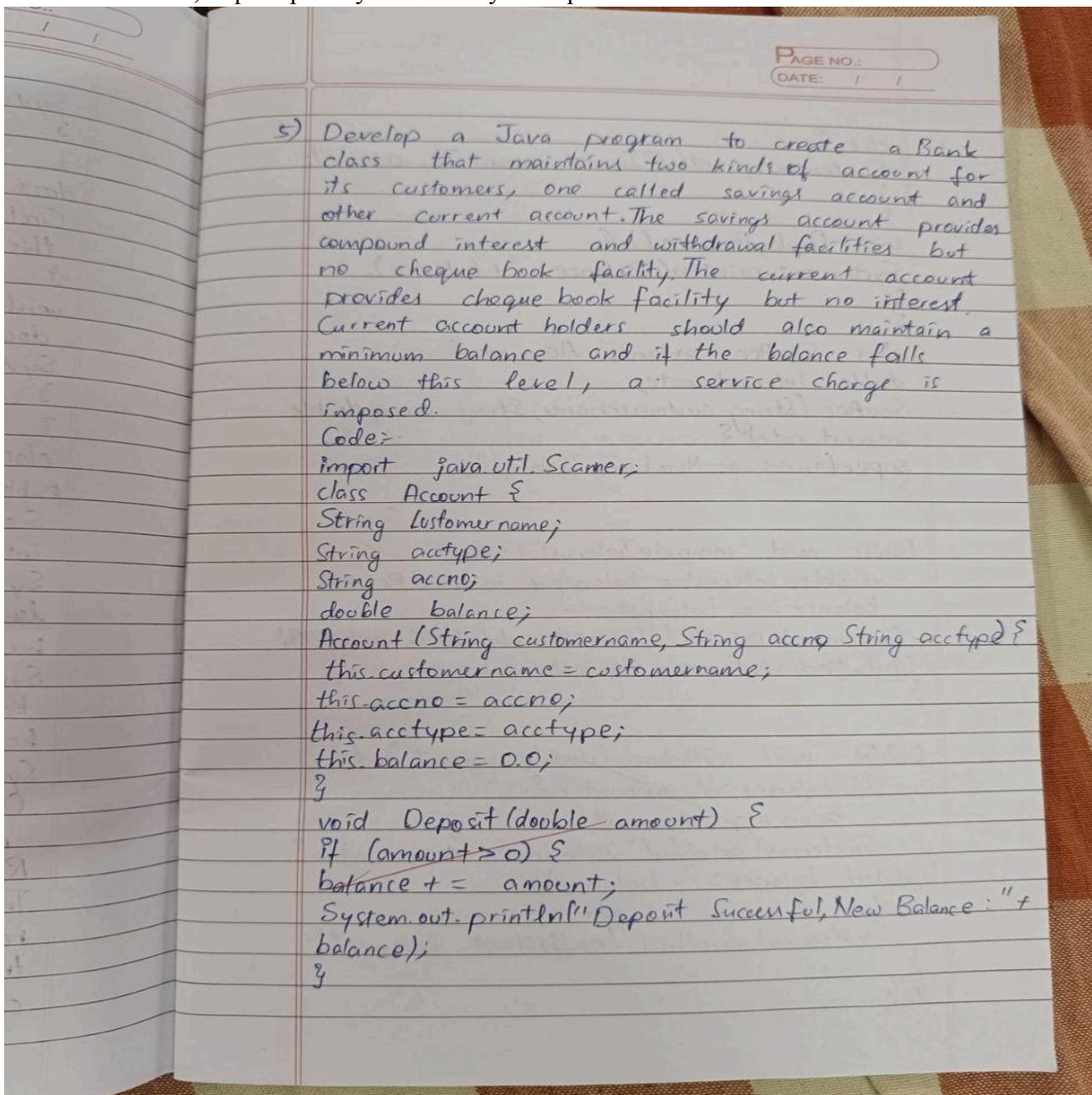
Name: Kathi Abhilash

USN: 1BM23CS143

D:\1BM23CS143>

### **Program 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.



```
else {
    System.out.println("Invalid amount Try again");
}

void DisplayBalance() {
    System.out.println("Balance : " + balance);
}

class SavAcc extends Account {
    double interestRate;
    SavAcc (String customername, String accno, double interestRate) {
        super(name, accNumber, "Savings");
    }

    public void computeInterest () {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added: " + interest);
        Updated balance: " + balance);
    }

    public void withdraw (double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount + ".");
            Updated balance: " + balance);
        } else {
            System.out.println ("Insufficient balance");
        }
    }
}
```

class  
double  
double  
curAcc  
S  
4  
Public  
if  
System  
char  
bola  
3  
3

Sys  
ba  
ch  
3

Co

3

3

3

3

1

26

```
class curAccount extends Account {  
    double minBalance = 500.0;  
    double servicecharge = 50.0;  
    curAccount(String name, int accNumber) {  
        super(name, accNumber, "current");  
    }  
}
```

```
public void checkMinBalance() {  
    if (balance < minBalance) {  
        balance -= servicecharge;  
        System.out.println("Balance below minimum Service  
charge imposed: " + servicecharge + ". Updated  
balance: " + balance);  
    }  
}
```

```
public void withdraw(double amount) {  
    if (balance >= amount) {  
        balance -= amount;  
        System.out.println("Withdrawn: " + amount + ". Updated  
balance: " + balance);  
        checkMinBalance();  
    } else {  
        System.out.println("Insufficient balance");  
    }  
}
```

```

Public class Bank {
    Public static void main (String [] args) {
        PrintInfo.print();
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter customer name");
        String name = sc.nextLine();
        System.out.println ("Enter account num");
        int accountnumber = sc.nextInt();
        SavAccount savingsAccount = new SavAccount (
            name, accountnumber);
        System.out.println ("Enter customer name");
        String name = sc.nextLine();
        System.out.println ("Enter account num");
        int accountnumber1 = sc.nextInt();
        curAccount currentaccount = new CurAccount (name,
            accountnumber1);
    }

    while (true) {
        System.out.println ("\n..... Menu .....");
        System.out.println ("1. Deposit\n2. Withdraw\n3. Compute
Interest for savings Account\n4. Display Account
Details\n5. Exit\n");
        System.out.println ("Enter your choice .");
        int choice = sc.nextInt();
        System.out.println ("Enter the type of account ");
        String accType = sc.nextLine();
        if (accType.equals ("Savings")) {
            switch (choice) {
                Case 1:
                    System.out.println ("Enter the deposit amount");
                    double depositAmount = sc.nextDouble();
                Case 2:
                    System.out.println ("Enter the withdraw amount");
                    double withdrawAmount = sc.nextDouble();
                    if (depositAmount > withdrawAmount) {
                        System.out.println ("Withdrawal successful");
                    } else {
                        System.out.println ("Insufficient funds");
                    }
                Case 3:
                    System.out.println ("Interest calculated");
                Case 4:
                    System.out.println ("Displaying account details");
                Case 5:
                    System.out.println ("Exiting program");
                    break;
            }
        }
    }
}

```

SavingAcc  
break;

case 2:  
System.out

double wit  
SavingAc  
break;

case 3:  
SavingA  
break;

case 4:  
System  
customer  
System  
accoun  
System  
accoun  
System  
accoun  
Saving  
break;

case 5:  
Syst  
bre

detau  
Syr  
?

3  
2

```

Public class Bank {
    Public static void main (String [] args) {
        PrintInfo.print();
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter customer name");
        String name = sc.nextLine();
        System.out.println ("Enter account num");
        int accountnumber = sc.nextInt();
        SavAccount savingsAccount = new SavAccount (
            name, accountnumber);
        System.out.println ("Enter customer name");
        String name = sc.nextLine();
        System.out.println ("Enter account num");
        int accountnumber1 = sc.nextInt();
        curAccount currentaccount = new CurAccount (name,
            accountnumber1);
        while (true) {
            System.out.println ("\n..... Menu .....");
            System.out.println ("1. Deposit\n2. Withdraw\n3. Compute
Interest for savings Account\n4. Display Account
Details\n5. Exit\n");
            System.out.println ("Enter your choice .");
            int choice = sc.nextInt();
            System.out.println ("Enter the type of account ");
            String accType = sc.nextLine();
            if (accType.equals ("Savings")) {
                switch (choice) {
                    Case 1:
                        System.out.println ("Enter the deposit amount");
                        double depositAmount = sc.nextDouble();
                }
            }
        }
    }
}

```

SavingAcc  
break;

case 2:  
System.out

double wi  
SavingAc  
break;

case 3:  
SavingA  
break;

case 4:  
System  
customer  
System  
accoun  
System  
accoun  
System  
accoun  
Saving  
break;

case 5  
Syst  
bre

detau  
Syr  
?

3  
2

```
else if (accType.equals("current")) {  
    switch (choice) {  
        case 1:  
            System.out.print("Enter the deposit amount: ");  
            double depositAmount = sc.nextDouble();  
            currentAccount.deposit(depositAmount);  
            break;  
    }  
}
```

```
case 2:  
    System.out.println("Enter the withdrawal amount: ");  
    double withdrawalAmount = sc.nextDouble();  
    currentAccount.withdraw(withdrawalAmount);  
    break;
```

```
case 3:  
    System.out.println("current's account do not earn  
    interest");  
    break;
```

```
case 4:  
    System.out.println("Customer name: " + currentAccount.  
    customerName);  
    System.out.println("Account number: " + currentAccount.  
    accountNumber);  
    System.out.println("Type of Account: " + currentAccount.  
    accountType);  
    break;
```

```
case 5:  
    System.exit(0);  
    break;
```

```
args) {  
    savingsAccount.deposit(depositAmount);  
    break;  
}
```

### Case 2:

```
System.out.println("Enter withdrawal amount");  
double withdrawlAmount = sc.nextDouble();
```

double withdrawalAmount = sc.nextInt();  
Scanner accountWithdrawal(withdrawal);

Saving Money will make a  
break;

## Car Account:

case 3:

SavingsAccount.computeInterest();

break;

(names)

(Case 4):

```
System.out.println("Customer name : "+ SavingAccount  
customerName);
```

```
System.out.println("Account number: " + SavingsAccount.  
accountNumber);
```

```
System.out.println("Type of Account: "+ SavingAccount.  
accounttype);
```

Savings Account. display Balance();

break;

### 3. Compute count

comes:

```
System.exit(0);
```

~~break~~

1

### details

```
System.out.println("Invalid choice");
```

3

2

~~ount "ji~~



(DATE: / /)

Deposited : 10000 Updated balance: 100000.

1. Deposit

2. Withdraw

3. compute Interest for savings Account

4. Display Account details

5. Exit

Enter your choice: 2

Enter the type of account (saving)/current).

Enter withdrawal amount: 15000

Withdrawn: 15000 Updated balance: 67000

Lab:

i) Crea

stu

me

der

st

co

st

th

ob

st

o

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

c

i

+

f

&lt;p

**Code:**

```
import java.util.Scanner;

public class Bank {
    static Scanner sc = new Scanner(System.in);
    Account ob1;

    void createAccount() {
        String customer;
        int account;
        String type;
        int initBal;

        System.out.print("Enter the customer name: ");
        customer = sc.nextLine();
        System.out.print("Enter account Number: ");
        account = sc.nextInt();
        sc.nextLine(); // Consume the newline
        System.out.print("Enter Account type (Savings or Current): ");
        type = sc.nextLine();
        System.out.print("Enter the initial Balance: ");
        initBal = sc.nextInt();

        if (type.equals("Savings")) {
            ob1 = new Savings(customer, account, initBal);
        } else {
            ob1 = new Current(customer, account, initBal);
        }
    }

    public static void main(String[] args) {
        Bank bank = new Bank();
        bank.createAccount();
        System.out.println("Kathi Abhilash 1BM23CS143");

        while (true) {
            System.out.println("-----MENU-----");
            System.out.println("1. Deposit   2. Withdraw");
            System.out.println("3. Compute interest");
            System.out.println("4. Display account details");
            System.out.println("5. exit " );
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    bank.ob1.deposit();
                    break;
            }
        }
    }
}
```

```

        case 2:
            bank.ob1.withdraw();
            break;
        case 3:
            if (bank.ob1 instanceof Savings) {
                ((Savings) bank.ob1).computeInterest();
            } else {
                System.out.println("Interest computation is only available for Savings accounts.");
            }
            break;
        case 4:
            bank.ob1.display();
            break;
        case 5:
            break ;
    default:
        System.out.println("Invalid choice. Please try again.");
    }
    if(choice == 5) break ;
}
}
}

class Account {
    String customerName;
    int accountNumber;
    int balance;

    Account(String customer, int accountNum, int bal) {
        customerName = customer;
        accountNumber = accountNum;
        balance = bal;
    }

    void deposit() {
        System.out.print("Enter the amount to deposit: ");
        int amt = Bank.sc.nextInt();
        balance += amt;
        System.out.println("Deposited: " + amt + ", New Balance: " + balance);
    }

    void withdraw() {
        System.out.print("Enter the amount to withdraw: ");
        int amt = Bank.sc.nextInt();
        if (balance - amt < 0) {
            System.out.println("Insufficient Balance to withdraw the given amount.");
        } else {

```

```

        balance -= amt;
        System.out.println("Amount of " + amt + " withdrawn successfully. Current Balance is " +
balance);
    }
}

void display() {
    System.out.println("The Balance in the account is " + balance);
}
}

class Savings extends Account {
    double interestPercent;

    Savings(String customer, int accountNum, int bal) {
        super(customer, accountNum, bal);
        System.out.print("Enter the interest percentage on the account: ");
        interestPercent = Bank.sc.nextDouble();
    }

    void computeInterest() {
        balance += balance * (interestPercent / 100);
        System.out.println("Amount after applying interest is: " + balance);
    }
}

class Current extends Account {
    int minBalance = 1000;

    Current(String customer, int accountNum, int bal) {
        super(customer, accountNum, bal);
    }

    void withdraw() {
        System.out.print("Enter the amount to withdraw: ");
        int amt = Bank.sc.nextInt();
        if (balance - amt < minBalance) {
            System.out.println("Insufficient Balance to maintain the minimum required.");
        } else {
            balance -= amt;
            System.out.println("Amount of " + amt + " withdrawn successfully. Current Balance is " +
balance);
        }
    }
}

```

## Output:

```
Command Prompt + X - O X

D:\java\LAB 5>javac Bank.java
D:\java\LAB 5>java Bank
Enter the customer name: abhilash
Enter account Number: 343
Enter Account type (Savings or Current): Savings
Enter the initial Balance: 50000
Enter the interest percentage on the account: 3
Kathi Abhilash 1BM23CS143
-----MENU-----
1. Deposit      2. Withdraw
3. Compute interest
4. Display account details
5. exit
4
The Balance in the account is 50000
-----MENU-----
1. Deposit      2. Withdraw
3. Compute interest
4. Display account details
5. exit
2
Enter the amount to withdraw: 3000
Amount of 3000 withdrawn successfully. Current Balance is 47000
-----MENU-----
1. Deposit      2. Withdraw
3. Compute interest
4. Display account details
5. exit
3
Amount after applying interest is: 48410
-----MENU-----
1. Deposit      2. Withdraw
3. Compute interest
4. Display account details
5. exit
5
D:\java\LAB 5>
```

## Program 6

Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals 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 Personal. 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.

### **Algorithm:**

PAGE NO.: / /  
DATE: / /

Lab-06

i) Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals 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 Personal. 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.

Code:

```
package CIE;
public class Personal {
    public string USN, name;
    public int sem;

    public Personal(String USN, String name, int sem) {
        this.USN = USN;
        this.name = name;
        this.sem = sem;
    }
}

package SEE;
import CIE.Personal;
```

```
public class External extends Personal {
    public int[] seeMarks;
```

```
public External(String usn, String name) {
    super(usn, name, sem);
    seeMarks = new int[5];
```

③

```
public void setMarks(int[] marks) {
    if (marks.length == 5) {
        System.arraycopy(marks, 0, 5);
        System.out.println("Please provide marks");
```

④

```
import CIE.*;
import SFE.*;
```

```
import java.util.Scanner;
public class FinalMarks {
```

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

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

```
        System.out.print("Enter the number of students");
```

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

```
        sc.nextLine();
```

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

```
        Internal[] internalMarks = new Internal[n];
```

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

```
            System.out.println("Student " + (i + 1) + ":");
```

```
            System.out.println("USN " + students[i].usn);
```

```
            System.out.println("Name " + students[i].name);
```

}

```
for (int i=0; i<5; i++) {  
    int finalMark = internalMark[i].internalMarks[i] +  
        students[i].seeMarks[i];  
    System.out.println(finalMark);
```

Output:-

Enter number of students: 5

Enter details for students:

USN: 123

Name: John Doe

Sem: 3

Enter Internal marks:

15 18 20 19 17

Enter SEF marks:

80 90 70 85 95

Final marks of students:

Students: 1 - 123 John Doe

USN: 123

Semester: 3

Final marks course wise:

85 63 55 61 54

**Code:**

```
import CIE.Internals;
import SEE.Externals;
import CIE.Student;
import java.util.Scanner;

class PrintInfo {
    static void print() {
        System.out.println("Name: Kathi Abhilash");
        System.out.println("USN: 1BM23CS143");
    }
}

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

        System.out.println("Enter the number of students: ");
        int n = sc.nextInt();

        Internals[] cieStudents = new Internals[n];
        Externals[] seeStudents = new Externals[n];
        Student[] students = new Student[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Student " + (i + 1) + ":");

            cieStudents[i] = new Internals();
            seeStudents[i] = new Externals();
            students[i] = new Student();

            students[i].inputStudentDetails();
            cieStudents[i].inputCIEmarks();
            seeStudents[i].inputSEEmarks();
        }

        System.out.println("\nFinal Results:");
        for (int i = 0; i < n; i++) {
            students[i].calcFinalMarks(cieStudents[i], seeStudents[i]);
        }
    }
}

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);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}

package CIE;

public class Internals extends Student {

    public int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}

```

```
}
```

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

## Output:

```
D:\LAB 6>javac Main.java  
D:\LAB 6>java Main  
Name: Kathi Abhilash  
USN: 1B123CS143  
Enter the number of students:  
2  
Enter details for Student 1:  
Enter USN: 149  
Enter Name: karthik  
Enter Semester: 3  
Enter CIE marks for 5 subjects:  
Subject 1: 33  
Subject 2: 44  
Subject 3: 43  
Subject 4: 34  
Subject 5: 45  
Enter SEE marks for 5 subjects:  
Subject 1: 88  
Subject 2: 99  
Subject 3: 77  
Subject 4: 66  
Subject 5: 55  
Enter details for Student 2:  
Enter USN: 102  
Enter Name: harsha  
Enter Semester: 5  
Enter CIE marks for 5 subjects:  
Subject 1: 3  
Subject 2: 43  
Subject 3: 43  
Subject 4: 43  
Subject 5: 43  
Enter SEE marks for 5 subjects:  
Subject 1: 78  
Subject 2: 86  
Subject 3: 77  
Subject 4: 56  
Subject 5: 99  
Final Results:  
USN: 149  
Name: karthik  
Semester: 3  
Final Marks for 5 subjects:  
Subject 1: CIE = 33, SEE = 88, Final = 77  
Subject 2: CIE = 44, SEE = 99, Final = 93  
Subject 3: CIE = 43, SEE = 77, Final = 81  
Subject 4: CIE = 34, SEE = 66, Final = 67  
Subject 5: CIE = 45, SEE = 55, Final = 72  
USN: 102  
Name: harsha  
Semester: 5  
Final Marks for 5 subjects:  
Subject 1: CIE = 3, SEE = 78, Final = 42  
Subject 2: CIE = 43, SEE = 86, Final = 86  
Subject 3: CIE = 43, SEE = 77, Final = 81  
Subject 4: CIE = 43, SEE = 56, Final = 71  
Subject 5: CIE = 43, SEE = 99, Final = 92  
D:\LAB 6>
```

## Program 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=father's age.

Algorithm:

Lab - 07

Q) write a program that demonstrate handling of exception in inheritance tree. Create a base class called Father & derived class called son which extends the base class. In son class, implement a constructor that use both Father & son.

Code:

```
class WrongAgeException extends Exception {  
    public WrongAgeException(String message) {  
        super(message);  
    }  
  
    class Father {  
        int age;  
        public Father(int age) throws WrongAgeException {  
            if (age < 0) {  
                throw new WrongAgeException("Father's age cannot be negative");  
            }  
            this.age = age;  
            System.out.println("Father's Age is " + this.age);  
        }  
  
        class Son extends Father {  
            int sonAge;  
            if (age < 0) {  
                throw new WrongAgeException("Son's age cannot be negative");  
            }  
        }  
    }  
}
```

if (sonAge >= fatherAge) {  
 throw new WrongAgeException("Son age  
cannot be equal to or greater than father");  
}

public class ExceptionInheritance {  
 public static void main (String args[]) {  
 try {  
 System.out.println("Creating Father & Son objects");  
 Father father = new Father(40);  
 Son son = new Son(40, 15);  
 }  
 }  
}

catch (WrongAgeException e) {  
 System.out.println("Exception caught "+ e.getMessage());  
}  
sc.close;  
}  
}

Output:-

Creating Father & Son objects:

Father's age is set to: 40

Son's age is set to: 15

Testing ~~invalid~~ inputs

Attempting to set father's age to -5

\*Exception caught: Father's age cannot be negative.

Attempting to set son's age to 45 and father's age to 40

\*Exception caught: Son's age cannot be greater than or equal to father's age.

Code:

```
import java.util.Scanner;

class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class Father {
    protected int fatherAge;

    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Father's age cannot be negative!");
        }
        this.fatherAge = age;
        System.out.println("Father's age is: " + fatherAge);
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be negative!");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to father's age!");
        }
        this.sonAge = sonAge;
        System.out.println("Son's age is: " + sonAge);
    }
}

public class ExceptionInInheritance {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter father's age: ");
            int fatherAge = scanner.nextInt();
            System.out.print("Enter son's age: ");
            int sonAge = scanner.nextInt();
        }
    }
}
```

```
        Son son = new Son(fatherAge, sonAge);
    } catch (WrongAgeException e) {
        System.err.println("Exception: " + e.getMessage());
    } catch (Exception e) {
        System.err.println("Invalid input! Please enter integers.");
    } finally {
        scanner.close();
    }
    System.out.println("Kathi Abhilash 1BM23CS143");
}
}
```

### Output:

```
D:\java\LAB 7>javac ExceptionInInheritance.java
D:\java\LAB 7>java ExceptionInInheritance
Enter father's age: 45
Enter son's age: 40
Father's age is: 45
Son's age is: 40
Kathi Abhilash 1BM23CS143

D:\java\LAB 7>java ExceptionInInheritance
Enter father's age: -40
Enter son's age: 55
Exception: Father's age cannot be negative!
Kathi Abhilash 1BM23CS143

D:\java\LAB 7>java ExceptionInInheritance
Enter father's age: 40
Enter son's age: 55
Father's age is: 40
Exception: Son's age cannot be greater than or equal to father's age!
Kathi Abhilash 1BM23CS143

D:\java\LAB 7>
```

## **Program 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.

Algorithm:

Lab-8

Q) Write a program to which create threads, one thread displaying "BMS College of Engineering" once every ten seconds & other displaying "CSE" every two seconds.

Code:

```
class CollegeThread extends Thread {  
    public void run() {  
        try {  
            while(true) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("College Thread interrupted");  
            e.getMessage();  
        }  
    }  
  
    class DepartmentThread extends Thread {  
        public void run() {  
            try {  
                while(true) {  
                    System.out.println("CSE");  
                    Thread.sleep(2000);  
                }  
            } catch (InterruptedException e) {  
                System.out.println("Department Thread interrupted");  
            }  
        }  
    }  
}
```

threads  
Engineering).  
displaying

" + e.getMessage());  
3  
4  
5  
6  
public class MultiThread {  
 public static void main (String [] args) {  
 CollegeThread collegeThread = new CollegeThread ();  
 DepartmentThread departmentThread = new Department  
 Thread ();  
 collegeThread.start ();  
 departmentThread.start ();  
 }  
}

Engineering);

Output:

CSE

CSF

CSE

CSE

BMS College of Engineering

CSE

CSE

CSF

CSE

interrupted");

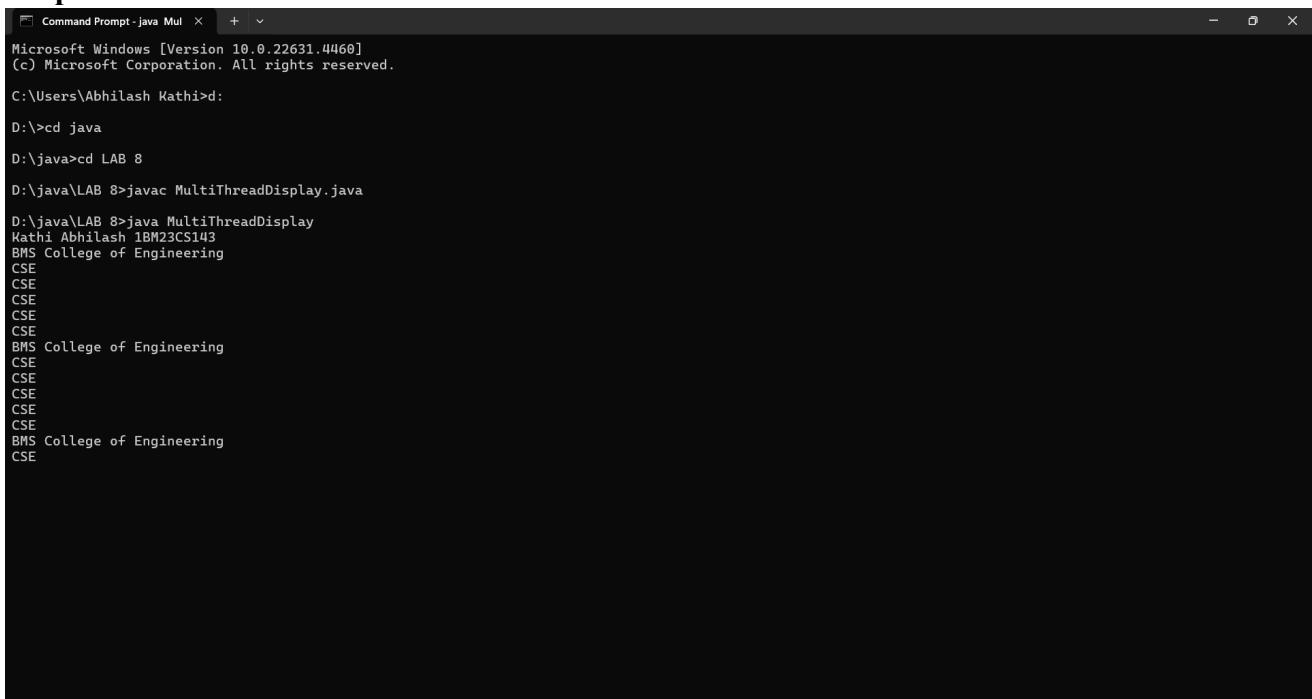
DS

of interrupted:

Code:

```
class DisplayMessage extends Thread {  
    private String message;  
    private int delay;  
  
    public DisplayMessage(String message, int delay) {  
        this.message = message;  
        this.delay = delay;  
    }  
  
    public void run() {  
        try {  
            while (true) {  
                System.out.println(message);  
                Thread.sleep(delay);  
            }  
        } catch (InterruptedException e) {  
            System.err.println("Thread interrupted: " + e.getMessage());  
        }  
    }  
}  
  
public class MultiThreadDisplay {  
    public static void main(String[] args) {  
        DisplayMessage thread1 = new DisplayMessage("BMS College of Engineering", 10000);  
        DisplayMessage thread2 = new DisplayMessage("CSE", 2000);  
        thread1.start();  
        thread2.start();  
        System.out.println("Kathi Abhilash 1BM23CS143");  
    }  
}
```

## Output:



```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Abhilash Kathi>d:
D:>cd java
D:\java>cd LAB 8
D:\java\LAB 8>javac MultiThreadDisplay.java
D:\java\LAB 8>java MultiThreadDisplay
Kathi Abhilash 1BM23CS143
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
```

## **Program 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 ArithmeticException. Display the exception in a message dialog box.

### **Algorithm:**

Lab-09

Q Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 & Num2. The division of Num1 & 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.

Code:

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

```
JLabel jlab = new JLabel("Enter the divisor and dividend");
```

```
JTextField ajtf = new JTextField(8);
```

```
JTextField bjtf = new JTextField(8);
```

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

```
JLabel err = new JLabel();
```

```
JLabel alab = new JLabel();
```

```
JLabel blab = new JLabel();
```

```
JLabel anslab = new JLabel();
```

nterface to  
two  
m2. The  
ed in the  
clicked. If  
er, the  
ception.  
d throw an  
tion in a

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

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

ajtf.addActionListener(I);
bjtf.addActionListener(I);
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(ArithemeticException e) {  
        alab.setText("");  
        blab.setText(" ");  
        anslab.setText(" ");  
        err.setText("B should be NON zero!");  
        }  
    }  
ifrm.setVisible(true);
```

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

white dialog  
white box

Output: Enter the divider and dividend

30

5

A=30

B=5

Ans=6

**Code:**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jlab = new JLabel("Enter the dividend and divisor:");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button = new JButton("Calculate");
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        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) {
                try {
                    int a = Integer.parseInt(ajtf.getText());
                    int b = Integer.parseInt(bjtf.getText());
                    int ans = a / b;

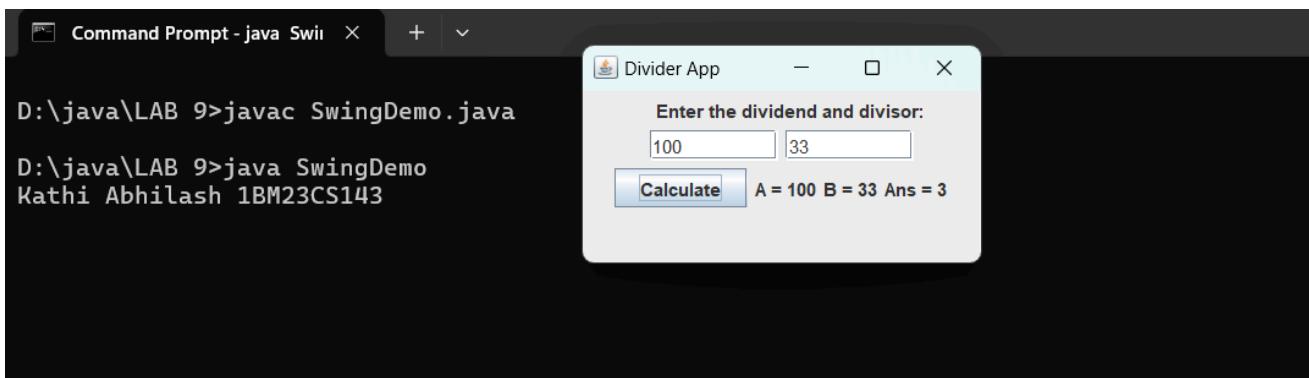
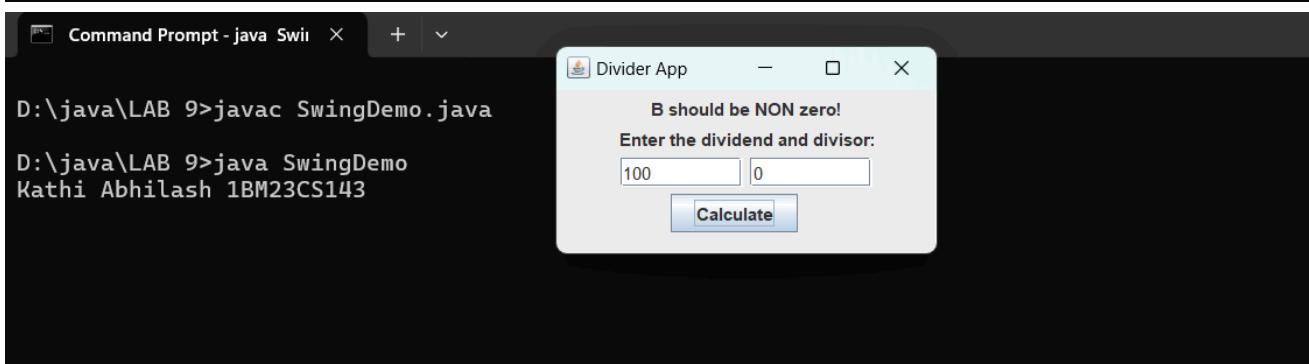
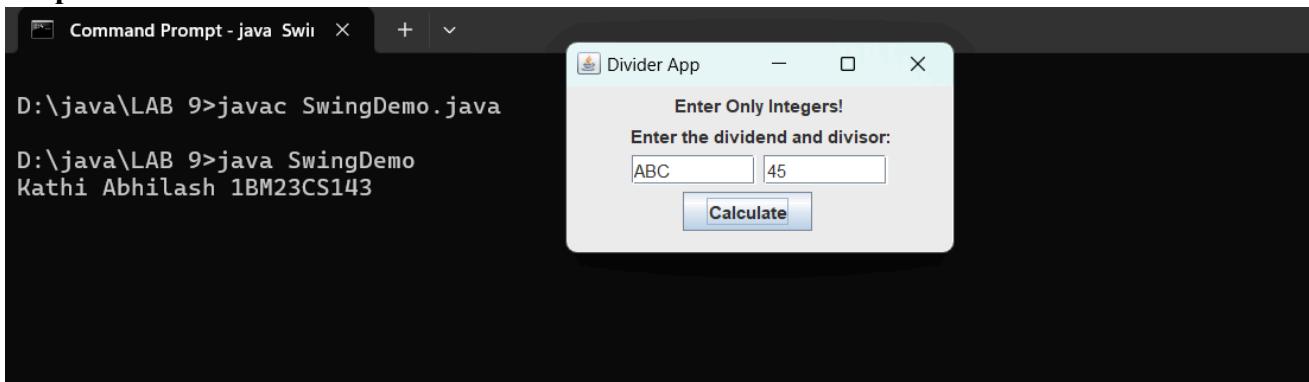
                    alab.setText("A = " + a);
                    blab.setText("B = " + b);
                    anslab.setText("Ans = " + ans);
                    err.setText("");
                } catch (NumberFormatException e) {
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("Enter Only Integers!");
                } catch (ArithmaticException e) {
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("B should be NON zero!");
                }
            }
        });
    }
}
```

```

        }
    });
jfrm.setVisible(true);
}
public static void main(String args[]) {
    System.out.println("Kathi Abhilash 1BM23CS143");
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDemo();
        }
    });
}

```

**Output:**



### Program 10:

Demonstrate Inter process Communication and deadlock

IPC

Algorithm:

DATE: / /

Lab - 10

Q) Demonstrate Inter process Communication and deadlock.

a) Code:

```
Class Q {
    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("Interrupted exception caught");
            }
        }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("In Intimate Producer\n");
        notify();
        return n;
    }

    synchronized void put(int n) {
        while (valueSet) {
            try {
                System.out.println("In Producer waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("Interrupted Exception caught");
            }
        }
    }
}
```

```
this.n = n;  
valueSet = true;  
System.out.println("Put: " + n);  
System.out.println("In Intimate Consumer (" + n + ");  
notify();
```

3

3

```
class Producer implements Runnable {
```

```
    Queue q;
```

```
    Producer(Q q) {
```

```
        this.q = q;
```

```
        new Thread(this, "Producer").start();
```

3

```
public void run() {
```

```
    int i = 0;
```

```
    while (i < 15) {
```

```
        q.put(i++);
```

3

3

3

```
class Consumer implements Runnable {
```

```
    Queue q;
```

```
    Consumer(Q q) {
```

```
        this.q = q;
```

```
        new Thread(this, "Consumer").start();
```

3

```
public void run() {
```

```
    int i = 0;
```

```
    while (i < 15) {
```

```
        int r = q.get();
```

Sy

2

3

public

P

Si

C

r

c

3

3

Wso

F

G

F

O

P

g

P

g

P

g

8

System.out.println("Consumed: " + r);  
    i++

    3  
    3  
    3

public class PCFixed {  
    public static void main(String args[]) {  
        System.out.println("Kathi Abhilash IBM23(S143)");  
        Q q = new Q();  
        new Producer(q);  
        new Consumer(q);  
        System.out.println("Press Control-C to stop");  
    } }  
    3

Wrote output

Output: Got 0

put: 1

Got: 1

put: 2

got: 2

put: 3

got: 3

put: 4

got: 4

put: 5

got: 5

**Code:**

```
class Q {  
    int n;  
    boolean valueSet = false;  
  
    synchronized int get() {  
        while (!valueSet) {  
            try {  
                System.out.println("\nConsumer waiting\n");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        }  
        System.out.println("Got: " + n);  
        valueSet = false;  
        System.out.println("\nIntimate Producer\n");  
        notify();  
        return n;  
    }  
  
    synchronized void put(int n) {  
        while (valueSet) {  
            try {  
                System.out.println("\nProducer waiting\n");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        }  
        this.n = n;  
        valueSet = true;  
        System.out.println("Put: " + n);  
        System.out.println("\nIntimate Consumer\n");  
        notify();  
    }  
}  
  
class Producer implements Runnable {  
    Q q;  
  
    Producer(Q q) {  
        this.q = q;  
        new Thread(this, "Producer").start();  
    }  
  
    public void run() {
```

```

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

class Consumer implements Runnable {
    Q q;

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

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

public class PCFixed {
    public static void main(String args[]) {
        System.out.println("Kathi Abhilash 1BM23CS143");
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
        System.out.println("Kathi Abhilash 1BM23CS143");
    }
}

```

## Output:

```
D:\java\LAB 10\IPC>javac PCFixed.java
D:\java\LAB 10\IPC>java PCFixed
Kathi Abhilash 1BM23CS143
Press Control-C to stop.
Kathi Abhilash 1BM23CS143
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

```

```
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
Got: 7
Intimate Producer
```

## Deadlock:

**Algorithm:**

b) Code+

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 B.last");

```

3
3
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 arg1[]) {
        new Deadlock();
        System.out.println("Kothi Abhilash IBM23CS143");
    }
}

```

d().getName());  
bar");

Output:-

RacingThread entered B.bar

MainThread entered A.foo

~~RacingThread entered trying to call A.foo()~~

MainThread trying to call B.bar()

Inside A.last

Inside B.last

Back in main thread

Kothi Abhilash IBM23CS143

Back in other thread

8  
02.12

**Code:**

```
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 B.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("Kathi Abhilash 1BM23CS143");
}
}

```

### Output:

```

D:\java\LAB 10\Deadlock>javac Deadlock.java
D:\java\LAB 10\Deadlock>java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
Inside A.last
MainThread trying to call B.last()
Back in other thread
Inside B.last
Back in main thread
Kathi Abhilash 1BM23CS143

D:\java\LAB 10\Deadlock>

```