

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Chethan N (**1BM23CS075**)

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 **CHEETHAN N (1BM23CS075)**, 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.

Prof. Swathi Sridharan Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
---	---

Index

Sl. No.	Date	Experiment Title	Page No.
1	1/10/24	Quadratic Equation	4-7
2	8/10/24	Calculating SGPA	8-12
3	15/10/24	Book Details	13-17
4	22/10/24	Abstract Class Shape	18-21
5	29/10/24	Bank Details	21-26
6	12/11/24	Packages	27-33
7	19/11/24	Interface	33-37
8	26/11/24	Exception Handling	37 -40
9	3/12/24	Threads	41-43
10	3/12/24	GUI – Java Swing	43-51

GitHub Link:

<https://github.com/chethannhub/OOJ-Clg>

Program 1

Implemente Quadratic Equation

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:

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

public class Quadratic {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter three numbers: ");
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();

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

        if (d < 0) {
            System.out.println("There are no real solutions");
        } else {
            double x1 = ((-b) + (Math.sqrt(d))) / 2*a;
            double x2 = (-b - (Math.sqrt(d))) / 2*a;
            System.out.println("The real solutions are: " + x1 + ", " + x2);
        }
    }
}
```

of Seen
Seen

Output

Enter three numbers:

1

-3

2

The real solutions are: 2.0, 1.0

Q8
seen

~~10/24~~

Code:

```
import java.util.Scanner;

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

        System.out.println("Enter three numbers:");
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();

        int d = (b * b) - (4 * a * c);
        if (d < 0) {
            System.out.println("There are no real solutions.");
        } else {
            double x1 = (-b + Math.sqrt(d)) / (2.0 * a);
            double x2 = (-b - Math.sqrt(d)) / (2.0 * a);
        }
    }
}
```

```
        System.out.println("The real solutions are: " + x1 + ", " + x2);
    }
}
}
```

Output:

```
D:\1BM23CS075\Lab1>javac Quadratic.java

D:\1BM23CS075\Lab1>java Quadratic.java
Enter three numbers:
1
-3
2
The real solutions are: 2.0, 1.0

D:\1BM23CS075\Lab1>
```

Program 2

Calculating SGPA

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.

Lab 2

4. Develop a Java program to create a class Student with members USN, an array credits & an array marks. Include methods to accept and display details & a method to calculate SGPA of a student.
import java.util.*;

→ public class student {

 String USN;

 int credits[];

 int marks[];

 void accept() {

 scanner sc = new scanner;

 System.out.println("Enter the USN: ");

 USN = sc.nextLine();

 System.out.println("Enter the credits: ");

 credits[] = sc.nextInt();

 System.out.println("Enter the marks: ");

 marks[] = sc.nextInt();

}

```

void sgpa() {
    int cred_total;
    int sub_total[];
    int total;
    int sgpa;
    for (int i=0; i<n; i++) {
        sub_total[i] = credits[i] * marks[i];
        total = total + sub_total[i];
        cred_total = cred_total + credits[i];
    }
    int sgpa = total / cred_total;
    System.out.println("SGPA : " + sgpa);
}

```

```

void detail()
void display() {
    System.out.println("USN : " + usn);
    System.out.println("Credits : ");
    for (int i=0; i<n; i++) {
        System.out.println("credits[" + i + "]");
    }
}

```

```

System.out.println("Marks : ");
for (int i=0; i<n; i++) {
    System.out.println(marks[i]);
}

```

sgpac();

3

```
public static void main(String args[]) {  
    accept();  
    display();  
}
```

3

~~seen~~

Output

Enter the USN : IBM23CS068

Enter the credits and marks.

Credits for subject 1 : 4

Marks for subject 1 : 97

Credits for subject 2 : 3

Marks for subject 2 : 83

Credits for subject 3 : 3

Marks for subject 3 : 94

Credits for subject 4 : 3

Marks for subject 4 : 91

Credits for subject 5 : 3

Marks for subject 5 : 81

Credits for subject 6 : 1

Marks for subject 6 : 86

Credits for subject 7 : 1

Marks for subject 7 : 98

Credits for subject 8 : 1

~~Marks~~ for subject 8 : 89

VSN : 1BMA3CS068

Credits for subject 1 : 4

Marks for subject 1 : 97

Credits for subject 2 : 3

Marks for subject 2 : 83

Credits for subject 3 : 3

Marks for subject 3 : 94

Credits for subject 4 : 3

Marks for subject 4 : 91

Credits for subject 5 : 3

Marks for subject 5 : 81

Credits for subject 6 :

Marks for subject 6 : 86

Credits for subject 7 : 1

Marks for subject 7 : 98

Credits for subject 8 : 1

Marks for subject 8 : 89

SLPA : 9.00

O/P seen
GJ
S/10

```

import java.util.*;

public class Student_sgpa {
    static String USN;
    static int credits[] = new int[8];
    static int marks[] = new int[8];

    Student_sgpa() {
        System.out.println("Enter your details: ");
    }

    static void accept() {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the USN: ");
        USN = sc.nextLine();
        System.out.println("Enter the credits and marks: ");
        for (int i = 0; i < 8; i++) {
            System.out.print("Credits for subject " + (i + 1) + ": ");
            credits[i] = sc.nextInt();
            System.out.print("Marks for subject " + (i + 1) + ": ");
            marks[i] = sc.nextInt();
        }
    }

    static int sgpa() {
        int grade_points[] = new int[8];
        int sub_points[] = new int[8];
        int total_points = 0;
        int total_credits = 0;
        for (int i = 0; i < 8; i++) {
            if (marks[i] >= 90) {
                grade_points[i] = 10;
            } else if (marks[i] >= 80) {
                grade_points[i] = 9;
            } else if (marks[i] >= 70) {
                grade_points[i] = 8;
            } else if (marks[i] >= 60) {
                grade_points[i] = 7;
            } else if (marks[i] >= 50) {
                grade_points[i] = 6;
            }
            sub_points[i] = credits[i] * grade_points[i];
            total_points += sub_points[i];
            total_credits += credits[i];
        }
        float sgpa = (float) total_points / total_credits;
        return (int) sgpa;
    }
}

```

```

        grade_points[i] = 6;
    } else if (marks[i] >= 40) {
        grade_points[i] = 5;
    } else {
        grade_points[i] = 0;
    }

    sub_points[i] = credits[i] * grade_points[i];
    total_points += sub_points[i];
    total_credits += credits[i];
}

if (total_credits == 0) return 0;
return total_points / total_credits;
}

static void display() {

    System.out.println("\n\nUSN: " + USN);

    for (int i = 0; i < 8; i++) {
        System.out.println("Credits for subject " + (i + 1) + ": " + credits[i]);
        System.out.println("Marks for subject " + (i + 1) + ": " + marks[i]);
    }

    System.out.println("\nSGPA: " + sgpa());
}

public static void main(String args[]) {
    accept();
    display();
}
}

```

Output:

```
due
: if
ade D:\1BM23CS075>javac Student.java
if
ade D:\1BM23CS075>java Student.java
Enter the USN:
{ 1BM23CS068
ade Enter the credits and marks:
Credits for subject 1: 4
int Marks for subject 1: 97
poi Credits for subject 2: 3
cre Marks for subject 2: 83
.Cre Credits for subject 3: 3
Marks for subject 3: 94
.cre Credits for subject 4: 3
.al_ Marks for subject 4: 91
Credits for subject 5: 3
Marks for subject 5: 81
.spl Credits for subject 6: 1
Marks for subject 6: 86
Credits for subject 7: 1
Marks for subject 7: 98
.= Credits for subject 8: 1
Marks for subject 8: 89
.l.out
l.out
```

```
ide
if Credits for subject 1: 4
ide Marks for subject 1: 97
ide
ide Credits for subject 2: 3
ide Marks for subject 2: 83
ide
ide Credits for subject 3: 3
ide Marks for subject 3: 94
ide
ide Credits for subject 4: 3
ide Marks for subject 4: 91
int
ide Credits for subject 5: 3
ide Marks for subject 5: 81
ide
ide Credits for subject 6: 1
ide Marks for subject 6: 86
ide
ide Credits for subject 7: 1
ide Marks for subject 7: 98
ide
ide Credits for subject 8: 1
ide Marks for subject 8: 89
.= SGPA: 9
ou
ou D:\1BM23CS075>
```

Enter command number:

Program 3

Book Details

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()` that could display the complete details of the book. Develop a Java program to create n book objects.

Algorithm:

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



```
import java.util.*;  
  
public class Book {  
    private String name;  
    private String author;  
    private double price;  
    private int num-pages;  
  
    public Book() {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.num-pages = num-pages;  
    }
```

```
    public void setname(name) {  
        this.name = name;
```

```
    }  
    public void display getname() {  
        return name;
```

3

```
public void setauthor(author) {  
    this.author = author;  
}
```

```
public String getauthor() {  
    return author;  
}
```

```
public void setprice(price) {  
    this.price = price;  
}
```

```
public double getprice() {  
    return price;  
}
```

```
public void setnumpages(numPages) {  
    this.numPages = numPages;  
}
```

```
public int getnumpages() {  
    return numPages;  
}
```

@ override

```
public String toString() {  
    return "Book:" + getbook + "Author:" +  
        getauthor + "Price:" + getprice + "Pages:  
        " + getnumpages;  
}
```

P1b

```

public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the number books : ");
    int n = sc.nextInt();
    System.out.println("Enter the details");
    for(int i=0; i<n; i++) {
        Book book[i] = new Book();
        System.out.println("BookName : ");
        setName(sc.nextLine());
        System.out.println("Author : ");
        setAuthor(sc.nextLine());
        System.out.println("price : ");
        setPrice(sc.nextDouble());
        System.out.println("Num-pages : ");
        setNumPages(sc.nextInt());
        Book book[i] = new Book();
    }
}

```

```

public void display() {
    for(int i=0; i<n; i++) {
        System.out.println("Book" + i);
        System.out.println(book[i].toString());
    }
}

```

g
g
g
g

seen execute

Output:

Enter the number of books : 2

Enter details for book 1:

Name : Headfirst Java

Author : Kjm

Price : 599

Num pages : 658

Enter details for book 2:

Name : Atomic Habit

Author : James

Price : 199

Num pages : 254

Book 1

Name : Headfirst Java

Author : Kjm

Price : 599

Num pages : 658

Book 2

Name : Atomic habit

Author : James

Price : 199

Num pages : 254

OP Seen

gk
15/10/24

Code:

```
import java.util.Scanner;

public class Book {
    private String name;
    private String author;
    private double price;
    private int numPages;

    public Book() {}

    public Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getAuthor() {
        return author;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public double getPrice() {
        return price;
    }
}
```

```

public void setNumPages(int numPages) {
    this.numPages = numPages;
}

public int getNumPages() {
    return numPages;
}

@Override
public String toString() {
    return "Name: " + getName() + "\nAuthor: " + getAuthor() + "\nPrice: " + getPrice() + "\nNum
Pages: " + getNumPages();
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the number of books: ");
    int n = sc.nextInt();
    sc.nextLine(); // Consume the newline character

    Book[] books = new Book[n];

    for (int i = 0; i < n; i++) {
        System.out.println("\nEnter details for book " + (i + 1) + ":");
        System.out.print("Name: ");
        String name = sc.nextLine();

        System.out.print("Author: ");
        String author = sc.nextLine();

        System.out.print("Price: ");
        double price = sc.nextDouble();

        System.out.print("Num Pages: ");
        int numPages = sc.nextInt();
        sc.nextLine();

        books[i] = new Book(name, author, price, numPages);
    }

    display(books);
    sc.close();
}

```

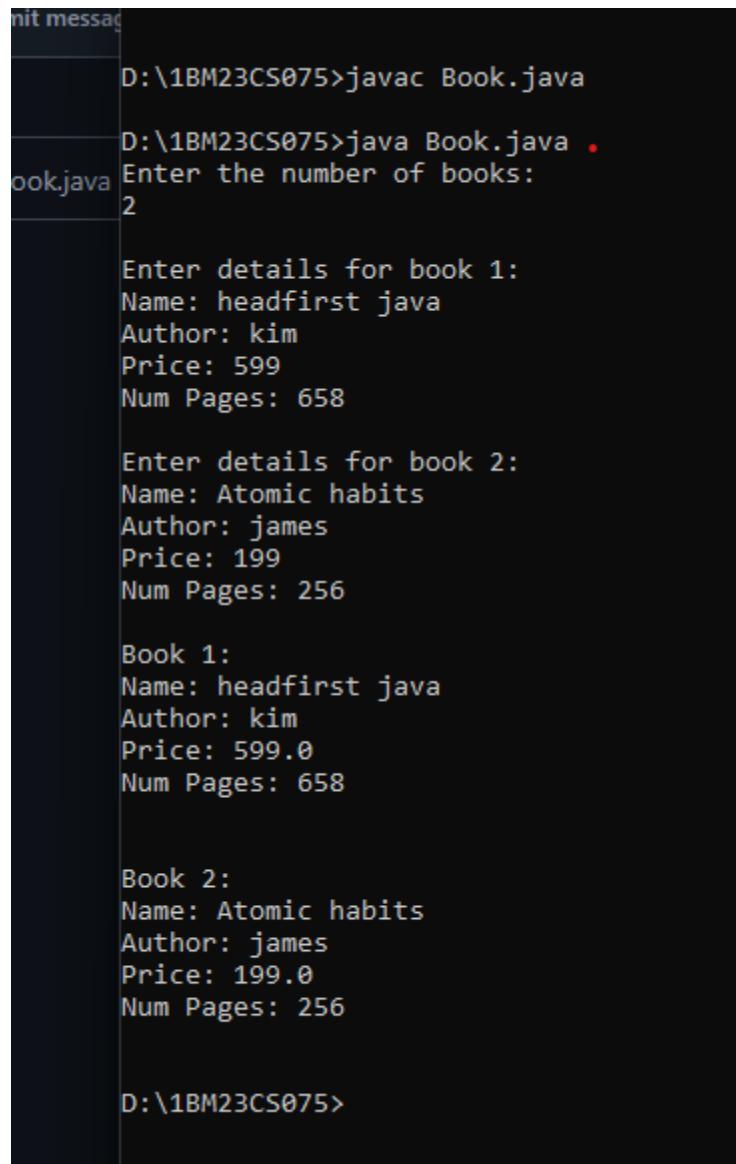
```

    }

static void display(Book[] books) {
    for (int i = 0; i < books.length; i++) {
        System.out.println("\nBook " + (i + 1) + ":");
        System.out.println(books[i].toString());
        System.out.println();
    }
}
}

```

Output:



```

init message
D:\1BM23CS075>javac Book.java
D:\1BM23CS075>java Book.java .
ook.java
Enter the number of books:
2

Enter details for book 1:
Name: headfirst java
Author: kim
Price: 599
Num Pages: 658

Enter details for book 2:
Name: Atomic habits
Author: james
Price: 199
Num Pages: 256

Book 1:
Name: headfirst java
Author: kim
Price: 599.0
Num Pages: 658

Book 2:
Name: Atomic habits
Author: james
Price: 199.0
Num Pages: 256

D:\1BM23CS075>

```

Program 4

Abstract Class Shape

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:

Develop a java program to create class named Shape that contains two integers & 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 contains only the method printArea() that prints the area of the given shape.

⇒

```
abstract class Shape {  
    private int dim1, dim2;  
  
    Shape (int dim1, int dim2) {  
        this.dim1 = dim1;  
        this.dim2 = dim2;  
    }  
  
    public void abstract void printArea();  
}
```

```
class Rectangle extends Shape {  
    int length is;  
    int breadth is;  
    Rectangle (int length, int breadth) {  
        Super (length, breadth);  
    }  
  
    public void printArea (int length, int breadth)  
    {  
        int area = length * breadth;  
        System.out.println ("Rectangle area - " + area);  
    }  
}
```

```

class Triangle extends shape {
    int height = 0;
    int breadth = 0;
    Triangle (int height, int breadth) {
        super (height, breadth);
    }
    public void parentArea (int height, int breadth) {
        double area = 0.5 * height * breadth;
        System.out.println ("Triangle area : " + area);
    }
}

```

```

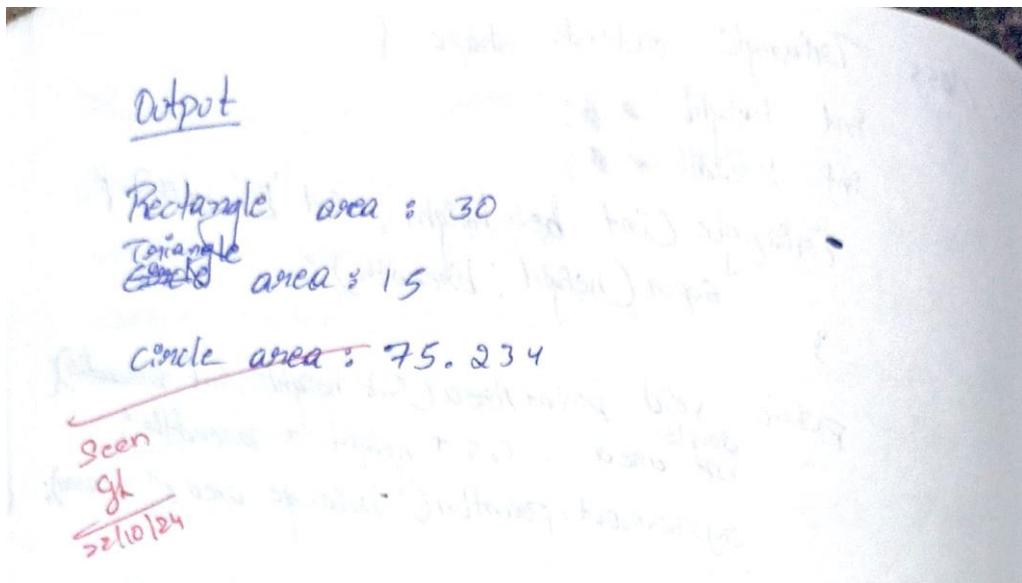
class Circle extends Circle {
    int radius = 0;
    Circle (int radius) {
        super (radius);
    }
    public void parentArea (int radius) {
        double area = Math.PI * radius * radius;
        System.out.println ("Circle area : " + area);
    }
}

```

```

class Test {
    public static void main (String args[]) {
        Rectangle r1 = new Rectangle ();
        Triangle t = new Triangle ();
        Circle c = new Circle ();
        r1.parentArea (6, 5);
        t.parentArea (6, 5);
        c.parentArea (5);
    }
}

```



Code:

```
import java.util.Scanner;

abstract class Shape {
    int a, b;

    abstract double printArea();

    void Value() {
        Scanner s = new Scanner(System.in);
        a = s.nextInt();
        b = s.nextInt();
    }

    void Valuerad() {
        Scanner s = new Scanner(System.in);
        a = s.nextInt();
    }
}

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

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

class Circle extends Shape {
```

```

        double printArea() {
            return 3.14 * a * a;
        }
    }

class shapeArea {
    public static void main(String args[]) {
        Rectangle r = new Rectangle();
        System.out.println("Enter the values of length and breadth: ");
        r.Value();
        Triangle t = new Triangle();
        System.out.println("Enter the values of base and height: ");
        t.Value();
        Circle c = new Circle();
        System.out.println("Enter the value of radius: ");
        c.Valuerad();
        System.out.println("The area of rectangle is: " + r.printArea());
        System.out.println("The area of Triangle is: " + t.printArea());
        System.out.println("The area of Circle is: " + c.printArea());
    }
}

```

Output :

}

```

Enter the values of length and breadth:
5
6
Enter the values of base and height:
6
7
Enter the value of radius:
9
The area of rectangle is: 30.0
The area of Triangle is: 21.0
The area of Circle is: 254.34

```

Program 5

Bank Details

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

Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

Algorithm:

```
public class Bank {
```

```
    boolean withdraw;
```

```
    boolean interest;
```

```
    boolean cheque-book;
```

```
Bank (boolean withdraw, boolean interest,  
boolean cheque-book) {
```

```
    this.withdraw = withdraw;
```

```
    this.interest = interest;
```

```
    this.cheque-book = cheque-book;
```

```
}
```

1

```
public void saving-acc (boolean withdraw,  
boolean interest, boolean cheque-book) {
```

```
    withdraw = true;
```

```
    interest = true;
```

```
    cheque-book = false;
```

```
}
```

public void current-acc (boolean withdraw,
boolean interest, boolean cheque-book) {

~~withdraw = true;~~

~~interest = false;~~

~~cheque-book = true;~~

3

4

```

public class Account {
    char name[50];
    int acc-no;
    char acc-type[50];
}

Account (char name[50], int acc-no, char
acc-type[50]) {
    this.name = name;
    this.acc-no = acc-no;
    this.acc-type = acc-type;
}

```

```

class Sav-act extends Account {
    int balance;
    int amt;
    Super. acc-type = bank.saving-acc;

    Sav-act (char name[50], int acc-no, char
    acc-type[50], balance) {
        Super.name = name;
        Super.acc-no = acc-no;
        Super.acc-type = acc-type;
        this.balance = balance;
    }
}

```

```

    deposit(balance);
    balance(balance);
    withdraw(balance);
    interest(balance);
}

```

~~class~~ ~~Bank~~ over-acct extends Account {
 int balance;
 int amt;

~~Scope~~

deposit (balance);

balance (balance);

withdraw (balance);

y

```
public void deposit(int amt) {  
    System.out.println("Enter the amount: ");  
    int amt = sc.nextInt();  
    int balance += amt;  
    System.out.println("Balance: " + balance);  
}
```

```
public int balance() {  
    System.out.println("Balance: " + balance);  
}
```

```
public int withdraw(int amt, int balance) {  
    System.out.println("Enter the amount: ");  
    int amt = sc.nextInt();  
    int balance -= amt;  
    return balance;  
}
```

```
public int interest(int balance) {  
    int days = 30;  
    balance = (balance * (days / 30) * 0.1);  
    return balance;  
}
```

```
public int account() {  
    public int act-type;  
    public int account() {  
        S  
    }
```

```

public static void main(String args[]) {
    System.out.println("Account type : ");
    Scanner sc = new Scanner(System.in);
    char acc_type = sc.nextString();
    while (true) {
        System.out.println("Options");
        System.out.println("1. Create account");
        System.out.println("2. Deposit");
        System.out.println("3. Withdraw");
        System.out.println("4. Balance");
        System.out.println("5. Exit");
        int input = sc.nextInt();
        if (acc_type == sav_acct) {
            switch (input) {
                case 1: sav_act.deposit(balance);
                break;
                case 2: sav_act.withdraw(balance);
                break;
                case 3: sav_act.balance(balance);
                break;
                case 4: exist(a);
                break;
                default: System.out.println("Invalid Input");
            }
        }
    }
}

```

```

public static void main(String args[]) {
    System.out.println("Account type : ");
    Scanner sc = new Scanner(System.in);
    char acc_type = sc.nextString();
    while (true) {
        System.out.println("Options");
        System.out.println("1. Create account");
        System.out.println("2. Deposit");
        System.out.println("3. Withdraw");
        System.out.println("4. Balance");
        System.out.println("5. Exit");
        int input = sc.nextInt();
        if (acc_type == sav_acct) {
            switch (input) {
                case 1: sav_act.deposit(balance);
                break;
                case 2: sav_act.withdraw(balance);
                break;
                case 3: sav_act.balance(balance);
                break;
                case 4: exist(a);
                break;
                default: System.out.println("Invalid Input");
            }
        }
    }
}

```

```

        closeif(acc-type = curr-acct) {
            switch (input) {
                case 1: curr-acct.deposite(balance);
                break;
                case 2: curr-acct.withdraw(balance);
                break;
                case 3: curr-acct.balance(balance);
                break;
                case 4: exist();
                break;
                default: System.out.println("Invalid input");
            }
        }
    }
}

```

3
3
~~seen~~
~~12~~
16/12/24

Code:

```

import java.util.Scanner;

class Account {
    String name;
    int accNo;
    String accType;
    double balance;

    Account(String name, int accNo, String accType, double balance) {
        this.name = name;
        this.accNo = accNo;
        this.accType = accType;
        this.balance = balance;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Amount deposited: " + amount);
        displayBalance();
    }
}

```

```

}

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

class SavAcct extends Account {
    final double interestRate = 0.04; // Assuming 4% annual interest

    SavAcct(String name, int accNo, double balance) {
        super(name, accNo, "Savings", balance);
    }

    public void computeInterest(int months) {
        double interest = balance * (interestRate / 12) * months;
        balance += interest;
        System.out.println("Interest added for " + months + " months: " + interest);
        displayBalance();
    }

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

class CurAcct extends Account {
    final double minBalance = 500.0;
    final double penalty = 50.0;

    CurAcct(String name, int accNo, double balance) {
        super(name, accNo, "Current", balance);
    }

    public void checkMinimumBalance() {
        if (balance < minBalance) {

```

```

        balance -= penalty;
        System.out.println("Penalty imposed due to minimum balance violation: " + penalty);
    }
    displayBalance();
}

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

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

        System.out.print("Enter name: ");
        String name = sc.nextLine();

        System.out.print("Enter account number: ");
        int accNo = sc.nextInt();

        System.out.print("Select account type (1 for Savings, 2 for Current): ");
        int accTypeChoice = sc.nextInt();

        System.out.print("Enter initial balance: ");
        double balance = sc.nextDouble();

        Account account;
        if (accTypeChoice == 1) {
            account = new SavAcct(name, accNo, balance);
        } else {
            account = new CurAcct(name, accNo, balance);
        }

        int choice;
    }
}

```

```

do {
    System.out.println("\nOptions:");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");
    if (account instanceof SavAcct) {
        System.out.println("4. Compute Interest");
    }
    System.out.println("5. Exit");
    System.out.print("Enter your choice: ");
    choice = sc.nextInt();

    switch (choice) {
        case 1:
            System.out.print("Enter deposit amount: ");
            double depositAmount = sc.nextDouble();
            account.deposit(depositAmount);
            break;
        case 2:
            System.out.print("Enter withdrawal amount: ");
            double withdrawAmount = sc.nextDouble();
            if (account instanceof SavAcct) {
                ((SavAcct) account).withdraw(withdrawAmount);
            } else {
                ((CurAcct) account).withdraw(withdrawAmount);
            }
            break;
        case 3:
            account.displayBalance();
            break;
        case 4:
            if (account instanceof SavAcct) {
                System.out.print("Enter number of months for interest calculation: ");
                int months = sc.nextInt();
                ((SavAcct) account).computeInterest(months);
            } else {
                System.out.println("Invalid option for current account.");
            }
            break;
        case 5:
            System.out.println("Exiting...");
            break;
    }
}

```

```
        default:  
            System.out.println("Invalid choice. Try again.");  
        }  
    } while (choice != 5);  
  
    sc.close();  
}  
}  
    input.close();  
}  
}
```

Output:

```
PS D:\1BM23CS075> java BankApp
Enter name: Chethan N
Enter account number: 654
Select account type (1 for Savings, 2 for Current): 1
Enter initial balance: 1000

Options:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
5. Exit
Enter your choice: 2
Enter withdrawal amount: 500
Amount withdrawn: 500.0
Current balance: 500.0

Options:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
5. Exit
Enter your choice: 3
Current balance: 500.0

Options:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
5. Exit
Enter your choice: 5
Exiting...
PS D:\1BM23CS075> 
```

Program 6

Packages

Create a package CIE which has two classes- Student 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 Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

Algorithm:

Lab 6

Create a package CIE which has two classes Student and Internal, the class Personal has members like usn, name, sem. The class Internal has an array that stores the internal marks scored in 5 courses of the current semester of the student.

Create another package SEE which has the class external

=> Package CIE :

```
public class Student {  
    public int usn;  
    public String name;  
    public int sem;
```

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

y

```

package CIE;
import java.util.*;

public class Internals {
    int inMarks[] = new int[5];

    public void inputInternalMarks() {
        System.out.println("Enter the internal marks");
        for(int i=0; i<inMarks.length; i++) {
            System.out.print("Sub " + (i+1) + ": ");
            Scanner sc = new Scanner(System.in);
            inMarks[i] = sc.nextInt();
        }
    }
}

```

```

package SEE;
import CIE.*;
import SEE.*;

public class MainExam {
    public static void main(String args[]) {
        Student
    }
}

```

```

package SEE;
import java.util.*;

public class Externals {
    int exMarks[] = new int[5];

    public void inputExternalMarks() {
        System.out.println("Enter the external
marks");
    }
}

```

```

for(int i=0; i<exMarks.length; i++) {
    System.out.print ("Sob " + (i+1) + ": ");
    scanner sc = new Scanner(System.in);
    exMarks[i] = sc.nextInt();
}

import CIE.*;
import SEE.*;

public class MainExam {
    public static void main(String args[]) {
        Student s1 = new Student(1, "Raja", 3);
        Internals i1 = new Internals();
        Externals e1 = new Externals();

        i1.InputInternalMarks();
        e1.InputExternalMarks();
    }
}

```

OP
 Enter the internal marks:
 sub1 : 93
 sub2 : 85
 sub3 : 86
 sub4 : 94
 sub5 : 90

Enter the external marks:
 sub1 : 90
 sub2 : 94
 sub3 : 90
~~sub4 : 90~~
~~sub5 : 86~~

GL
 16/12/24

Code:

```

package CIE;

public class Student {
  public int usn;
  public String name;
  public int sem;

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

package CIE;
import java.util.*;

public class Internals {
  ...
}
  
```

```

int inMarks[] = new int[5];

public void inputInternalMarks() {
    System.out.println("\nEnter the internal marks");
    for(int i = 0; i<inMarks.length; i++) {
        System.out.print("Sub " + (i+1) + " : ");

        Scanner sc = new Scanner(System.in);
        inMarks[i] = sc.nextInt();
    }
}

package SEE;
import java.util.*;

public class External {
    int exMarks[] = new int[5];

    public void inputExternalMarks() {
        System.out.println("\nEnter the external marks");
        for(int i = 0; i<exMarks.length; i++) {
            System.out.print("Sub " + (i+1) + " : ");

            Scanner sc = new Scanner(System.in);
            exMarks[i] = sc.nextInt();
        }
    }
}

import CIE.*;
import SEE.*;

public class MainExam {
    public static void main(String args[]) {
        Student s1 = new Student(1, "Raja", 3);
        Internals i1 = new Internals();
        External e1 = new External();

        i1.inputInternalMarks();
        e1.inputExternalMarks();
    }
}

```

```
    }  
}
```

Output:

```
D:\IBM23CS075>javac MainExam.java  
D:\IBM23CS075>java MainExam.java  
Enter the internal marks  
Sub 1 : 93  
Sub 2 : 85  
Sub 3 : 86  
Sub 4 : 94  
Sub 5 : 90  
Enter the external marks  
Sub 1 : 90  
Sub 2 : 94  
Sub 3 : 90  
Sub 4 : 90  
Sub 5 : 86
```

```
Final Marks of Students:  
Student 1: CHETHAN K S (USN: IBM23CS074)  
Course Internal SEE Final Marks  
Course 1: 40 98 138  
Course 2: 36 86 122  
Course 3: 39 91 130  
Course 4: 35 87 122  
Course 5: 29 78 107
```

Program 7

Interfaces

Algorithm:

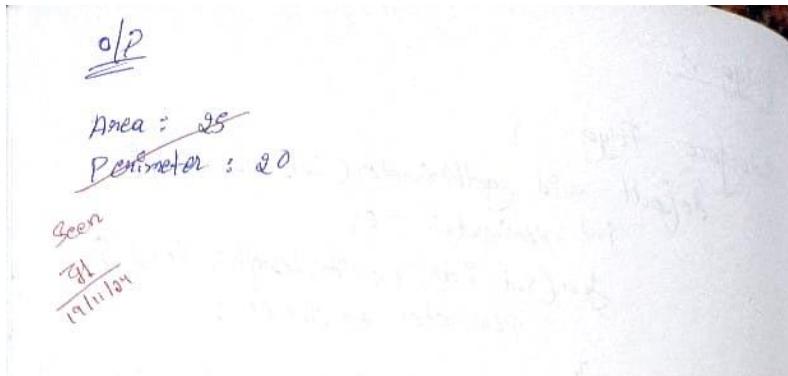
g) Polygon

```
interface Polygon {  
    default void getPerimeter(int sides[]) {  
        int perimeter = 0;  
        for (int i=0; i < sides.length; i++) {  
            perimeter += sides[i];  
        }  
        System.out.println("Perimeter: " +  
                           perimeter);  
    }  
}
```

```
void getArea(int a);
```

```
class Square implements Polygon {  
    void getArea(int sideLength) {  
        System.out.println("Area: " + sideLength *  
                           sideLength);  
    }  
}
```

```
public class PolygonMain {  
    public static void main(String args[]) {  
        Square s1 = new Square();  
        s1.getArea(5);  
        int sides[] = {5, 6, 5, 6};  
        s1.getPerimeter(sides);  
    }  
}
```



Code:

```

interface Polygon {
    default void getPerimeter(int sides[]) {
        int perimeter = 0;
        for(int i=0; i<sides.length; i++) {
            perimeter += sides[i];
        }
        System.out.println("Perimeter: " + perimeter);
    }

    void getArea(int sideLength1, int sideLength2);
}

class Square implements Polygon {
    public void getArea(int sideLength1, int sideLength2) {
        System.out.println("Square Area: " + sideLength1*sideLength2);
    }
}

class Rectangle implements Polygon {
    public void getArea(int sideLength1, int sideLength2) {
        System.out.println("Rectangle Area: " + sideLength1*sideLength2);
    }
}

public class PolygonMain {
    public static void main(String args[]) {
        Square s1 = new Square();
        s1.getArea(5,5);

        int sq_sides[] = {5,5,5,5};
        s1.getPerimeter(sq_sides);
    }
}

```

```

        Rectangle r1 = new Rectangle();
        r1.getArea(5,6);

        int rect_sides[] = {5,6,5,6};
        r1.getPerimeter(rect_sides);
    }
}

```

Output

```

D:\1BM23CS075\Lab_6>javac PolygonMain.java

D:\1BM23CS075\Lab_6>java PolygonMain
Square Area: 25
Perimeter: 20
Rectangle Area: 30
Perimeter: 22
viaD:\1BM23CS075\Lab_6>_

```

Program 8

Exception Handling

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

Algorithm:

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

```

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

class Father {
    int father-age;
    Father (int father-age) {
        father (int father-age) {
            this.father-age = father-age;
        }
    }
    Father (int father-age) {
        if if (father-age < 0) {
            wrongAge ("Father age should not less
            than 0");
        } else {
            this.father-age = father-age;
            System.out.println ("Father's age is "
                + father-age);
        }
    }
}

```

```
class son extends Father {  
    int son-age;
```

```
son (int father-age, int son-age) {
```

```
    try {
```

```
        super(father-age);
```

```
        if (father-age <= son-age) {
```

```
            wrongAge ("Son's age should not  
            more than father's age");
```

```
}
```

```
        this.son-age = son-age;
```

```
        System.out.println ("Son's age : " +  
                           son-age);
```

```
} catch (wrongAge e) {
```

```
    System.out.println (e.wrongAge);
```

```
} catch (Exception e) {
```

```
    System.out.println (e.Exception);
```

```
}
```

```
3
```

```

public class FatherSonAge {
    public static void main(String args[]) {
        try {
            Father f1 = new Father(45);
            Son s1 = new Son(f1);
        } catch {
            System.out.println("e.getMessage()");
        }
    }
}

```

Output:

Father's age is 45

Father's age is 45

Wrong Age Expectation: Son's age cannot be greater than father's age

seen

YH
36/124

Code:

```

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

```

```

class Father {
    int age;

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

class Son extends Father {
    int sonAge;

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

public class WrongAgeMain {
    public static void main(String[] args) {
        try {
            Father father = new Father(45);
            Son son = new Son(45, 50);
        } catch (WrongAge e) {
            System.out.println("WrongAge Exception: " + e.getMessage());
        }
    }
}

```

Output :

```

D:\1BM23CS075\Lab_8>java Main
Father's age is 45
Father's age is 45
WrongAge Exception: Son's age cannot be greater than or equal to Father's age
.

D:\1BM23CS075\Lab_8>

```

Program 9

Threads

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:

1. write a program which creates two threads
on thread display

```
class NewThread extends Thread {  
    String message;  
    int interval;  
  
    NewThread (String message, int interval) {  
        this.message = message;  
        this.interval = interval;  
    }  
  
    public void run() {  
        try {  
            for (int i=0; i<3; i++) {  
                System.out.println (message);  
                Thread.sleep (interval);  
            }  
        } catch (InterruptedException e) {  
            System.out.println (e);  
        }  
    }  
}
```

```
public class ObjThread {
    public static void main(String args[]) {
        NewThread n1 = new NewThread("BMSCE",
                                      10000);
        NewThread n2 = new NewThread("CSE",
                                      9000);
        n1.start();
        n2.start();
    }
}
```

obj
BMSCE
CSE
CSE
CSE
~~CSE~~
BMSCE
BMSCE

Given
03/12/24

Code:

```
class NewThread extends Thread {
    String message;
    int interval;
```

```

NewThread(String message, int interval) {
    this.message = message;
    this.interval = interval;
}

public void run() {
    try {
        for (int i = 0; i < 3; i++) {
            System.out.println(message);
            Thread.sleep(interval);
        }
    } catch (InterruptedException e) {
        System.out.println("Interrupted");
    }
}

public class ClgThread {
    public static void main(String args[]) {
        NewThread n1 = new NewThread("BMSCE", 1000);
        NewThread n2 = new NewThread("CSE", 2000);

        n1.start();
        n2.start();
    }
}

```

Output:

```

D:\1BM23CS075\lab9>javac ClgThread.java

D:\1BM23CS075\lab9>java ClgThread
BMSCE
CSE
BMSCE
BMSCE
CSE
CSE

D:\1BM23CS075\lab9>

```

Program 10

GUI – Java Swing

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.

Lab

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 by Num2 is displayed in the result field when the Divide button is clicked. If Num1 or Num2 were not a integer the program would throw a NumberFormat Exception. If Num2 were zero, the program would throw an Arithmetic Exception. Display the exception.

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

public class DivisionApp {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Integer Division");
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);

        JLabel label1 = new JLabel("Num1:");
        label1.setBounds(20, 20, 50, 20);
        frame.add(label1);

        JTextField num1Field = new JTextField();
        num1Field.setBounds(80, 20, 100, 20);
        frame.add(num1Field);
    }
}
```

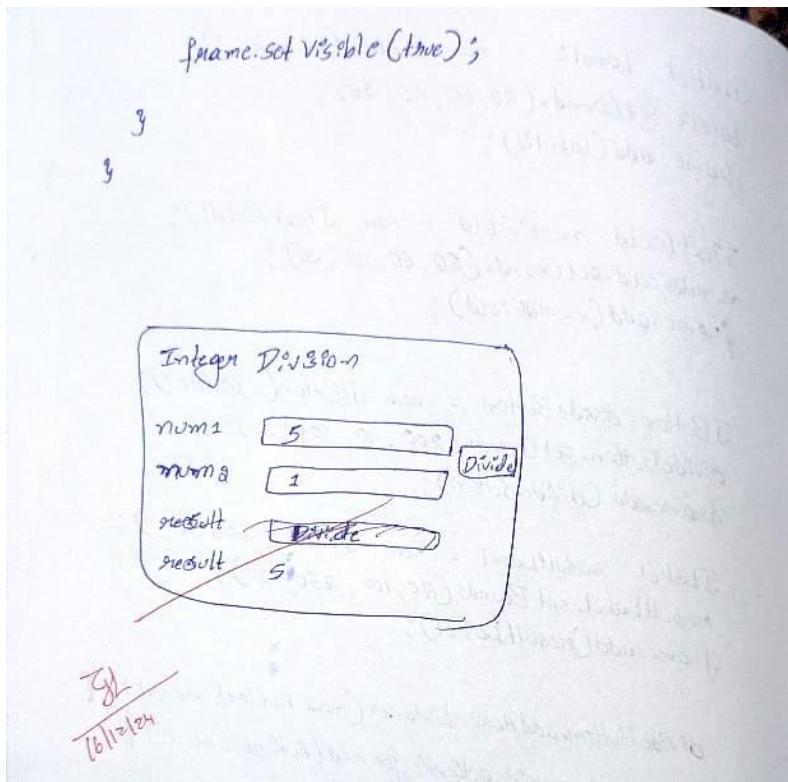
```
JLabel label2 = new JLabel("Num2:");
label2.setBounds(20, 60, 50, 20);
frame.add(label2);
```

```
JTextField num2Field = new JTextField();
num2Field.setBounds(80, 60, 100, 20);
frame.add(num2Field);
```

```
JButton divideButton = new JButton("Divide");
divideButton.setBounds(200, 40, 100, 30);
frame.add(divideButton);
```

```
JLabel resultLabel = new JLabel("Result:");
resultLabel.setBounds(20, 100, 250, 20);
frame.add(resultLabel);
```

```
divideButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            int num1 = Integer.parseInt(num1Field.getText());
            int num2 = Integer.parseInt(num2Field.getText());
            int result = num1 / num2;
            resultLabel.setText("Result: " + result);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(frame,
                "Please enter valid integers!", "Input Error",
                JOptionPane.ERROR_MESSAGE);
        } catch (ArithmaticException e) {
            JOptionPane.showMessageDialog(frame,
                "Cannot divide by zero!", "MathError",
                JOptionPane.ERROR_MESSAGE);
        }
    }
});
```



Code:

```

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class DivisionApp {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Integer Division");
        frame.setSize(400, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);

        JLabel label1 = new JLabel("Num1:");
        label1.setBounds(20, 20, 50, 20);
        frame.add(label1);
    }
}

```

```

JTextField num1Field = new JTextField();
num1Field.setBounds(80, 20, 100, 20);
frame.add(num1Field);

JLabel label2 = new JLabel("Num2:");
label2.setBounds(20, 60, 50, 20);
frame.add(label2);

JTextField num2Field = new JTextField();
num2Field.setBounds(80, 60, 100, 20);
frame.add(num2Field);

JButton divideButton = new JButton("Divide");
divideButton.setBounds(200, 40, 100, 30);
frame.add(divideButton);

JLabel resultLabel = new JLabel("Result:");
resultLabel.setBounds(20, 100, 250, 20);
frame.add(resultLabel);

divideButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            int num1 = Integer.parseInt(num1Field.getText());
            int num2 = Integer.parseInt(num2Field.getText());

            int result = num1 / num2;

            resultLabel.setText("Result: " + result);
        } catch (NumberFormatException nfe) {
            JOptionPane.showMessageDialog(frame, "Please enter valid integers!", "Input Error",
                JOptionPane.ERROR_MESSAGE);
        } catch (ArithmaticException ae) {
            JOptionPane.showMessageDialog(frame, "Cannot divide by zero!", "Math Error",
                JOptionPane.ERROR_MESSAGE);
        }
    }
});

frame.setVisible(true);
}
}

```

Output:

