

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:

ADITYA SINGH
1BM22CS022

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 Calculation	3
3	12/01/24	Book Details	6
4	12/01/24	Area Calculation	9
5	19/01/24	Bank Account Details	12
6	02/02/24	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 discriminant 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) {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the coefficients : ");
        int a = s.nextInt();
        int b = s.nextInt();
        int c = s.nextInt();
        if (a == 0)
            System.out.println ("Enter valid value for a");
        else
            int d = b * b - 4 * a * c;
            if (d > 0)
                System.out.println ("Roots are real and distinct");
            float r1 = (-b + Math.sqrt (d)) / (2 * a);
            float r2 = (-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 are
                        real and equal");
    float r = (float) -b / (2*a);
    System.out.println(r);
```

OUTPUT :

```
Enter the coefficient
1 -2 1
Roots real and equal
1.0
PS C:\Users\whys0\OneDrive\Desktop\1BM22CS044> java Lab2_1
Aditya Singh
1BM22CS022

Enter the coefficient
1 -3 2
Roots are real and distinct
2.0
1.0
PS C:\Users\whys0\OneDrive\Desktop\1BM22CS044>
```

LAB-2: STUDENT SGPA CALCULATOR

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 student
{
    String usn;
    String name;
    private static int credit[] = {4, 4, 3, 3, 3, 3, 2, 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");
        getDetails();
        getMarks();
        display();
    }
    public void getDetails()
    {
        System.out.println("Enter your USN:");
        usn = s.nextLine();
    }
```

```
System.out.println("Enter your name:");
name = s.nextLine();
}

public void setMarks()
{
    System.out.println("Enter your
marks:");
    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]*10;
            else
                temp += credit[i]*((int)(marks[i]
+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[])
{
    student s1 = new student();
}
```

OUTPUT :

Aditya Singh
1IBM22CS022

Enter your details below to calculate your SGPA

Enter your USN:

IBM22CS022

Enter your name:

ADI

Enter your marks in order

100 98 97 90 92 99 100 98

Name: ADI

USN: IBM22CS022

SGPA: 10.0

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.

The image shows handwritten Java code on lined paper. The code defines a class named Book with four members: name, author, price, and num_pages. It includes a constructor and a setDetails() method that uses Scanner to read input from System.in. The code is written in a cursive style with some corrections and annotations.

```
import java.util.Scanner;
class book
{
    String name;
    String author;
    float price;
    int num_pages;

    void setDetails()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the book name, author name, price and number of pages");
        name = s.next();
        author = s.next();
        price = s.nextFloat();
        num_pages = s.nextInt();
    }
}
```

```

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

    void toString() {
        return "The book " + name + " is  

            written by " + author + " consists of "  

            + numPages + " pages and cost  

            around " + price;
    }

    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the  

            number of books");
        int n = s.nextInt();
        Book b[] = new Book[n];
        for (int i = 0; i < n; i++) {
            b[i] = new Book();
            b[i].setDetails();
        }
        for (int i = 0; i < n; i++) {
            b[i].getDetails();
        }
    }
}

```

OUTPUT :

```
Aditya Singh  
1BM22CS022
```

```
enter no of books you want to generate
```

```
2
```

```
enter bookname,author,price,num_pages
```

```
book1
```

```
author1
```

```
250
```

```
500
```

```
enter bookname,author,price,num_pages
```

```
book2
```

```
author2
```

```
250
```

```
500
```

```
book details
```

```
the book book1 was written by author1 it consists of 500 pages and costs around 250.0
```

```
the book book2 was written by author2 it consists of 500 pages and costs around 250.0
```

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  
  
abstract class shape  
{  
    int a,b;  
    abstract void printArea();  
}  
  
class rectangle extends shape  
{  
    rectangle (int l, int b)  
    {  
        a=l; b=b;  
    }  
    void printArea()  
    {  
        System.out.println ("Area of  
rectangle : " + (l * b));  
    }  
}
```

```
class triangle extends shape  
{  
    int base, height;  
    double area;  
    public triangle (int ba, int h)  
    {  
        base = ba;  
        height = h;  
        area = 0.5 * base * height;  
    }  
    void printArea()  
    {  
        System.out.println ("Area of  
triangle : " + area);  
    }  
}
```

```
class circle extends shape  
{  
    int radius;  
    double area;  
    public circle (int r)  
    {  
        radius = r;  
        area = 3.14 * radius * radius;  
    }  
    void printArea()  
    {  
        System.out.println ("Area of  
circle : " + area);  
    }  
}
```

```
class E  
{  
    public static void main (String [] args)  
    {  
        Scanner s = new Scanner (System.in);  
    }  
}
```

System.out.println("Enter the length
and breadth of rectangle");
rectangle rec = new rectangle
(s.nextInt(), s.nextInt());
rec.printArea();

System.out.println("Enter the
base and height of triangle");
triangle tri = new triangle (s.nextInt(),
s.nextInt());
tri.printArea();

System.out.println("Enter the
radius of circle");
circle cr = new circle (s.nextInt());
cr.printArea();

OUTPUT:

Aditya Singh
1BM22CS022

Enter length and breadth of a rectangle:

20 15

Area of rectangle: 300

Enter base and height of a triangle:

15 20

Area of the triangle: 150.0

Enter the radius of a circle:

15

Area of Circle: 706.5

PS C:\Users\whys0\OneDrive\Desktop\1BM22CS044> █

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 acctype;
```

```
    double balance;
```

```
    public Account (String customername,  
                    long accno, String acctype)
```

```
    {  
        this.customername = customername;
```

```
        this.accno = accno;
```

```
        this.acctype = acctype;
```

```
        this.balance = 0.0  
    }
```

```
    public void displayBalance ()
```

```
        System.out.println ("Customer  
Name " + customername);
```

```
        System.out.println ("Account  
Number " + accno);
```

```
class CurAcc extends Acc
```

```

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

public void compInterest()
{
    compInterest(double initialAmt,
                 int term)
    {
        double compInterest = initialAmt
            * Math.pow(1 + interestRate, term)
            - initialAmt;
        balance += compInterest;
        System.out.println("Compound
                           interest deposited. Current balance : "
                           + balance);
    }
}

```

```

public class Bank
{
    public static void main(String[] args)
    {
        Scanner p = new Scanner(System.in);
        System.out.println("Choose
                           account type");
        System.out.println("1. Current
                           2. Savings");
    }
}

```

```

System.out.println("Enter choice (1 or 2)
int ch = s.nextInt();
System.out.print("Enter customer
name, Account number.");
String customerName = s.next();
long accno = s.nextLong();
if (ch == 1) {
    CurAcc curacc = new CurAcc
        (customerName, accno);
    System.out.println("Enter
initial balance");
    double initialBalance =
        s.nextDouble();
    curacc.balance = initialBalance;
    System.out.print("Enter with
drawal amt:");
    double withdrawalAmount =
        s.nextDouble();
    curacc.withdraw(withdrawalAmt);
    curacc.imposeServiceCharge();
    curacc.displayBalance();
}
else if (ch == 2) {
    SavAcc savacc = new SavAcc
        (customerName, accno);
    System.out.print("Enter initial
balance");
    double initialBalance = s.nextDouble();
    savacc.balance = initialBalance;
    System.out.print("Enter with
amt");
    double withdrawalAmount = s.nextDouble();
}

```

```
System.out.print("Enter interest  
rate");  
double interestRate = s.nextDouble();  
savacc.interestRate = interestRate;  
savacc.displayBalance();  
System.out.print("Enter term for  
Compound interest");  
int term = s.nextInt();  
savacc.compInterest(initialBalance,  
term);  
savacc.displayBalance();  
}  
else if (choice == 2)  
{  
    System.out.println("Invalid choice");  
}
```

OUTPUT :

```
Aditya Singh
1BM22CS022

Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 1
Enter customer name: adi
Enter account number: 1232
Enter initial balance: $1000
Enter withdrawal amount: $300
Withdrawal successful. Current Balance: $700.0
Account Number: 1232
Customer Name: adi
Account Type: Current
Balance: $700.0
● PS C:\Users\whys0\OneDrive\Desktop\1BM22CS044> java Bank
Aditya Singh
1BM22CS022

Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 2
Enter customer name: adi
Enter account number: 1909
Enter initial balance: $1000
Enter withdrawal amount: $300
Withdrawal successful. Current Balance: $700.0
Enter interest rate: 700
Enter term (in years) for compound interest calculation: 5
Compound Interest deposited. Current Balance: Rs.1.69273934903500704E17
Account Number: 1909
Customer Name: adi
Account Type: Savings
Balance: $1.69273934903500704E17
○ PS C:\Users\whys0\OneDrive\Desktop\1BM22CS044>
```

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.

```
package CIE;
import java.util.Scanner;
public class Student {
    public int usn;
    public String name;
    public int sem;
    public void accept() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter
usn, name, sem\n");
        usn = s.nextInt();
        name = s.nextLine();
        sem = s.nextInt();
    }
}
package CIE;
public class Internals {
    public int iimarks[] = new int[5]
```

```

package SEE;
import CIF.student;

public class External extends student
{
    public int smark() = new
    int [5];
}

import java.util.Scanner;
import java SEE.*;
import CIF.*;

public class FinalMark {
    public static void main (String args[])
    {
        int fm [] = new int [5];
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter n : ");
        int n = s.nextInt ();
        SEE.External st [] = new
        SEE.External [n];
        CIF.Internal s [] = new
        CIF.Internal [n];
        for (int i=0; i<n; i++)
        {
            st [i] = new SEE.External();
            s [i] = new CIF.Internal()
        }
        System.out.println ("Enter
details " + (i+1));
        for (st [i]. accept(); i=0; i<5; i++)
    }
}

```

syro ("Enter im and sm of
sub "+(j+1));
 $s[i].im[j] = sc.nextIn();$
 $st[i].sm[j] = sc.nextIn();$
 $fm[j] = s[i].im[j] + st[i].sm[j];$

syro ("Final marks of "
+ st[i].name);
for (int k=0; k<5; k++)

syro ("Course "+(k+1)+"
= " + fm[k]);

63 : 01 957

final

OUTPUT :

```
D:\java\oops>java Main
Name: Aditya Singh
USN: 1BM22CS022
Enter n:
1
Enter details 1
Enter U, N, S:

15 Adi 2
adi
2
Enter im and sm of sub 1
23 28
Enter im and sm of sub 2
23 9
Enter im and sm of sub 3
45 78
Enter im and sm of sub 4
34 90
Enter im and sm of sub 5
35 90
Final marks of adi
Course 1 = 25
Course 2 = 16
Course 3 = 61
Course 4 = 62
Course 5 = 62

D:\java\oops>
```

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.util.Scanner  
class Wrongage extends Exception  
{ public Wrongage (String message)  
{ super (message);  
}  
  
class Father  
{ int fage;  
public Father (int fage) throws  
Wrongage  
{ if (fage<0){ throw new  
Wrongage ("Age can't be negative");  
}  
this .fage = fage;  
}  
}
```


class son extends father

{ int page;

public static son (int fage, int spage)
throws wrongage

{ super(father fage);
if (page >= fage)

throw new wrongage
("Son's age must be lesser
than father's age");

this.page = page;

Class Main

{ public static void main (String[] args)

Scanner s = new Scanner (System.in)
System.out.println ("Enter father's
age and son's age");

int fage = s.nextInt();

int sage = s.nextInt();

try

{ son p = new son (fage, sage)
System.out.println ("Father's age = "+ s.fage + " Son's age = "+ s.sage)

} catch (wrongage e)

{ System.out.println (e.getMessage()); }

OUTPUT :

```
Name: Aditya Singh
USN: 1BM22CS022
Enter father's age and son's age:
50 15
Father's age: 50
Son's age: 15
java -cp /tmp/30mqW0oyeK AgeOfSonFather
Name: Aditya Singh
USN: 1BM22CS022
Enter father's age and son's age:
20 25
ERROR!
Error: Son's age must be less than Father's age
```

LAB-8: MULTITHREADING

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 i=0;
public void run()
{
    int i=0;
    while(i<5)
    {
        i++;
        try
        {
            System.out.println("Bros");
            Thread.sleep(1000);
        }
        catch(InterruptedException e)
        {
            System.out.println("e. toString()");
        }
    }
}
```

class two extends Thread

```
{ public void run()
{
    int i=0;
```

Page 40

```
while (i < 5)
{
    i++;
    try
    {
        System.out.println("(" + i + ")");
        Thread.sleep(2000);
    }
    catch (InterruptedException e)
    {
        System.out.println("catching exception");
    }
}

class Main
{
    public static void main(String[] args)
    {
        One t1 = new One();
        Two t2 = new Two();
        t1.start();
        t2.start();
    }
}
```

OUTPUT :

```
Aditya Singh  
1BM22CS022
```

```
BMS COLLEGE OF ENGINEERING  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS COLLEGE OF ENGINEERING  
CSE  
CSE  
CSE  
CSE  
CSE  
}
```