VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

Submitted by

MOHAMMAD ADNAN KHAN (1BM21CS107)

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
Oct 2022-Feb 2023

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 **MOHAMMAD ADNAN KHAN (1BM21CS107),** 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 during the year 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of Data structures Lab - **(22CS3PCOOJ)** work prescribed for the said degree.

Rajeshwari Madli
Assistant professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index Sheet

Course Outcome

CO1	Apply the knowledge of Java concepts to find the solution for a given problem
CO2	Analyze the given Java application for correctness/functionalities.
CO3	Develop Java programs / applications for a given requirement.
CO4	Conduct practical experiments for demonstrating features of Java.

LAB PROGRAM 1:

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminant b2-4ac is negative, display a message stating that there are no real solutions.

CODE:

```
import java.util.Scanner;

class qe
{

public static void main (String args[])
 {

Scanner sc = new Scanner (System.in);

System.out.println ("Enter the Values of A,B,C:");

double a = sc.nextDouble ();

double b = sc.nextDouble ();

double c = sc.nextDouble ();

double d;

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

double r1, r2;

if (a == 0)

    {

System.out.println ("Invaild Input");

}

   else if (d > 0)

    {

System.out.println ("Roots are Real and Unique");
```

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

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

System.out.println ("Root 1: " + r1);

System.out.println ("Root 2: " + r2);

}

   else if (d == 0)

    {

System.out.println ("Roots are Real and Equal");

r1 = (-b / (2 * a));

r2 = (-b / (2 * a));

System.out.println ("Root 1:" + r1);

System.out.println ("Root 2:" + r2);

}

   else

    {

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

r1 = (-b / (2 * a));

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

System.out.println ("Root 1:" + r1 + "+i" + r2);

System.out.println ("Root 2:" + r1 + "-i" + r2);

}

}

}
```

OUTPUT:

```
Enter the Values of A,B,C:
4 3 2
Roots are Imaginary
Root 1:-0.375+i0.5994789404140899
Root 2:-0.375-i0.5994789404140899
```

```
Enter the Values of A,B,C:
2 3 1
Roots are Real and Unique
Root 1: -0.5
Root 2: -1.0
```

```
Enter the Values of A,B,C:
1 2 1
Roots are Real and Equal
Root 1:-1.0
Root 2:-1.0
```

```
Enter the Values of A,B,C:
0 1 2
Invaild Input
```

LAB PROGRAM 2:

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

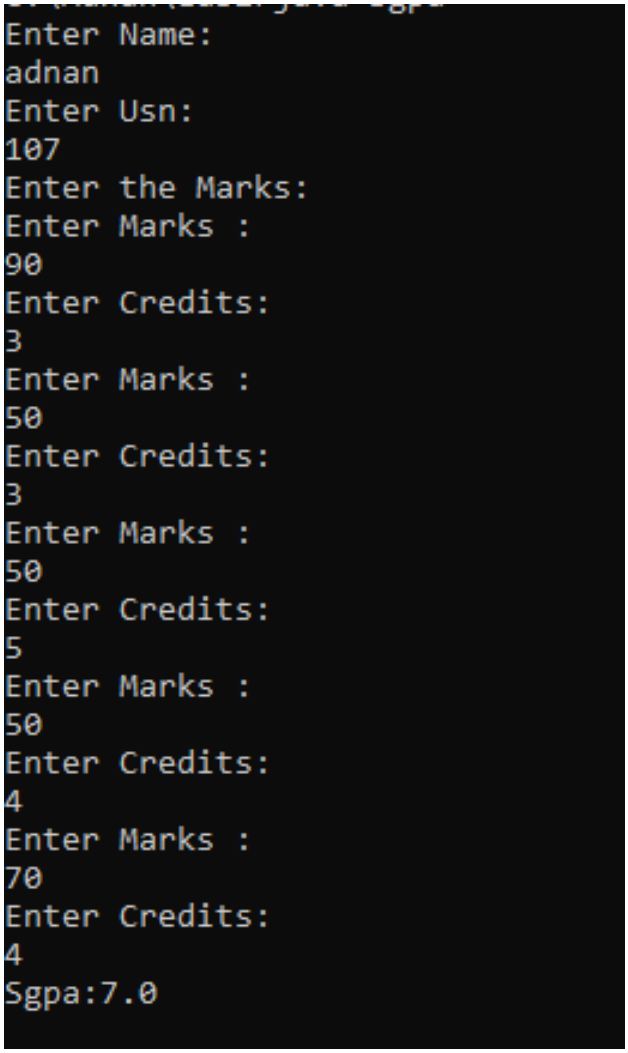
CODE:

```
import java.util.Scanner;


class student
{

String Name, usn;

int marks[] = new int[10];

int credits[] = new int[10];

int gradePoints[] = new int[10];

int a = 0;

int b = 0;

float cred;

void input ()
  {

Scanner sc = new Scanner (System.in);

System.out.println ("Enter Name:");

Name = sc.next ();

System.out.println ("Enter Usn:");

usn = sc.next ();

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

for (int i = 0; i < 5; i++)

    {
```

```
System.out.println ("Enter Marks :");

marks[i] = sc.nextInt ();

System.out.println ("Enter Credits:");

credits[i] = sc.nextInt ();

}
}
void calculator ()
 {

for (int i = 0; i < credits.length; i++)
    {

if (marks[i] >= 90)
           {

gradePoints[i] = 10;

}
         else if (marks[i] < 90 && marks[i] >= 80)
           {

gradePoints[i] = 9;

}
         else if (marks[i] < 80 && marks[i] >= 70)
           {

gradePoints[i] = 8;

}
         else if (marks[i] < 70 && marks[i] >= 60)
           {

gradePoints[i] = 7;

}
         else if (marks[i] < 60 && marks[i] >= 50)
           {

gradePoints[i] = 6;

}
```

```
        else if (marks[i] < 50 && marks[i] >= 40)
         {

gradePoints[i] = 5;

}
        else
         {

gradePoints[i] = 0;

}

a += (credits[i] * gradePoints[i]);

b += credits[i];

}

cred = (float) (a / b);

}


}
class sgpa

{

public static void main (String args[])
  {

student s1 = new student ();

s1.input ();

s1.calculator ();

System.out.println ("Sgpa:" + s1.cred);

}
}
```

OUTPUT:


Enter Name:
adnan
Enter Usn:
107
Enter the Marks:
Enter Marks :
90
Enter Credits:
3
Enter Marks :
50
Enter Credits:
3
Enter Marks :
50
Enter Credits:
5
Enter Marks :
50
Enter Credits:
4
Enter Marks :
70
Enter Credits:
4
Sgpa:7.0

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

CODE:

```
import java.util.Scanner;

import java.lang.*;


class Book1
{

String name;

String author;

float price;

Integer pages;

Book1 (String n, String a, float p, Integer pa)
  {

name = n;

author = a;

price = p;

pages = pa;

}
void display ()
  {

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

System.out.println ("Author : " + author);

String pricee = Float.toString (price);
```

```
System.out.println ("Price : " + pricee);

System.out.println ("Pages : " + pages.toString ());

}
}

class book

{

public static void main (String args[])
  {

int n;

String name;

String author;

float price;

Integer pages;

System.out.println ("Enter the Number of Books: ");

Scanner sc = new Scanner (System.in);

n = sc.nextInt ();

Book1[]arr;

arr = new Book1[n];

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

     {

System.out.println ("Enter Details of Book: " + (i + 1));

name = sc.next ();

author = sc.next ();

price = sc.nextFloat ();
```

```
pages = sc.nextInt ();

arr[i] = new Book1 (name, author, price, pages);

}
for (int i = 0; i < n; i++)

    {

System.out.println ("Book" + (i + 1) + "Details");

arr[i].display ();

}
}
}
```

OUTPUT:

```
Enter the Number of Books:
2
Enter Details of Book: 1
java
cverma
200
200
Enter Details of Book: 2
c
hverma
200
200
Book1Details
Name : java
Author : cverma
Price : 200.0
Pages : 200
Book2Details
Name : c
Author : hverma
Price : 200.0
Pages : 200
```

LAB PROGRAM 4:

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

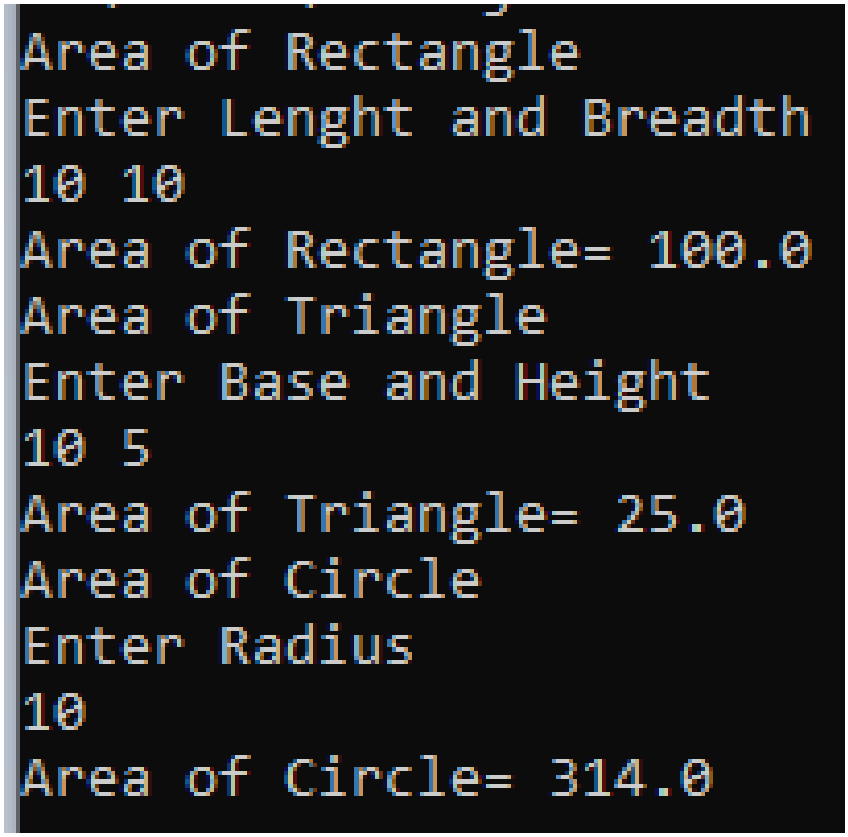
CODE:

```
import java.util.*;

abstract class Shape
{

Scanner sc = new Scanner (System.in);

abstract void printArea ();

}
class Rectangle extends Shape

{

void printArea ()
  {

int l, b;

System.out.println ("Area of Rectangle");

System.out.println ("Enter Lenght and Breadth");

l = sc.nextInt ();

b = sc.nextInt ();

double area = l * b;

System.out.println ("Area of Rectangle=" + " " + area);

}
}
class Triangle extends Shape

{
```

```
void printArea ()
  {

int b, h;

System.out.println ("Area of Triangle");

System.out.println ("Enter Base and Height");

b = sc.nextInt ();

h = sc.nextInt ();

double area = (b * h) / 2;

System.out.println ("Area of Triangle=" + " " + area);

}
}
class Circle extends Shape

{

void printArea ()
  {

int r;

System.out.println ("Area of Circle");

System.out.println ("Enter Radius");

r = sc.nextInt ();

double area = 3.14 * r * r;

System.out.println ("Area of Circle=" + " " + area);

}
}
class curves

{

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

```
Rectangle rec = new Rectangle ();

rec.printArea ();

Triangle tri = new Triangle ();

tri.printArea ();

Circle cir = new Circle ();

cir.printArea ();

}
}
```

OUTPUT:


Area of Rectangle
Enter Lenght and Breadth
10 10
Area of Rectangle= 100.0
Area of Triangle
Enter Base and Height
10 5
Area of Triangle= 25.0
Area of Circle
Enter Radius
10
Area of Circle= 314.0

LAB PROGRAM 5:

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides a check 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 deposits from customers 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.

CODE:

```
import java.util.Scanner;


class Account
{

String customerName;

String accType;

int accNum;


double balance = 1000;
```

```
void deposit (int amount)
  {

balance += amount;

}

void displayBalance ()
  {

System.out.println ("The balance in the account is : " + balance);

}
}

class SavAcct extends Account

{

int n = 4;

double r = 0.07;


void interest (double y)
  {

double x = balance;

balance = balance * Math.pow ((1 + r / n), (y * n));

System.out.println ("An amount of " + (balance - x) +
                                  " has been deposited as interest");

}

void withdrawal (int amount)
  {

if (balance >= amount)
     {

balance -= amount;

}
```

```
   else
    {

System.out.println ("You dont have the sufficient balance");

}

}

}



class CurrAcct extends Account
{

int minBalance = 1000, penalty = 7;


void withdrawal (int amount)
 {

if (balance <= minBalance)
    {

balance -= penalty;

System.out.println ("A penalty of 7 rupees has been imposed!");

}


if (balance >= amount)
    {

balance -= amount;

}
   else
    {

System.out.println ("You dont have the sufficient balance");

}

}
```

```
}



class Bank
{

public static void main (String args[])
  {

Scanner sc = new Scanner (System.in);

System.out.
     println
     (
"Enter the account you want to open :\n1 Savings Account\n2 Current Account : ");

int choice = sc.nextInt ();

if (choice == 1)
     {

SavAcct act = new SavAcct ();

while (true)
            {

System.out.
               println
               (
"Enter the transactions you would like to do :\n1 Deposit\n2 Withdraw\n3 Maintain balance for interest\n4
Display Balance: ");

choice = sc.nextInt ();

switch (choice)
               {

case 1:

System.out.println ("Enter the amount to be deposited : ");

int amount = sc.nextInt ();

act.deposit (amount);

break;
```

```
case 2:

System.out.println ("Enter the amount to be withdrawn : ");

int amt = sc.nextInt ();

act.withdrawal (amt);

break;

case 3:

System.out.println ("Enter the duration in years : ");

double y = sc.nextDouble ();

act.interest (y);

break;

case 4:

act.displayBalance ();

break;

default:

System.out.println ("Enter a valid choice!");

break;

}

}

}


   else if (choice == 2)
    {

CurrAcct act = new CurrAcct ();

while (true)
            {
```

```
System.out.
            println
            (
"Enter the transactions you would like to do :\n1 Deposit via chequebook\n2 Withdraw via chequebook\n3
Display Balance: ");

choice = sc.nextInt ();

switch (choice)
            {

case 1:

System.out.println ("Enter the amount to be deposited : ");

int amount = sc.nextInt ();

if (amount < act.minBalance)
                        {

act.balance -= act.penalty;

System.out.
                              println ("A penalty of 7 rupees has been imposed!");

}

act.deposit (amount);

break;

case 2:

System.out.println ("Enter the amount to be withdrawn : ");

int amt = sc.nextInt ();

act.withdrawal (amt);

break;

case 3:

act.displayBalance ();

break;
```

```
default:

System.out.println ("Enter a valid choice!");

break;

}

}

}


   else
    {

System.out.println ("C
nter a valid choice!");

System.exit (0);

}

sc.close ();

}

}
```

OUTPUT:

```
Enter the account you want to open :
1 Savings Account
2 Current Account :
2
Enter the transactions you would like to do :
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
1
Enter the amount to be deposited :
1000
Enter the transactions you would like to do :
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
3
The balance in the account is : 2000.0
```

```
Enter the transactions you would like to do
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
2
Enter the amount to be withdrawn :
1000
Enter the transactions you would like to do
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
2
Enter the amount to be withdrawn :
200
A penalty of 7 rupees has been imposed!
Enter the transactions you would like to do
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
3
The balance in the account is : 793.0
```

LAB PROGRAM 6:

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.

CODE:

```
import java.util.Scanner;


class WrongAgeException extends Exception

{

public String toString ()

 {

return ("Entered age is negative ");

}

}



class AgeException extends Exception
{

public String toString ()
 {

return ("Age entered of the son is greater than that of the father");

}

}
```

```
class Father
{

int father_age;

Father (int x) throws WrongAgeException
 {

father_age = x;

if (father_age < 0)
    {

throw new WrongAgeException ();

}


}

}


class Son extends Father
{

int son_age;

Son (int x, int y) throws AgeException, WrongAgeException
 {

super (x);

son_age = y;

if (son_age < 0)
    {

throw new WrongAgeException ();

}

if (son_age >= father_age)
    {

throw new AgeException ();
```

```
}

}

}



class Excep
{

public static void main (String[]args)
 {

try
   {

Scanner s = new Scanner (System.in);

System.out.println ("Enter father's and son's ages");

int x = s.nextInt ();

int y = s.nextInt ();

Son so = new Son (x, y);

System.out.println ("Father is  " + so.father_age + " " +
                                "years old and son is  " + so.son_age + "" +
                                " years old");

} catch (WrongAgeException wa)
   {

System.out.println (wa);

}
catch (AgeException a)
   {

System.out.println (a);

}
catch (Exception e)
   {
```

```
System.out.println ("Enter valid values");

}
}
}
```

OUTPUT:

```
Enter father's and son's ages
50
69
Age entered of the son is greater than that of the father

C:\Adnan\Lab7>java Excep
Enter father's and son's ages
10 20
Age entered of the son is greater than that of the father

C:\Adnan\Lab7>java Excep
Enter father's and son's ages
-10
20
Entered age is negative

C:\Adnan\Lab7>java Excep
Enter father's and son's ages
10 -20
Entered age is negative

C:\Adnan\Lab7>java Excep
Enter father's and son's ages
30 10
Father is  30 years old and son is  10 years old
```

LAB PROGRAM 7:

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.

CODE:

```
class thread1 extends Thread
{
   public void run()
   {
      try{//Sleep throws exception
      for(int i=1;i<=3;i++)
      {
      System.out.println("BMS College of Engineering");
      Thread.sleep(10000);
      }
   }
   catch(InterruptedException e)
   {
      System.out.println("Interrupted error "+e);
   }
   }
}
class thread2 extends Thread
```

```
{
   public void run()
   {
      try{
      for(int i=1;i<=5;i++)
      {
      System.out.println("CSE");
      Thread.sleep(2000);
      }
   }
   catch(InterruptedException e)
   {
      System.out.println("Interrupted error "+e);
   }
   }
}
class multithreading
{
   public static void main(String[] args) {
      thread1 t1=new thread1();
      thread2 t2=new thread2();
      t1.start();
      t2.start();
```

```
  }
}
```

OUTPUT:

```
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
```