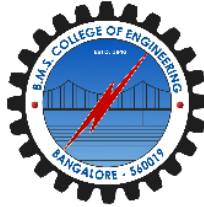


B.M.S. COLLEGE OF ENGINEERING
Basavanagudi, Bengaluru- 560019
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



LAB REPORT

On

Object Oriented Java Programming
(23CS3PCOOJ)

Submitted By :

ADITHYA RAVIKEERTHI
1BM22CS020

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

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

```
import java.util.Scanner;

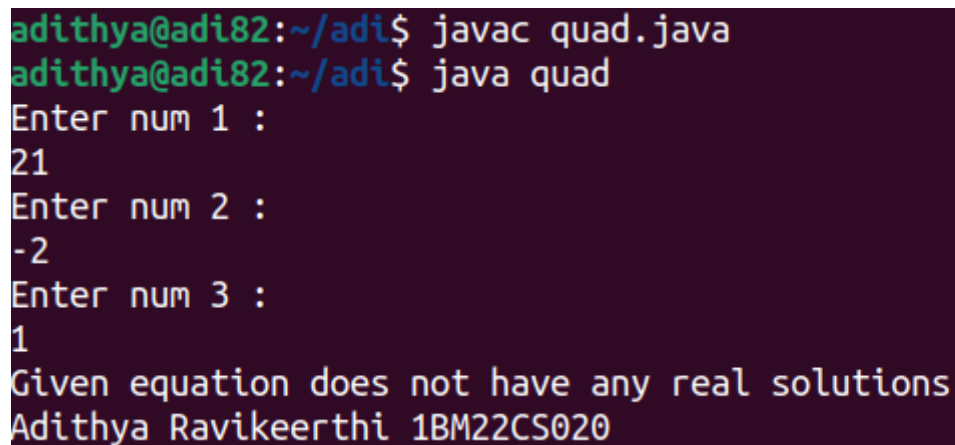
public class quad
{
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter num 1 : ");
        double a = s.nextDouble();
        System.out.println("Enter num 2 : ");
        double b = s.nextDouble();
        System.out.println("Enter num 3 : ");
        double c = s.nextDouble();
        double y = ((b*b)-(4*a*c));
        double z = Math.sqrt(y);
        if(y>0)
        {
            System.out.println("Given equation has 2 real solutions and they are : ");
            double d = ((-b+z)/(2*a));
            double e = ((-b-z)/(2*a));
            System.out.println(d+" and "+e);
        }
        else if(y==0)
        {
            double f = -b/(2*a);
```

```

System.out.println("Given equation has 1 real solution and that is : "+f);
}
else if(y<0)
{
System.out.println("Given equation does not have any real solutons");
}
else
{
System.out.println("Invalid Input");
}
System.out.println("Adithya Ravikeerthi 1BM22CS020");
}
}

```

OUTPUT :



```

adithya@adi82:~/adi$ javac quad.java
adithya@adi82:~/adi$ java quad
Enter num 1 :
21
Enter num 2 :
-2
Enter num 3 :
1
Given equation does not have any real solutions
Adithya Ravikeerthi 1BM22CS020

```

LAB-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. */

```
import java.util.Scanner;

class student
{
    Scanner s = new Scanner(System.in);
    String usn;
    String name;
    int[] credits = {4,4,3,3,3,1,1,1};
    int[] marks = new int[8];
    public void enterdet()
    {
        System.out.print("Enter your usn : ");
        usn = s.next();
        System.out.print("Enter your name : ");
        name= s.next();
        for(int i=0;i<8;i++)
        {
            System.out.print("Enter your marks for subject "+(i+1)+ " : ");
            marks[i] = s.nextInt();
        }
    }

    public void displaydet()
    {
        System.out.println("Your usn is : "+usn);
        System.out.println("Your name is : "+name);
        for(int j=0;j<8;j++)
        {
```

```

System.out.println("Your marks for subject "+(j+1)+ " : "+marks[j]);
}
}

public void sgpa()
{
int g=0;
for(int k =0;k<8;k++)
{
int v=0;
v = credits[k]*((marks[k]/10)+1);
g = g+v;
}
System.out.println("Your sgpa is : "+(g/20));
}
}

public class sgpathing
{
public static void main(String[] args) {
student p = new student();
p.enterdet();
p.displaydet();
p.sgpa();
System.out.println("Adithya Ravikeerthi 1BM22CS020");
}
}

```

OUTPUT:

```
adithya@adi82:~/adi$ javac SGPAProgram.java
adithya@adi82:~/adi$ java SGPAProgram
Enter your USN: 20
Enter your name: adithya
Enter your marks for subject 1: 80
Enter your marks for subject 2: 75
Enter your marks for subject 3: 96
Enter your marks for subject 4: 62
Enter your marks for subject 5: 87
Enter your marks for subject 6: 90
Enter your marks for subject 7: 76
Enter your marks for subject 8: 88

Student Details:
USN: 20
Name: adithya
Marks for subject 1: 80
Marks for subject 2: 75
Marks for subject 3: 96
Marks for subject 4: 62
Marks for subject 5: 87
Marks for subject 6: 90
Marks for subject 7: 76
Marks for subject 8: 88

SGPA: 8.65
```

LAB-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.* /

```
class Book {  
    private String name;  
    private String author;  
    private int price;  
    private int num_of_pages;  
    public Book(String name, String author, int price, int num_of_pages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.num_of_pages = num_of_pages;  
    }  
  
    public void setAuthor(String author) {  
        this.author = author;  
    }  
  
    public void setPrice(int price) {  
        this.price = price;  
    }  
  
    public void setNum_of_pages(int num_of_pages) {  
        this.num_of_pages = num_of_pages;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public String getAuthor() {  
        return author;  
    }  
  
    public int getPrice() {  
        return price;  
    }  
  
    public int getNum_of_pages() {  
        return num_of_pages;  
    }  
}
```

```
public String toString() {  
    return "Book name=" + name + ", author=" + author + ", price=" + price + ", num_of_pages=" +  
    num_of_pages  
    + " ";  
}  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        int n = 2;  
        Book[] b = new Book[n];  
        b[0] = new Book("GOT", "George R. R. Martin", 1000, 2000);  
        b[1] = new Book("Crime and Punishment", "Fyodor Dostoevsky", 2500, 1500);  
        System.out.println(b[0]);  
        System.out.println(b[1]);  
        System.out.println("Adithya Ravikeerthi IBM22CS020");  
    }  
}
```

OUTPUT:

```
adithya@adi82:~/adi$ javac Main.java  
adithya@adi82:~/adi$ java Main  
Book name=GOT, author=George R. R. Martin, price=1000, num_of_pages=2000  
Book name=Crime and Punishment, author=Fyodor Dostoevsky, price=2500, num_of_pages=1500  
Adithya Ravikeerthi IBM22CS020
```


LAB-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 contain only the method printArea() that prints the area of the given shape*/

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

```
abstract class Shape {
```

```
    double a, b;
```

```
    Shape(double a, double b) {
```

```
        this.a = a;
```

```
        this.b = b;
```

```
    }
```

```
    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(double a, double b) {
        super(a, b);
    }

    void printArea() {
        System.out.println("The area is : " + (a * b));
    }
}

class Triangle extends Shape {
    Triangle(double a, double b) {
        super(a, b);
    }

    void printArea() {
        System.out.println("The area is : " + ((a * b) / 2));
    }
}

class Circle extends Shape {
    Circle(double a) {
        super(a, a);
    }

    void printArea() {
```

```

        System.out.println("The area is : " + (3.14 * a * a));
    }
}

public class Main {

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);

        Rectangle r = new Rectangle(5, 10);

        Triangle t = new Triangle(4, 8);

        Circle c = new Circle(6);

        System.out.print("Rectangle: ");

        r.printArea();

        System.out.print("Triangle: ");

        t.printArea();

        System.out.print("Circle: ");

        c.printArea();

        System.out.println("Adithya Ravikeerthi 1BM22CS020");

    }

}

```

OUTPUT:

```

adithya@adi82:~/adi$ javac Main.java
adithya@adi82:~/adi$ java Main
Rectangle: The area is : 50.0
Triangle: The area is : 16.0
Circle: The area is : 113.03999999999999
Adithya Ravikeerthi 1BM22CS020

```

LAB-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 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;
```

```
abstract class Account {
```

```
    protected String customerName;
```

```
    protected long accountNumber;
```

```
    protected String accountType;
```

```
    protected double balance;
```

```
    public Account(String customerName, long accountNumber, String accountType, double balance) {
```

```
        this.customerName = customerName;
```

```
        this.accountNumber = accountNumber;
```

```
        this.accountType = accountType;
```

```
        this.balance = balance;
```

```
    }
```

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

```
        balance += amount;
```

```
        System.out.println("Deposit of $" + amount + " successful.");
```

```
    }
```

```
    public void displayBalance() {
```

```
        System.out.println("Account Balance: $" + balance);
```

```
}
```

```
public void depositInterest(double interestRate) {  
    double interest = balance * interestRate / 100;  
    balance += interest;  
    System.out.println("Interest of $" + interest + " deposited.");  
}
```

```
public void withdraw(double amount) {  
    if (balance >= amount) {  
        balance -= amount;  
        System.out.println("Withdrawal of $" + amount + " successful.");  
    } else {  
        System.out.println("Insufficient funds!");  
    }  
}
```

```
public abstract void checkMinimumBalance();  
}
```

```
class CurAcct extends Account {  
    public CurAcct(String customerName, long accountNumber, double balance, double  
    overdraftLimit) {  
        super(customerName, accountNumber, "Current Account", balance);  
    }  
}
```

```
@Override
```

```
public void checkMinimumBalance() {  
    if (balance < 0) {  
        double penalty = Math.abs(balance) * 0.1;  
    }  
}
```

```
        balance -= penalty;

        System.out.println("Minimum balance not maintained. Penalty of $" + penalty + "
imposed.");
    }
}
}
```

```
class SavAcct extends Account {

    public SavAcct(String customerName, long accountNumber, double balance, double
interestRate) {

        super(customerName, accountNumber, "Savings Account", balance);
    }

    @Override
    public void checkMinimumBalance() {

        if (balance < 100) {

            double penalty = 20;

            balance -= penalty;

            System.out.println("Minimum balance not maintained. Penalty of $" + penalty + "
imposed.");
        }
    }
}
```

```
public class AccountTest {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        CurAcct currentAccount = new CurAcct("John Doe", 123456789, 5000, 1000);

        System.out.println("Current Account Details:");

        currentAccount.displayBalance();
    }
}
```

```
currentAccount.deposit(2000);
currentAccount.displayBalance();
currentAccount.depositInterest(2);
currentAccount.displayBalance();
currentAccount.withdraw(3000);
currentAccount.displayBalance();
currentAccount.checkMinimumBalance();
currentAccount.displayBalance();
```

```
SavAcct savingsAccount = new SavAcct("Jane Doe", 987654321, 1500, 3);
System.out.println("\nSavings Account Details:");
savingsAccount.displayBalance();
savingsAccount.deposit(500);
savingsAccount.displayBalance();
savingsAccount.depositInterest(2.5);
savingsAccount.displayBalance();
savingsAccount.withdraw(1000);
savingsAccount.displayBalance();
savingsAccount.checkMinimumBalance();
savingsAccount.displayBalance();
```

```
scanner.close();
System.out.println("Adithya Ravikeerthi 1BM22CS020");
}
```

```
}
```

OUTPUT:


```
adithya@adi82:~/adi$ javac AccountTest.java
adithya@adi82:~/adi$ java AccountTest
Current Account Details:
Account Balance: $5000.0
Deposit of $2000.0 successful.
Account Balance: $7000.0
Interest of $140.0 deposited.
Account Balance: $7140.0
Withdrawal of $3000.0 successful.
Account Balance: $4140.0
Account Balance: $4140.0

Savings Account Details:
Account Balance: $1500.0
Deposit of $500.0 successful.
Account Balance: $2000.0
Interest of $50.0 deposited.
Account Balance: $2050.0
Withdrawal of $1000.0 successful.
Account Balance: $1050.0
Account Balance: $1050.0
Adithya Ravikeerthi 1BM22CS020
```

LAB-6

/*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 com.example.cie;
```

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

```
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 the details:");
```

```
    usn = scan.nextLine();
```

```
    name = scan.nextLine();
```

```
    sem = scan.nextInt();
```

```
}
```

```
}
```

```
package com.example.cie;
```

```
public class Internal {
```

```
    public int im[] = new int[5];
```

```
}
```

```
package com.example.see;
```

```
import com.example.cie.Student;
```

```
public class External extends Student {
```

```
    public int sm[] = new int[5];
```

```
}
```

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

```
import com.example.see.External;
```

```
import com.example.cie.Internal;
```

```

public class FinalMarks {
    public static void main(String args[]) {
        int fm[] = new int[5];
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter n: ");
        int n = sc.nextInt();
        External st[] = new External[n];
        Internal s[] = new Internal[n];
        for (int i = 0; i < n; i++) {
            st[i] = new External();
            s[i] = new 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]);
            }
        }
        System.out.println("Adithya Ravikeerthi 1BM22CS020");
    }
}

```

OUTPUT:

```
adithya@adi82:~/java$ javac FinalMarks.java
```

```
^[[Aadithya@adi82:~/java$ java FinalMarks
```

```
Enter n:
```

```
1
```

```
Enter details 1
```

```
Enter the details:
```

```
Adithya
```

```
20
```

```
3
```

```
Enter im and sm of sub 1
```

```
50
```

```
50
```

```
Enter im and sm of sub 2
```

```
5050
```

```
50
```

```
Enter im and sm of sub 3
```

```
50
```

```
50
```

```
Enter im and sm of sub 4
```

```
50
```

```
50
```

```
Enter im and sm of sub 5
```

```
50
```

```
50
```

```
Final marks of 20
```

```
Course 1 = 100
```

```
Course 2 = 5100
```

```
Course 3 = 100
```

```
Course 4 = 100
```

```
Course 5 = 100
```

```
Adithya Ravikeerthi 1BM22CS020
```

LAB-7

/*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.*/

```
class WrongAge extends Exception {  
    public WrongAge() {  
        super("Age cannot be negative");  
    }  
}
```

```
class Input extends Exception {  
    public Input() {  
        super("Wrong input");  
    }  
}
```

```
class Father {  
    public int age;  
  
    Father(int age) throws WrongAge {  
        if (age < 0) {  
            throw new WrongAge();  
        }  
        this.age = age;  
    }  
}
```

```
}  
}
```

```
class Son extends Father {
```

```
    int s_age;
```

```
    Son(int f_age, int s_age) throws WrongAge, Input {
```

```
        super(f_age);
```

```
        if (s_age < 0) {
```

```
            throw new WrongAge();
```

```
        } else if (f_age <= s_age) {
```

```
            throw new Input();
```

```
        }
```

```
        this.s_age = s_age;
```

```
    }
```

```
}
```

```
public class J {
```

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

```
        try {
```

```
            Father f = new Father(40);
```

```
            System.out.println("Father's age: " + f.age);
```

```
            Son s = new Son(40, 56);
```

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

```
        } catch (WrongAge e) {
```

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

```
        } catch (Input ae) {
```

```
            System.out.println(ae.toString());
```

```
            System.out.println("Adithya Ravikeerthi 1BM22CS020");
```

```
    }  
    }  
}
```

OUTPUT:

```
adithya@adi82:~/java$ javac J.java  
adithya@adi82:~/java$ java J  
Father's age: 40  
Input: Wrong input  
Adithya Ravikeerthi 1BM22CS020  
adithya@adi82:~/java$ javac J.java  
adithya@adi82:~/java$ java J  
Father's age: 40  
Son's age: 30  
adithya@adi82:~/java$ javac J.java  
adithya@adi82:~/java$ java J  
Father's age: 40  
WrongAge: Age cannot be negative
```

LAB-8

/*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 A implements Runnable {  
    public void run() {  
        for (int i = 0; i < 5; i++) {  
            System.out.println("BMS COLLEGE OF ENGINEERING");  
            try {  
                Thread.sleep(10000);  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```



```
class B implements Runnable {  
    public void run() {  
        for (int i = 0; i < 5; i++) {  
            System.out.println("CSE");  
            try {  
                Thread.sleep(2000);  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        A a1 = new A();  
        B b2 = new B();  
        Thread t1 = new Thread(a1);  
        Thread t2 = new Thread(b2);  
        t1.start();  
        t2.start();  
        System.out.println("Adithya Ravikeerthi 1BM22CS020");  
    }  
}
```

OUTPUT:

```
adithya@adi82:~/java$ javac Main.java
```

```
adithya@adi82:~/java$ java Main
```

```
BMS COLLEGE OF ENGINEERING
```

```
Adithya Ravikeerthi 1BM22CS020
```

```
CSE
```

```
CSE
```

```
CSE
```

```
CSE
```

```
CSE
```

```
BMS COLLEGE OF ENGINEERING
```

```
BMS COLLEGE OF ENGINEERING
```

```
BMS COLLEGE OF ENGINEERING
```

```
BMS COLLEGE OF ENGINEERING
```