

**VISVESVARAYATECHNOLOGICAL  
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
on  
**ObjectOrientedJavaProgramming**  
**(23CS3PCOOJ)**

*Submitted by*

**ABDUL AHAD (1BM23AI002)**

*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” carried out by **ABDUL AHAD(1BM23AI002)** 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 Artificial Intelligence (23CS5PCAIN) work prescribed for the said degree.

Ms AMBUJA K Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
--	--

## Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	
1	12/11/2024	QUADRATIC EQUATION	
2	19/11/2024	SGPA CALCULATOR	
3	26/11/2024	BOOKS	
4	26/11/2024	AREA OF SHAPE	
5	3/12/2024	BANK	
6	3/12/2024	PACKAGE CIE SEE	
7	10/12/2024	FATHER SON AGE EXCEPTION	
8	15/12/2024	MULTI THREADING	
9	15/12/2024	INTEGER DIVISION	
10	15/12/2024	OPEN ENDED –DEADLOCK IPC	

## Program 1

Code:

```
import java.util.Scanner;
public class lab1{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter coefficient a: ");
        double a = sc.nextDouble();
        System.out.print("Enter coefficient b: ");
        double b = sc.nextDouble();
        System.out.print("Enter coefficient c: ");
        double c = sc.nextDouble();

        double discriminant = b * b - 4 * a * c;

        if (discriminant < 0) {
            System.out.println("The equation has no real solutions.");
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("The equation has one real solution: " + root);
        } else {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("The equation has two real solutions:");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        }
    }
}
```

## Quadratic Equation

- 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 is no real solutions.

```
import java.util.Scanner;  
public class Main {  
    public static void Main (String [] args) {  
        Scanner scan = new Scanner (System.in);  
        System.out.println ("enter the first no.");  
        double a = scan.nextDouble ();  
        System.out.println ("enter the second no.");  
        double b = scan.nextDouble ();  
        System.out.println ("enter the third no.");  
        double c = scan.nextDouble ();  
        double discriminant = (b*b)-(4*a*c);  
        if (discriminant > 0) {  
            double root1 = (-b + Math.sqrt(discriminant)) /  
                (2*a);  
            double root2 = (-b - Math.sqrt(discriminant)) /  
                (2*a);  
            System.out.println ("The roots are equal = "+  
                root1 + ", " + root2);  
        } else if (discriminant == 0) {  
            double root1 = (-b) / (2*a);  
            double root2 = (-b) / (2*a);  
            System.out.println ("The roots are equal = "+  
                root1 + ", " + root2);  
        }  
    }  
}
```

```
else {
```

```
    System.out.println ("discriminant  
is negative");
```

```
    System.out.println ("no roots");
```

```
}
```

```
}
```

```
}
```

Output

enter the first no = 2

enter the second no = 5

enter the third no = 2

The roots are = -0.5, -2.0

enter the first no = 11

enter the second no = 1

enter the third no = 22

```
C:\Users\lenovo\Desktop\package\lab>javac lab1.java

C:\Users\lenovo\Desktop\package\lab>java lab1.java
Enter coefficient a: 1
Enter coefficient b: 2
Enter coefficient c: 1
The equation has one real solution: -1.0

C:\Users\lenovo\Desktop\package\lab>java lab1.java
Enter coefficient a: 4
Enter coefficient b: 2
Enter coefficient c: 6
The equation has no real solutions.

C:\Users\lenovo\Desktop\package\lab>java lab1.java
Enter coefficient a: 3
Enter coefficient b: 5
Enter coefficient c: 1
The equation has two real solutions:
Root 1: -0.2324081207560018
Root 2: -1.434258545910665
```

**PROGRAM 2:**

```
import java.io.*;
import java.util.Scanner;
public class student{
    Scanner sc=new Scanner(System.in);
    String usn;
    String name;
    int [] credit;
    int [] marks;
    public student(int numsubjects){
        credit=new int[numsubjects];
        marks=new int[numsubjects];
    }
    void accept(){
        System.out.println("enter usn=");
        usn=sc.nextLine();
        System.out.println("enter name=");
        name=sc.nextLine();
        for(int i=0;i<credit.length();i++){
            System.out.printf("enter credit of subject%d=",i);
            credit[i]=sc.nextLine();
        }
        for(int j=0;j<marks.length();j++){
            System.out.printf("enter marks of subject%d=",i);
            marks[i]=sc.nextLine();
        }
    }
    void display(){
        System.out.println("usn="+usn);
        System.out.println("name="+name);
        for(int i=0;i<credit.length();i++){
            System.out.printf("credit of subject%d=%d",i,credit[i]);
        }
        for(int j=0;j<marks.length();j++){
            System.out.printf("marks of subject%d=%d",i,marks[j]);
        }
    }
    void calc(){
        int totalcredit=0;
        int wmarks;
        for(int i=0;i<credit.length();i++){
            int mmars=(marks[i]/10)+1;
            totalcredit=totalcredit+credit[i];
            wmarks=wmarks+mmarks*credit[i];
        }
    }
}
```

```
if(totalcredit==0){
    return 0;
}
System.out.println("gpa="+ (wmarks/totalcredit));
}
public static void main(String[] args){
    int numsubjects=sc.nextInt();
    student s1=new student(numsubjects);
    s1.accept();
    s1.display();
    s1.calc();
}
}
```

1/1/24

Date \_\_\_\_\_  
Page \_\_\_\_\_

## Lab Program - 2

Develop a java program to create a class student with members USN, name, an array credits and an array marks. Include method to accept and display details and a method to calculate CGPA of a student.

```
import java.util.Scanner;  
class Student {  
    String usn; // usn usn;  
    String name;  
    int[] credits; // int credits;  
    int[] marks; // int marks;  
    student (int numsubjects) {  
        credits = new int [numsubjects]; // Constructor  
        marks = new int [numsubjects];  
    }  
    void acceptdetails() {  
        Scanner sc = new Scanner (System.in);  
        System.out.print ("Enter USN: ");  
        usn = sc.nextLine();  
        System.out.print ("Enter name: ");  
        name = sc.nextLine();  
        for (int i=0; i < credits.length; i++) {  
            System.out.print ("Enter credits for subject " +  
                +(i+1) + ": ");  
            credits[i] = sc.nextInt();  
            System.out.print ("Enter marks for subject " +  
                +(i+1) + ": ");  
            marks[i] = sc.nextInt();  
        }  
    }  
}
```

```
void displayDetails () {  
    System.out.println("Student Details:");  
    System.out.println("USN: " + usn);  
    System.out.println("Name: " + name);  
    System.out.println("Subjects, credits, marks:");  
    for (int i = 0; i < credits.length; i++) {  
        System.out.println("Subject" + (i + 1) + ": Credit: " +  
            credits[i] + ", Marks = " + marks[i]);  
    }  
}
```

3

```
double calculateCGPA () {  
    double totalCredits = 0;  
    double weightedMarks = 0;  
    for (int i = 0; i < credits.length; i++) {  
        int adjustedMarks = (marks[i] / 10) + 1;  
        weightedMarks += adjustedMarks * credits[i];  
        totalCredits += credits[i];  
    }  
}
```

3

```
if (totalCredits == 0) {  
    return 0;  
}  
return weightedMarks / totalCredits;  
3
```

```
public class Main {
```

```
    public static void main (String [] args) {  
        Scanner sc = new Scanner (System.in);  
        System.out.print ("Enter no of subjects:");  
        int numSubjects = sc.nextInt();  
        Student student = new Student (numSubjects);  
    }
```

Date \_\_\_\_\_  
Page \_\_\_\_\_

```
student.acceptdetails();  
student.displaydetails();  
double sgpa = student.calculategpa();  
System.out.println("InSGrPA:" + sgpa);  
3
```

### Output

Enter the number of subjects: 5

Enter USN : IBM21 CS056

Enter name : Utk

Enter credits for subject 1 : 4

Enter marks for subject 1 : 93

Enter credits for subject 2 : 4

Enter marks for subject 2 : 82

Enter credits for subject 3 : 3

Enter marks for subject 3 : 91

Enter credits for subject 4 : 3

Enter marks for subject 4 : 72

Enter credits for subject 5 : 1

Enter marks for subject 5 : 81

SGrPA : 9.26

Utkarsh

```
Enter number of subjects:  
3  
Enter USN:  
1bm2cs056  
Enter name:  
abc  
Enter credit of subject 1: 4  
Enter credit of subject 2: 3  
Enter credit of subject 3: 4  
Enter marks of subject 1: 98  
Enter marks of subject 2: 87  
Enter marks of subject 3: 95  
USN: 1bm2cs056  
Name: abc  
Credit of subject 1: 4  
Credit of subject 2: 3  
Credit of subject 3: 4  
Marks of subject 1: 98  
Marks of subject 2: 87  
Marks of subject 3: 95  
GPA: 9.73
```

### PROGRAM 3

```
import java.util.Scanner;

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

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

    public String getName() {
        return name;
    }

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

    public String getAuthor() {
        return author;
    }

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

    public double getPrice() {
        return price;
    }

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

    public int getNumPages() {
        return numPages;
    }

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

    @Override
    public String toString() {
        return "Book Name: " + name + "\n" +
               "Author: " + author + "\n" +
               "Price: " + price + "\n" +
               "Number of Pages: " + numPages;
    }
}
```

```

        "Author: " + author + "\n" +
        "Price: $" + price + "\n" +
        "Number of Pages: " + numPages;
    }
}

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

        System.out.print("Enter the number of books: ");
        int n = sc.nextInt();
        sc.nextLine();

        Book[] books = new Book[n];

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

            System.out.print("Enter book name: ");
            String name = sc.nextLine();
            System.out.print("Enter author name: ");
            String author = sc.nextLine();
            System.out.print("Enter price: ");
            double price = sc.nextDouble();
            System.out.print("Enter number of pages: ");
            int numPages = sc.nextInt();
            sc.nextLine();

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

        System.out.println("\nDetails of all books:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":");

            System.out.println(books[i].toString());
        }
    }
}

```

26/11/2024

### Lab-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 working() method that could display the complete details of the book. Develop a java program to create n book objects.

```
import java.util.Scanner;  
public class Book {  
    private String name;  
    private String author;  
    private int price;  
    private int num_pages;  
    //Constructor  
    public Book (String name, String author,  
                int price, int numPages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages  
    }
```

```
    public String getName () {  
        return name;  
    }
```

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

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

Date \_\_\_\_\_  
Page \_\_\_\_\_

```
public void setAuthor (String Author) {  
    this.author = author;  
}  
  
public int getPrice () {  
    return price;  
}  
  
public void setPrice (int Price) {  
    this.price = price;  
}  
  
public int getNumPages () {  
    return numPages;  
}  
  
public void setNumPages () {  
    this.numPages = numPages;  
}  
  
public String toString () {  
    return "Name:" + name + "Author:" +  
        author + "Price:" + price +  
        "Num pages:" + NumPages;  
}  
  
public class BookStore {  
    Scanner sc = new Scanner (System.in);  
    System.out.println ("Enter the no of book:");  
    int n = sc.nextInt();  
    Book[] books = new Book[n];  
    for (int i = 0; i < n; i++) {  
        System.out.println ("Enter the name of Book:");  
        name = sc.nextLine();  
        System.out.println ("Enter the name of Author:");  
        author = sc.nextLine();  
    }  
}
```

```
System.out.println("Enter the price:");  
price = sc.nextInt();  
System.out.println("Enter the no of pages:");  
numPages = sc.nextInt();  
book[i] = newBook(name, author, price, numPages);  
}  
for (int i=0; i<n; i++) {  
    System.out.println(book[i].toString());  
}
```

Output:

```
Enter the number of Books: 1  
Enter the detail of Book 1:  
Enter the name of Book: Java  
Enter the name of author: abc  
Enter the price of book: 200  
Enter the number of pages: 500
```

Name of book: Java

Name of author: abc

Price of book: 200

Number of pages: 500

Ans 2011

**OUTPUT:**

```
Enter the number of books: 2
Enter details for Book 1:
Enter book name: The Great Gatsby
Enter author name: F. Scott Fitzgerald
Enter price: 10.99
Enter number of pages: 180
Enter details for Book 2:
Enter book name: To Kill a Mockingbird
Enter author name: Harper Lee
Enter price: 12.50
Enter number of pages: 281
```

```
Details of all books:
```

```
Book 1:
Book Name: The Great Gatsby
Author: F. Scott Fitzgerald
Price: $10.99
Number of Pages: 180
```

```
Book 2:
Book Name: To Kill a Mockingbird
Author: Harper Lee
Price: $12.5
Number of Pages: 281
```

```

PROGRAM 4
import java.util.Scanner;

abstract class Shape {
    int l,b,r;

    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int length, int breadth) {
        this.l = length;
        this.b = breadth;
    }

    void printArea() {
        System.out.println("Area of Rectangle: " + l* b);
    }
}

class Triangle extends Shape {
    Triangle(int base, int height) {
        this.l = base;
        this.b = height;
    }

    void printArea() {
        System.out.println("Area of Triangle: " + (0.5 * l * b));
    }
}

class Circle extends Shape {
    Circle(int radius) {
        this.r = radius;
    }

    void printArea() {
        System.out.println("Area of Circle: " + (Math.PI * l * l));
    }
}

public class area{
    public static void main(String[] args) {
        Rectangle r1=new Rectangle(10,20);
        r1.printArea();
        Triangle t1=new Triangle(10,20);
        t1.printArea();
        Circle c1=new Circle(20);
        c1.printArea();
    }
}

```

### Lab-program -4

① Develop a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes name, 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

```

public abstract class Shape {
    public abstract printArea();
    double l, b, r;
}

class Rectangle extends Shape {
    double l;
    double b;
    public Rectangle (double l, double b) {
        this.l = l;
        this.b = b;
    }
    public void printArea () {
        double area = l * b;
        System.out.println ("The area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    double l;
    double b;
    public Triangle (double l, double b) {
        this.l = l;
    }
}

```

This.b = b;

}

public void printArea () {

double area = 0.5 \* b \* l;

System.out.println ("The area of triangle :" +  
area);

}

3

class Circle extends Shape {

double r;

public Circle (double r) {

This.r = r;

}

public void printArea () {

double area = 3.142 \* (r \* r);

System.out.println ("Area of circle :" + area);

3

3

public class Area {

public static void main (String args[]) {

Shape rec = new Rectangle (10, 20);

rec.printArea ();

Shape tri = new Triangle (10, 20);

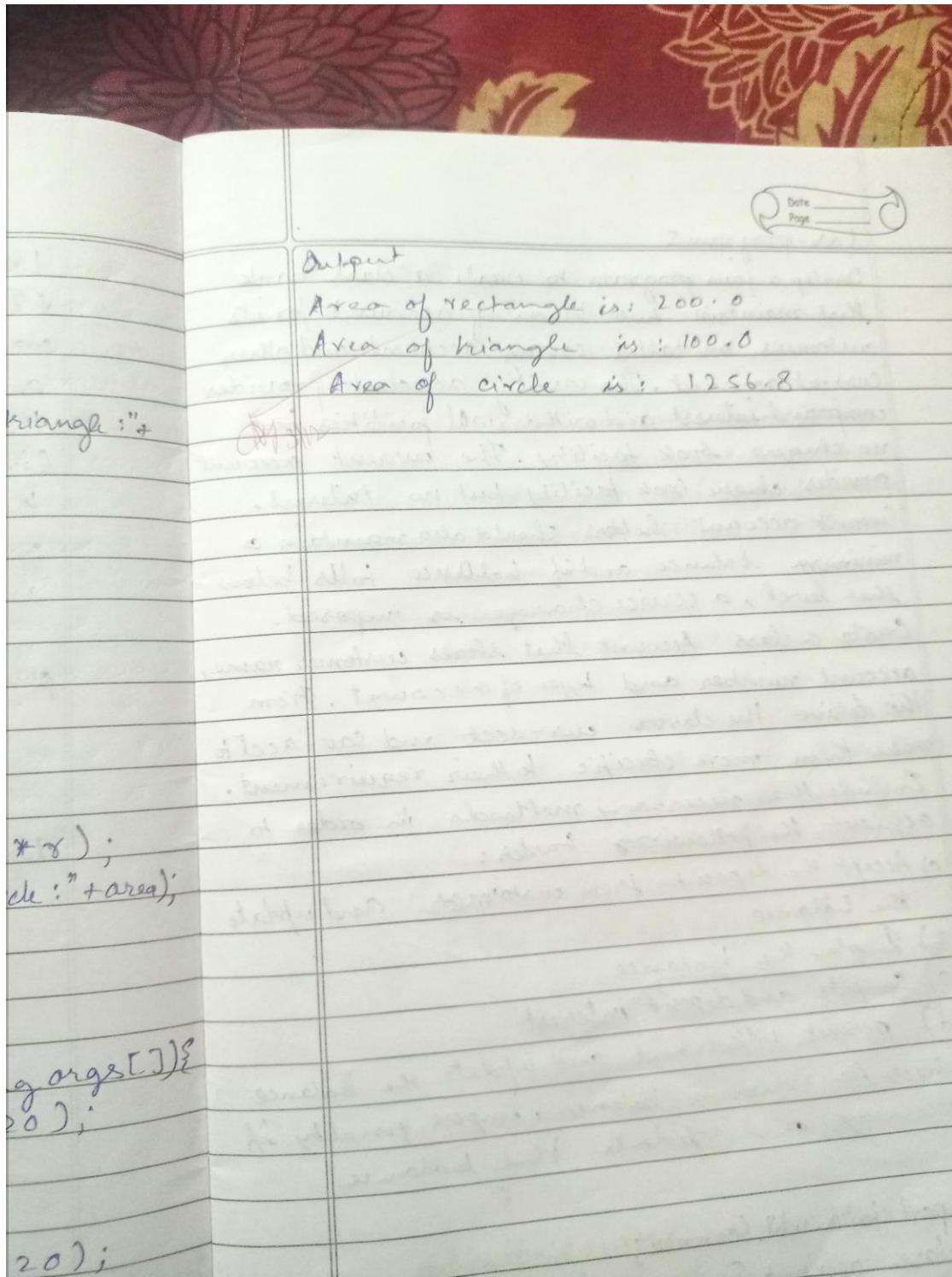
tri.printArea ();

Shape cir = new Circle (20);

cir.printArea ();

3

3



## OUTPUT:

```
C:\Users\lenovo\Desktop\CODING>cd "c:\Users\lenovo\Desktop\CODING\java\" && javac area.java && java area
Area of Rectangle: 200
Area of Triangle: 100.0
Area of Circle: 1256.6370614359173
```

## PROGRAM 5

```
import java.util.Scanner;

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

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

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful! Current balance = " + balance);
    }

    public void display() {
        System.out.println("Current balance = " + balance);
    }
}

class SavAccount extends Account {
    public SavAccount(String name, int accNo, double balance) {
        super(name, accNo, balance);
    }

    public void calculateInterest() {
        double rate = 0.05;
        double interest = rate * balance;
        balance += interest;
        System.out.println("Interest applied! Current balance = " + balance);
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful! Current balance = " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

class CurAccount extends Account {
    int minBalance = 1000;
    int penalty = 50;

    public CurAccount(String name, int accNo, double balance) {
```

```

        super(name, accNo, balance);
    }

    public void withdraw(double amount) {
        if (balance - amount >= minBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful! Current balance = " + balance);
        } else {
            balance -= penalty;
            System.out.println("Penalty applied! Current balance = " + balance);
        }
    }
}

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

        SavAccount sav = new SavAccount("Abe", 203, 1000);
        CurAccount cur = new CurAccount("Abc", 204, 2000);

        int choice;
        while (true) {
            System.out.println("Press 1 for Savings Account, 2 for Current Account, 0 to Exit:");
            choice = sc.nextInt();

            if (choice == 1) {
                System.out.println("1 - Display Balance\n2 - Withdraw\n3 - Deposit\n4 - Calculate Interest");
                int option = sc.nextInt();
                if (option == 1) {
                    sav.display();
                } else if (option == 2) {
                    System.out.print("Enter amount to withdraw: ");
                    double amount = sc.nextDouble();
                    sav.withdraw(amount);
                } else if (option == 3) {
                    System.out.print("Enter amount to deposit: ");
                    double amount = sc.nextDouble();
                    sav.deposit(amount);
                } else if (option == 4) {
                    sav.calculateInterest();
                } else {
                    System.out.println("Invalid option.");
                }
            } else if (choice == 2) {
                System.out.println("1 - Display Balance\n2 - Withdraw\n3 - Deposit");
                int option = sc.nextInt();
                if (option == 1) {
                    cur.display();
                } else if (option == 2) {

```

```

        System.out.print("Enter amount to withdraw: ");
        double amount = sc.nextDouble();
        cur.withdraw(amount);
    } else if (option == 3) {
        System.out.print("Enter amount to deposit: ");
        double amount = sc.nextDouble();
        cur.deposit(amount);
    } else {
        System.out.println("Invalid option.");
    }
} else if (choice == 0) {
    System.out.println("Exiting program.");
    break;
} else {
    System.out.println("Invalid choice.");
}
}
}
}
}

```

Lab-program  
 Develop a java program to create a class 'bank' that maintain two kinds of account for its customers, one called saving account and other current account. The saving 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 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 requirement.  
 Include the necessary methods in order to achieve the following tasks:  
 a) Accept the 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 minimum balance, impose penalty if necessary and update the balance

```

import java.util.Scanner;
class account {
    String name;
    int accno;
    double balance;
    public account(String name, int accno,
    double balance) {
  
```

this.name = name;  
this.accno = accno;  
this.balance = balance;

{

public void deposit(double amount) {  
    balance = balance + amount;  
    System.out.println("Deposit successful  
        current balance = " + balance);

{

public void display() {

    System.out.println("Current balance = " + balance);

{

class savings extends Account {

    public savings (String Name, int accno,  
        double balance) {  
        super (Name, accno, balance);

{

    public void cal\_inter() {

        double interest <sup>rate</sup> = 0.05;

        double interest = rate \* balance;

        balance = balance + interest;

        System.out.println("Interest applied  
            current balance = " + balance);

{

    public void withdraw(double amount) {

        if (amount <= balance) {

            balance = balance - amount;

            System.out.println("withdrawal

            successful of " + amount + " current balance = "  
            + balance);

        else {

            System.out.println("Insufficient balance");

{

}

```
public class cur acct extends account {  
    public curracct (String name, int accno,  
        double balance) {  
        super (name, accno, balance);
```

3

```
    int minbalance = 1000; int penalty = 50;
```

```
    public void withdraw (double amount) {
```

```
        if (amount >= minbalance) {
```

```
            balance -= amount;
```

```
            balance -= penalty;
```

```
            System.out.println ("Penalty applied");
```

```
            current balance = "+ balance);
```

3

```
    } else {
```

```
        balance -= amount;
```

```
        System.out.println ("withdraw successful");
```

```
        current balance = "+ balance);
```

3

```
class Bank {
```

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

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

```
        int n;
```

```
        int i = 0;
```

```
        Savings savacct sav = new savacct ("SavingsAbe",  
            203, 1000);
```

```
        Current cur = new curracct ("Abe", 203, 2000);
```

```
        while (i == 0) {
```

```
            System.out.println ("Press 1 for saving account  
                2 for current account);
```

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

Date \_\_\_\_\_  
Page \_\_\_\_\_

```
if(x == -1) {  
    sout("1->balance; 2->withdraw; 3->deposit;  
        4->interest");  
    x = sc.nextInt();  
    if(x == 1) {  
        sout("1->balance; 2->withdraw  
            sc.display());  
    }  
    else if(x == 2) {  
        sout("Extra amount : ");  
        int y = sc.nextInt();  
        sc.withdraw(y);  
    }  
    else if(x == 3) {  
        sout("Enter amount : ");  
        int y = sc.nextInt();  
        sc.deposit(y);  
    }  
    else if(x == 4) {  
        sc.calInterest();  
    }  
    else {  
        break;  
    }  
}
```

"Abc",  
00);  
03,2000);  
count in

```
else if(x == 2) {  
    sout("1->balance; 2->withdraw, 3->deposit");  
    x = sc.nextInt();  
    if(x == 1) {  
        sc.display();  
    }  
    else if(x == 2) {  
        sout("Enter the amount: ");  
    }
```

```
int y = sc.nextInt();  
cur. withdraw(y);
```

3

```
else if (y == 3) {  
    cout("Enter the amount: ");  
    int y = sc.nextInt();  
    cur. deposit(int y);
```

3

```
else {
```

```
    break;
```

3

### Output

Enter 1 → Saving acc ; 2 → Current acc

Enter 1 → balance ; 2 → deposit ; 3 →  
withdraw , 4 → Interest      ~~5 → exit~~  
current balance = 1000

2

Enter amount 200

Deposit successful current balance = 1200.

3

Enter amount

200

withdraw successful ; current balance = 1000

Interest applied current balance = 1050

QM 3/12

## OUTPUT:

```
C:\Users\lenovo\Desktop\CODING>java>cd "c:\Users\lenovo\Desktop\CODING\java\" && javac Bank.java && java Bank
Press 1 for Savings Account, 2 for Current Account, 0 to Exit:
1
1 - Display Balance
2 - Withdraw
3 - Deposit
4 - Calculate Interest
1
Current balance = 1000.0
Press 1 for Savings Account, 2 for Current Account, 0 to Exit:
1
1 - Display Balance
2 - Withdraw
3 - Deposit
4 - Calculate Interest
2
Enter amount to withdraw: 100
Withdrawal successful! Current balance = 900.0
Press 1 for Savings Account, 2 for Current Account, 0 to Exit:
4
Invalid choice.
Press 1 for Savings Account, 2 for Current Account, 0 to Exit:
1
1 - Display Balance
2 - Withdraw
3 - Deposit
4 - Calculate Interest
4
Interest applied! Current balance = 945.0
Press 1 for Savings Account, 2 for Current Account, 0 to Exit:
|
```

**PROGRAM 6:**

```
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 displayDetails() {
```

```
        System.out.println("USN: " + USN);  
        System.out.println("Name: " + Name);  
        System.out.println("Semester: " + Sem);  
    }
```

```
}
```

```
package CIE;
```

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

```
public class Internals extends Student {
```

```
    public int[] marks = new int[5];  
    Scanner input = new Scanner(System.in);
```

```
    public Internals(String USN, String Name, int Sem) {
```

```
        super(USN, Name, Sem);  
    }
```

```
    public void setMarks() {
```

```
        System.out.println("Enter the internal marks for 5 subjects:");  
        for (int i = 0; i < 5; i++) {  
            marks[i] = input.nextInt();  
        }  
    }
```

```
    public void displayMarks() {
```

```
        System.out.print("Internal Marks: ");  
        for (int mark : marks) {  
            System.out.print(mark + " ");  
        }  
        System.out.println();  
    }
```

```
}
```

```

package SEE;

import CIE.Internals;
import java.util.Scanner;

public class External extends Internals {

    protected int[] externalMarks = new int[5];
    protected int[] finalMarks = new int[5];
    Scanner input = new Scanner(System.in);

    public External(String USN, String Name, int Sem) {
        super(USN, Name, Sem);
    }

    public void setMarksSEE() {
        System.out.println("Enter the external marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            externalMarks[i] = input.nextInt();
        }
    }

    public void calcFinalMarks() {
        System.out.println("Final Marks:");
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = marks[i] + externalMarks[i];
            System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
        }
    }
}

import SEE.External;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        int n = 3;
        External[] students = new External[n];
        Scanner scanner = new Scanner(System.in);

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

            System.out.print("Name: ");
            String name = scanner.nextLine();

            System.out.print("USN: ");
            String usn = scanner.nextLine();

            System.out.print("Semester: ");
            int sem = scanner.nextInt();
        }
    }
}

```

```
students[i] = new External(usn, name, sem);

students[i].setMarks();
students[i].setMarksSEE();

scanner.nextLine(); // Clear the newline
}

System.out.println("\nStudent Details and Final Marks:");
for (External student : students) {
    student.displayDetails();
    student.displayMarks();
    student.calcFinalMarks();
}

scanner.close();
}
}
```

### Lap program - 6

Create a package CIE which has two classes - Student and Internals. The class <sup>student</sup> 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 ODE 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

→ exit

```
import java.util.*;  
package CIE;  
public class student {  
    public String USN;  
    public String Name;  
    public int Sem;  
    Student(String USN, String Name, int Sem){  
        this.USN=USN;  
        this.Name=Name;  
        this.Sem=Sem;  
    }  
}
```

```
package CIE;
public class Internals extends Student {
    Scanner input = new Scanner (System.in);
    public int[] marks = new int[5];
    public void setmarks() {
        for(int i=0; i<5; i++) {
            System.out.println ("Enter the marks of subject " + i);
            marks[i] = input.nextInt();
        }
    }
}
```

```
package SEE
import CIE.Internals;
import java.util.Scanner;
public class External extends Internals {
    protected int[] externalmarks = new int[5];
    protected int[] finalmarks = new int[5];
    Scanner input = new Scanner (System.in);
    public void setmarksSEE() {
        System.out.println ("Enter the marks of subjects");
        for(int i=0; i<5; i++) {
            externalmarks[i] = input.nextInt();
        }
    }
    public void calcfinalmarks() {
        for(int i=0; i<5; i++) {
            finalmarks[i] = Maths[i] + externalmarks[i];
        }
        System.out.println (finalmarks[i]);
    }
}
```

```

import SEE.external;
import java.util.Scanner;
import CIF.internal;
public class Main {
    public static void main(String[] args) {
        int n = 3;
        External[] students = new External[n];
        Scanner s = new Scanner(System.in);
        for (int i = 0; i < n; i++) {
            students[i] = new External();
            System.out.println("Name:");
            String name = s.nextLine();
            System.out.println("USN:");
            String USN = s.nextLine();
            System.out.println("Sem:");
            int marks = s.nextInt();
            students[i] = Student(USN, name, Sem);
            students[i].setMarks();
            students[i].setMarksSEE();
            students[i].calcFinalMarks();
        }
    }
}

```

3 3 3

Enter the number of students : 1

Enter detail of student 1

Enter USN:

IBM2CCS008

Enter Name:

Amar

Enter Sem:

Enter marks for course 1:

20

Enter marks for course 2:

22

Enter marks for course 3:

18

Enter marks for course 4:

25

Enter marks for course 5:

24

Enter External marks for 5 courses:

Enter marks for course 1:

55

Enter marks for course 2:

60

Enter marks for course 3:

50

Enter marks for course 4:

70

Enter marks for course 5:

65

Student Details

USN: IBM21CS008

Name: Amar

Semester: 5

Tutorial Marks:

Course 1: 20

Course 2: 22

Course 3: 18

Course 4: 25

Date \_\_\_\_\_  
Page \_\_\_\_\_

External Marks

course 1: 55

course 2: 60

course 3: 50

course 4: 70

course 5: 65

Final marks

course 1: 75

course 2: 82

course 3: 68

course 4: 95

course 5: 89

## OUTPUT:

```
Enter the USN:  
1RV22CS001  
Enter the Name:  
John Doe  
Enter the Semester:  
5  
Enter the internal marks for 5 subjects:  
20  
22  
18  
19  
24  
Enter the external marks for 5 subjects:  
68  
65  
55  
78  
75  
  
Final Marks:  
Subject 1: 80  
Subject 2: 87  
Subject 3: 73  
Subject 4: 89  
Subject 5: 99
```

## PROGRAM 7

```
import java.util.Scanner;
class wrongage extends Exception{
    public wrongage(String msg){
        super(msg);
    }
}
class father{
    int age;
    public father(int age) throws wrongage{
        if(age<0){
            throw new wrongage("age can not be negative");
        }
        this.age=age;
    }
}
class son extends father{
    int sage;
    public son(int fage,int sage) throws wrongage{

        super(fage);
        if(sage<0){
            throw new wrongage("son age can not be negative");
        }
        if(sage>=fage){
            throw new wrongage("son age can not be greater than father age");
        }
        this.sage=sage;
    }
}
public class lab7{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        System.out.print("enter father age=");
        int fage=sc.nextInt();
        System.out.print("enter son age=");
        int sage=sc.nextInt();
        try{
            son son=new son(fage,sage);
            System.out.print("the difference is");
            System.out.print(fage-sage);
        }
        catch (Exception e){
            System.out.println(e.getMessage());
        }
    }
}
```

### Lab program - 7

Write a program that demonstrates handling of exception, 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's and son's age throws an exception if son age is  $\geq$  father's age.

Class WrongageException extends Exception {

```
public WrongageException (String msg) {  
    super (msg);  
}
```

class Father {

int age;

public Father (int age) throws WrongageException

if (age < 0)

throw new WrongageException ("Father's age

cannot be -ve")

this. age = age;

class Son extends Father {

int sonAge;

public Son (int fage, int sage) throws WrongageException

super (fage);

if (sage < 0)

throw new WrongageException ("Son's age cannot  
-ve");

if (sonage  $\geq$  fatherage)

throw new WrongageException ("Son's age cannot  
be greater than that of father's  
age");

This sage = sage;

import java.util.\*;  
public class Main {

    public static void main (String [] args) throws  
        Exception {

        Scanner sc = new Scanner (System.in);

        System.out.print ("Enter father's age : ");

        int fage = sc.nextInt();

        System.out.print ("Enter son's age : ");

        int sage = sc.nextInt();

    try {

        Son s = new Son (fage, sage);

        System.out.print ("The difference is : ");

        System.out.print (fage - sage);

    }

    catch (WrongAgeException e) {

        System.out.print (e.getMessage());

    }

    }

O/P : Enter father's age : 15

Enter son's age : 21.

WrongAgeException : Son's age can't be greater than that  
of father's age

's age can't be  
);

age cannot  
father's

## OUTPUT:

```
C:\Users\lenovo\Desktop\CODING\java>cd "c:\Users\lenovo\Desktop\CODING\java\" && javac lab7.java && java lab7
enter father age=28
enter son age=9
the difference is19
C:\Users\lenovo\Desktop\CODING\java>cd "c:\Users\lenovo\Desktop\CODING\java\" && javac lab7.java && java lab7
enter father age=25
enter son age=29
son age can not be greater than father age
```

**PROGRAM 8:**

```
class mythread1 extends Thread{
    public void run(){
        try{
            while(true){
                System.out.println("Bmsce");
                Thread.sleep(10000);
            }
        }
        catch(Exception e){
            System.out.println("Thread 1 in interrupted");
        }
    }
}
class mythread2 extends Thread{
    public void run(){
        try{

            while(true){
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        }
        catch(Exception e){
            System.out.println("Thread 2 in interrupted");
        }
    }
}
public class lab8{
    public static void main(String[] args){
        mythread1 t1=new mythread1();
        mythread2 t2=new mythread2();
        t1.start();
        t2.start();
    }
}
```

## Lab program - 8

a)

Write a program to create 2 threads, one thread displaying "BMS college of Engineering" once every 1 sec and other displaying "CSE" once every two second

class TwoThread Together {

static class T extends Thread {

@ override

public void run() {

try {

while (true) {

System.out.println("BMSC E");

Thread.sleep(10000);

}

} }

catch (InterruptedException e) {

System.out.println("Thread 1 Interrupted");

} }

static class T2 extends Thread {

@ override

public void run() {

try {

while (true) {

System.out.println("CSE");

Thread.sleep(2000);

}

catch (InterruptedException e) {

System.out.println("Thread 2 interrupted");

} }

} }

thread  
so  
once

public static void main (String [] args) {  
    T1 a = new T1();  
    T2 b = new T2();  
    a.start();  
    b.start();

3

3

O/P      CSE

CSE

CSE

CSE

CSE

BMSCE

CSE

CSE

CSE

CSE

CSE

BMSCE

interrupted");

);

interrupted");

## OUTPUT:

```
C:\Users\lenovo\Desktop\CODING\java>cd "c:\Users\lenovo\Desktop\CODING\java\" && javac lab8.java && java lab8
Bmsce
CSE
CSE
CSE
CSE
CSE
Bmsce
CSE
CSE
CSE
CSE
CSE
CSE
Bmsce
CSE
```

**PROGRAM 9:**

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

class Division implements ActionListener {
    static JLabel label1, label2, resultLabel;
    static JTextField textField1, textField2;

    public static void main(String[] args) {
        JFrame frame = new JFrame("Division of Numbers");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 200);

        label1 = new JLabel("Enter the first number: ");
        label2 = new JLabel("Enter the second number: ");
        resultLabel = new JLabel("Result will appear here.");

        textField1 = new JTextField(10);
        textField2 = new JTextField(10);

        JButton submit = new JButton("Submit");

        Division division = new Division();
        submit.addActionListener(division);

        JPanel panel = new JPanel();
        panel.add(label1);
        panel.add(textField1);
        panel.add(label2);
        panel.add(textField2);
        panel.add(submit);
        panel.add(resultLabel);

        frame.add(panel);
        frame.setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        try {
            int num1 = Integer.parseInt(textField1.getText());
            int num2 = Integer.parseInt(textField2.getText());

            if (num2 == 0) {
                resultLabel.setText("Error: Division by zero!");
            } else {
                int result = num1 / num2;
                resultLabel.setText("The result of the division is: " + result);
            }
        } catch (NumberFormatException ex) {
            resultLabel.setText("Error: Please enter valid numbers.");
        }
    }
}
```

}

15/12/2014

## Lab program - 9

```
import java.util.*;
public class Division {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print ("Enter num1: ");
        int a2
```

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 NumberFormat exception. If num2 were zero, the program would throw an Arithmetic exception. Display the exception in message dialog box.

```
import javax.swing.*;  
import java.awt.event.*;  
class division implements ActionListener {  
    static JLabel l3;  
    static JTextField t1, t2;  
    public static void main (String [] args) {  
        JFrame frame = new JFrame ("division of two  
        numbers");  
        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);  
        frame.setSize (400, 400);  
        JLabel l1 = new JLabel ("enter the first number");  
        JLabel l2 = new JLabel ("enter the second number");  
        l3 = new JLabel ("1111");  
        JPanel p = new JPanel();  
        t1 = new JTextField (10);  
        t2 = new JTextField (10);  
        JButton submit = new JButton ("Submit");  
        Division dc = new Division();  
        submit.addActionListener (dc);  
        p.add (l1);  
        p.add (t1);  
        p.add (l2);  
        p.add (t2);  
        p.add (l3);  
        p.add (submit);  
        frame.add (p);  
        frame.setVisible (true);  
    }  
}
```

```

public void actionPerformed(ActionEvent e) {
    String s = e.getActionCommand();
    if (s.equals("submit")) {
        try {
            int num1 = Integer.parseInt(t1.getText());
            int num2 = Integer.parseInt(t2.getText());
            num1 /= num2;
            String ans = "The answer of the division is " + num1;
            t3.setText(ans);
        } catch (Exception f) {
            JFrame frame = new JFrame("error occurred");
            frame.setSize(300, 100);
            JLabel l1 = new JLabel(f.toString());
            frame.add(l1);
            frame.setVisible(true);
        }
    }
}

```

O/P

division of two numbers -  X

enter the first number

enter the second number

The answer of the division is 4

Date \_\_\_\_\_  
Page \_\_\_\_\_

division of two number - [ ] x [ ]

enter the first number [ 1 ]

enter the second number [ 0 ]

[ Submit ]

getText(),  
getText();

num1;

[ error occurred ] - [ ] x [ ]

java.lang.ArithmeticException/by zero;

urred");

Program

04/11/12

**PROGRAM 10:**

```
class SharedResource {  
    synchronized void method1(SharedResource other) {  
        System.out.println(Thread.currentThread().getName() + " is executing method1");  
        try { Thread.sleep(100); } catch (InterruptedException e) {}  
        System.out.println(Thread.currentThread().getName() + " is trying to call method2 on other  
resource");  
        other.method2(this);  
    }  
  
    synchronized void method2(SharedResource other) {  
        System.out.println(Thread.currentThread().getName() + " is executing method2");  
        try { Thread.sleep(100); } catch (InterruptedException e) {}  
        System.out.println(Thread.currentThread().getName() + " is trying to call method1 on other  
resource");  
        other.method1(this);  
    }  
}  
  
class DeadlockDemo {  
    public static void main(String[] args) {  
        SharedResource res1 = new SharedResource();  
        SharedResource res2 = new SharedResource();  
  
        Thread t1 = new Thread(() -> res1.method1(res2), "Thread-1");  
        Thread t2 = new Thread(() -> res2.method1(res1), "Thread-2");  
  
        t1.start();  
        t2.start();  
    }  
}
```

'Lab program - 10

```
class SharedResource {  
    synchronized void method1(SharedResource other){  
        System.out.println(Thread.currentThread().  
            getName() + " is executing  
            method1");  
    }  
}
```

```
try { Thread.sleep(100); }  
catch (InterruptedException e) {}  
System.out.println(Thread.currentThread().getName() +  
    " is trying to call method 2 on other resource");  
other.method2(this);  
}  
}
```

```
synchronized void method2(SharedResource other){  
    System.out.println(Thread.currentThread().getName() +  
        " is executing method2");  
}
```

```
try { Thread.sleep(100); }  
catch (InterruptedException e) {}  
System.out.println(Thread.currentThread().getName() +  
    " is trying to call method 1 on other resource");  
other.method1(this);  
}  
}
```

```
class DeadlockDemo {
```

```
public static void main(String[] args) {  
    SharedResource res1 = new SharedResource();  
    SharedResource res2 = new SharedResource();  
    Thread t1 = new Thread(() -> res1.method(res2),  
        "Thread-1");  
    Thread t2 = new Thread(() -> res2.method(res1),  
        "Thread-2");  
}
```

```
t1.start();  
t2.start();  
}
```

OUTPUT:

Thread-1 is executing method1

Thread-2 is executing method1

Thread-1 is trying to call method2 on other resource

Thread-2 is trying to call method2 on other resource