

**B.M.S. COLLEGE OF ENGINEERING**  
Basavanagudi, Bengaluru- 560019  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**LAB REPORT**

On

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

Submitted By:  
**AJAY N M**  
**1BM22CS026**

*In partial fulfilment of*  
**BACHELOR OF ENGINEERING**  
In  
**COMPUTER SCIENCE AND ENGINEERING**  
2023-24

Faculty-In-Charge  
**Swathi Sridharan**  
Assistant Professor  
Department of Computer Science and Engineering

SL. NO	DATE	TOPIC	PageNo
1	22/12/23	Quadratic Equation	1
2	29/12/23	Student SGPA Calculator	3
3	12/01/24	Book Problem	6
4	12/01/24	Shapes	9
5	19/01/24	Bank Problem	12
6	02/02/24	Student External And Internal Marks	18
7	16/02/24	Exception Handling	22
8	16/02/24	Threads	25

## **LAB-1: 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 discriminate  $b^2-4ac$  is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;

public class SquareRoot{

    public static void main(String[] args){

        System.out.println("1BM22CS026 ");

        System.out.println("AJAY N M");

        Scanner s= new Scanner(System.in);

        System.out.println("Enter the coefficient");

        int a=s.nextInt();

        int b=s.nextInt();

        int c=s.nextInt();

        if(a==0){

            System.out.println("Enter valid value of a");

        }

        else{

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

            if(d>0){

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

                float r1= (float)((-b+Math.sqrt(d))/(2*a));

                float r2= (float)((-b-Math.sqrt(d))/(2*a));

                System.out.println(r1);

                System.out.println(r2);

            }

            else if(d<0){

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

            }

        }

    }

}
```

```

    }
    else{
        System.out.println("Roots real and equal");
        float r=(float)-b/(2*a);
        System.out.println(r);
    }
}
}
}
}

```

#### OUTPUT :

```

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>javac QuadraticSolver.java

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>java QuadraticSolver
AJAY N M
1BM22CS026
Enter the coefficients of the quadratic equation ax^2 + bx + c = 0:
Enter a: 1
Enter b: -2
Enter c: 1
Real Solution:
Root: 1.0

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>java QuadraticSolver
AJAY N M
1BM22CS026
Enter the coefficients of the quadratic equation ax^2 + bx + c = 0:
Enter a: 1
Enter b: -3
Enter c: 2
Real Solutions:
Root 1: 2.0
Root 2: 1.0

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>java QuadraticSolver
AJAY N M
1BM22CS026
Enter the coefficients of the quadratic equation ax^2 + bx + c = 0:
Enter a: 1
Enter b: 1
Enter c: 1
No real solutions exist for the given quadratic equation.

```

## **LAB-2: STUDENT SGPA CALCULATION**

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

```
import java.util.Scanner;

public class Sgpa {

    String usn;

    String name;

    private static int credit[] = {4,4,3,3,3,1,1,1};

    private int numOfSub = 8;

    int marks[] = new int [numOfSub];

    Scanner s = new Scanner(System.in)

    Student()

    {

        System.out.println("Enter your details below to calculate your SGPA\n");

        get_details();

        set_marks();

        display();

    }

    public void get_details()

    {

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

        usn = s.next();

        System.out.println("Enter your name:");

        name = s.next();

    }

}
```

```

public void set_marks()
{
    System.out.println("Enter your marks in order");
    for(int i=0;i<numOfSub;++i)
    {
        marks[i] = s.nextInt();
    }
}

public double sgpa()
{
    double sgpa=0,temp=0;
    for(int i=0;i<numOfSub;++i)
    {
        if(marks[i]>=40) {
            if(marks[i]==100)
            {
                temp += credit[i] * ((int) (marks[i] / 10) );
            }
            else
            {
                temp += credit[i] * ((int) (marks[i] / 10)+1 );
            }
        }
        else
        {
            temp+= 0;
        }
    }
}

```

```

    }

    sgpa= temp/20;

    return sgpa;

}

public void display()
{
    System.out.println("Name: "+name);

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

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

public static void main(String[] args) {

    System.out.println("1BM22CS026 ");

    System.out.println("AJAY N M");

    Student s1 = new Student();
}

```

#### OUTPUT :

```

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>javac Student.java
C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>java Student
AJAY N M
1BM22CS026
Enter your USN:
1BM22CS026
Enter your name:
AJAYNM
Enter your marks in order
81
78
64
58
63
66
69
74
Name: AJAYNM
USN: 1BM22CS026
SGPA: 7.5

```

### **LAB-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( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;

class book
{
    String name;
    String author;
    float price;
    int num_pages;

    void set_details()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter bookname,author,price,num_pages");
        name=sc.next();
        author=sc.next();
        price=sc.nextFloat();
        num_pages=sc.nextInt();
    }

    void get_details()
    {
        String details=toString();
        System.out.println(details);
    }
}
```



```

public String toString()
{
    return "the book "+name+" was written by "+author+" it consists of "+num_pages+"
pages and costs around "+price;
}

public static void main(String []args)
{
    System.out.println("1BM22CS026 ");
    System.out.println("AJAY N M");
    Scanner s=new Scanner(System.in);
    System.out.println("enter no of books you want to generate");
    int n=s.nextInt();
    book b[]=new book[n];
    for(int i=0;i<n;i++)
    {
        b[i]=new book();
        b[i].set_details();
    }
    System.out.println("book details");
    System.out.println();
    for(int i=0;i<n;i++)
    {
        b[i].get_details();
    }
}
}

```

## OUTPUT :

```
C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>javac D.java

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>java D
AJAY N M
1BM22CS026
Enter number of books
2
Enter details of books 1 in name, author, price, num_pages order
REX
Zhongli]
1000
989
Enter details of books 2 in name, author, price, num_pages order
Guizhong
xiaoyun
1500
600

Details of Book 1
Name: REX
Author: Zhongli]
Price: 1000
No of pages: 989
Details of Book 2
Name: Guizhong
Author: xiaoyun
Price: 1500
No of pages: 600

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>|
```

## **LAB-4: AREA CALCULATION**

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

```
import java.util.Scanner;

abstract class Shape{
    int a,b;

    abstract void printArea();
}

class Rectangle extends Shape{
    Rectangle(int l,int br){
        a=l;
        b=br;
    }

    void printArea(){
        int area=a*b;

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

class Triangle extends Shape{
    Triangle(int ba,int h){
        a=ba;
        b=h;
    }

    void printArea(){
        double area = 0.5*a*b;
```

```

        System.out.println("Area of the triangle: "+area);
    }
}
class Circle extends Shape{
    Circle(int r){
        a=r;
    }
    void printArea(){
        double area=3.14*a*a;
        System.out.println("Area of Circle: "+area);
    }
}
class AreaCalculation {
    public static void main(String[] args) {
        System.out.println("1BM22CS026 ");
        System.out.println("AJAY N M");
        Scanner in=new Scanner(System.in);
        System.out.println("Enter length and breadth of a rectangle:");
        Rectangle rec=new Rectangle(in.nextInt(), in.nextInt());
        rec.printArea();
        System.out.println("Enter base and height of a triangle: ");
        Triangle tri = new Triangle(in.nextInt(), in.nextInt());
        tri.printArea();
        System.out.println("Enter the radius of a circle:");
        Circle cir = new Circle(in.nextInt());
        cir.printArea();
    }
}

```

## OUTPUT :

```
C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>javac Main.java
```

```
C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>java Main
```

```
AJAY N M
```

```
1BM22CS026
```

```
Enter length of Rectangle: 5
```

```
Enter width of Rectangle: 4
```

```
Enter base of Triangle: 6
```

```
Enter height of Triangle: 3
```

```
Enter radius of Circle: 4
```

```
Area of Rectangle: 20
```

```
Area of Triangle: 9.0
```

```
Area of Circle: 50.26548245743669
```

## **LAB-5: BANK ACCOUNT 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
- d) Permit withdrawal and update the balance

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

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

```
class Account {  
    String customerName;  
    long accno;  
    String accountType;  
    double balance;  
    public Account(String customerName, long accno, String accountType) {  
        this.customerName = customerName;  
        this.accno = accno;  
        this.accountType = accountType;  
    }  
}
```

```

        this.balance = 0.0;
    }

    public void displayBalance() {
        System.out.println("Account Number: " + accno);
        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Type: " + accountType);
        System.out.println("Balance: $" + balance);
    }
}

class CurAcct extends Account {
    double minBalance;
    double serviceCharge;
    public CurAcct(String customerName, long accno) {
        super(customerName, accno, "Current");
        this.minBalance = 500.0; // Set minimum balance
        this.serviceCharge = 50.0; // Set service charge
    }

    public void withdraw(double amount) {
        if (balance - amount >= minBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Current Balance: $" + balance);
        } else {
            System.out.println("Insufficient funds. Withdrawal not allowed.");
        }
    }

    public void imposeServiceCharge() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge imposed. Current Balance: Rs." + balance);
        }
    }
}

```

```

    }
}
}
class SavAcct extends Account {
    double interestRate;

    public SavAcct(String customerName, long accno) {
        super(customerName, accno, "Savings");
        this.interestRate = 0.05;
    }

    public void depositInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest deposited. Current Balance: $" + balance);
    }

    public void compoundInterest(double initialAmount, int term) {
        double compoundInterest = initialAmount * Math.pow((1 + interestRate), term) -
        initialAmount;
        balance += compoundInterest;
        System.out.println("Compound Interest deposited. Current Balance: Rs." + balance);
    }
}

public class bank {
    public static void main(String[] args) {
        System.out.println("1BM22CS026");
        System.out.println("AJAY N M\n");
        Scanner scanner = new Scanner(System.in);
        System.out.println("Choose account type:");
    }
}

```



```

System.out.println("1. Current");
System.out.println("2. Savings");
System.out.print("Enter choice (1 or 2): ");
int choice = scanner.nextInt();
System.out.print("Enter customer name: ");
String customerName = scanner.next();
System.out.print("Enter account number: ");
long accno = scanner.nextLong();
if (choice == 1) {
    CurAcct curAccount = new CurAcct(customerName, accno);
    System.out.print("Enter initial balance: $");
    double initialBalance = scanner.nextDouble();
    curAccount.balance = initialBalance;
    System.out.print("Enter withdrawal amount: $");
    double withdrawalAmount = scanner.nextDouble();
    curAccount.withdraw(withdrawalAmount);
    curAccount.imposeServiceCharge();
    curAccount.displayBalance();
} else if (choice == 2) {
    SavAcct savAccount = new SavAcct(customerName, accno);
    System.out.print("Enter initial balance: $");
    double initialBalance = scanner.nextDouble();
    savAccount.balance = initialBalance;
    System.out.print("Enter withdrawal amount: $");
    double withdrawalAmount = scanner.nextDouble();
    savAccount.balance -= withdrawalAmount;
    System.out.println("Withdrawal successful. Current Balance: $" +
savAccount.balance);
    System.out.print("Enter interest rate: ");
    double interestRate = scanner.nextDouble();

```

```
savAccount.interestRate = interestRate;

System.out.print("Enter term (in years) for compound interest calculation: ");

int term = scanner.nextInt();

savAccount.compoundInterest(initialBalance, term);

savAccount.displayBalance();

} else {

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

}

}

}
```

## OUTPUT :

```
C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>javac Bank.java

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>java Bank
AJAY N M
1BM22CS026
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 1
Enter customer name: AJAY
Enter account number: 12345
Enter initial balance: $1000000
Enter withdrawal amount: $6000
Withdrawal successful. Current Balance: $994000.0
Account Number: 12345
Customer Name: AJAY
Account Type: Current
Balance: $994000.0

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>java Bank
AJAY N M
1BM22CS026
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 2
Enter customer name: AJAY
Enter account number: 12345
Enter initial balance: $1000000
Enter withdrawal amount: $50000
Withdrawal successful. Current Balance: $950000.0
Enter interest rate: 8
Account Number: 12345
Customer Name: AJAY
Account Type: Savings
Balance: $950000.0
Enter term (in years) for compound interest calculation: 10
Compound Interest deposited. Current Balance: $3.48678440095E15
Account Number: 12345
Customer Name: AJAY
Account Type: Savings
Balance: $3.48678440095E15
```

## **LAB-6: CALCULATION OF MARKS**

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.

```
//CIE PACKAGE

package CIE;

import java.util.*;

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

    public void accept()
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter U , N , S:\n");
        usn=scan.nextLine();
        name=scan.nextLine();
        sem=scan.nextInt();
    }
}

//Internals.java

package CIE;
```

```

public class Internal {
    public int im[] = new int[5];
}

//SEE PACKAGE
package SEE;
import CIE.Student;
public class External extends Student {
    public int sm[] = new int[5];
}

//FinalMarks.java
import java.util.*;
import SEE.*;
import CIE.*;
public class Finalmarks
{
    public static void main(String args[])
    {
        System.out.println("Name: AJAY N M");
        System.out.println("USN: 1BM22CS026 ");
        int fm[]=new int[5];
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter n: ");
        int n=sc.nextInt();
        SEE.External st[]=new SEE.External[n];
        CIE.Internal s[]=new CIE.Internal[n];
        for(int i=0; i<n; i++)

```

```

{
    st[i]=new SEE.External();
    s[i]=new CIE.Internal();
    System.out.println("Enter details "+(i+1));
    st[i].accept();
    for(int j=0; j<5; j++)
    {
        System.out.println("Enter im and sm of sub "+(j+1));
        s[i].im[j]=sc.nextInt();
        st[i].sm[j]=sc.nextInt();
        fm[j]=s[i].im[j]+st[i].sm[j];
    }
    System.out.println("Final marks of "+st[i].name);
    for(int k=0; k<5; k++)
    {
        System.out.println("Course "+(k+1)+" = "+fm[k]);
    }
}
}

```

## OUTPUT :

```
C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY\package>java Finalmarks
Enter n:
2
Enter details 1
Enter the details :

1BM22CS026
AJAYNM
3
Enter im and sm of sub 1
81
86
Enter im and sm of sub 2
75
64
Enter im and sm of sub 3
85
84
Enter im and sm of sub 4
80
85
Enter im and sm of sub 5
78
79
Final marks of AJAYNM
Course 1 = 167
Course 2 = 139
Course 3 = 169
Course 4 = 165
Course 5 = 157
```

## **LAB-7: 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 cases both father and son’s age and throws an exception if son’s age is >=father’s age.

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

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

class Father {
    int fatherAge;

    public Father(int fatherAge) throws WrongAge {
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
        this.fatherAge = fatherAge;
    }
}

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 must be less than Father's age");
        }
    }
}
```



```

    }
    this.sonAge = sonAge;
}
}
class AgeOfSonFather {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Name: AJAY N M");
        System.out.println("USN: 1BM22CS026 ");
        System.out.println("Enter father's age and son's age: ");
        int fa=sc.nextInt();
        int sa=sc.nextInt();
        try {
            Son s = new Son(fa, sa);
            System.out.println("Father's age: " + s.fatherAge);
            System.out.println("Son's age: " + s.sonAge);
        }
        catch (WrongAge e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```

## OUTPUT :

```
C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>javac fatherson.java

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>java fatherson
AJAY N M
1BM22CS026
Enter father's age and son's age:
60
20
Father's age: 60
Son's age: 20

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>java fatherson
AJAY N M
1BM22CS026
Enter father's age and son's age:
20
60
Error: Son's age must be less than Father's age
```

## **LAB-8: MULTI THREADING**

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

```
class One extends Thread
{
    int t1;
    One(){
        t1=1;

    }
    public void run()
    {
        while(t1<=2)
        {
            System.out.println("BMS COLLEGE OF ENGINEERING");
            try {
                sleep(10000);
            } catch (Exception e) {
                System.out.println(e.toString());
            }
            t1++;
        }
    }
}

class Two extends Thread{
    int t2;
    Two(){
```

```

        t2=1;
    }
    public void run()
    {
        while(t2<=5)
        {
            System.out.println("CSE");
            try{
                sleep(2000);
            }
            catch(Exception e)
            {
                System.out.println(e.toString());
            }
            t2++;
        }
    }
}

class Threads
{
    public static void main(String args[])
    {
        System.out.println("Name: AJAY N M");
        System.out.println("USN: 1BM22CS026 \n");
        One t1=new One();
        Two t2=new Two();
        t1.start();
        t2.start();
    }
}

```

## OUTPUT :

```
C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>javac th.java

C:\Users\REX LAPIS\Desktop\1BM22CS026_AJAY N M>java th
AJAY N M
1BM22CS026
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
CSE
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
```