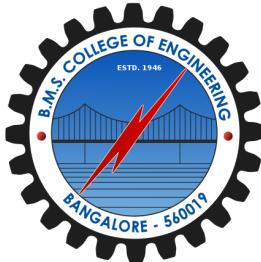


B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



LAB REPORT

23CS3PCOOJ

Submitted in partial fulfillment of the requirements
for
Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

PRIYANSHU KUMAR

1BM22CS210

Department of Computer Science and Engineering,
B.M.S College of Engineering,
Bull Temple Road, Basavanagudi, Bangalore, 560 019

2023-2024

12/12/23

12-12-23

classmate

Date _____
Page _____

Q. WAP to find the roots of a quadratic equation.

→ import java.util.*;

class QuadraticEquation
{

 public static void main (String args [])
 {

 Scanner s = new Scanner (System.in);

 double a, b, c;

 double D;

 double r1, r2;

 System.out.print ("Enter a, b & c: ");

 a = s.nextDouble();

 b = s.nextDouble();

 c = s.nextDouble();

$$D = b * b - 4 * a * c,$$

 if (D >= 0)
 {

$$r1 = (-b + \text{Math.sqrt}(D)) / (2 * a);$$

$$r2 = (-b - \text{Math.sqrt}(D)) / (2 * a);$$

 System.out.printf ("r1 = %.3f\n", r1);

 System.out.printf ("r2 = %.3f\n", r2);
 }

else

{

$$r_1 = -b / (2 * a);$$

$$r_2 = \text{Math.abs}(\text{Math.sqrt}(-D) / (2 * a));$$

System.out.printf("r₁ = %.3f + %.3f
i(n", r₁, r₂);

System.out.printf("r₂ = %.3f - %.3f
i(n", r₁, r₂);

3

{

}

Output:

Case 1: Enter a, b, & c:

$$r_1 = -1.000 + 1.732i$$

$$r_2 = -1.000 - 1.732i$$

Case 2: Enter a, b & c: 1 3 1

$$r_1 = -0.382$$

$$r_2 = -2.618$$

12/12/23

12-12-23

classmate

Date _____

Page _____

Q. WAP to find the roots of a quadratic equation.

→ import java.util.*;

class QuadraticEquation

{ public static void main (String args [])

Scanner s = new Scanner (System.in);

double a, b, c;

double D;

double r1, r2;

System.out.print ("Enter a, b & c: ");

a = s.nextDouble();

b = s.nextDouble();

c = s.nextDouble();

$$D = b * b - 4 * a * c,$$

if ($D \geq 0$)

$$r1 = (-b + \text{Math.sqrt}(D)) / (2 * a);$$

$$r2 = (-b - \text{Math.sqrt}(D)) / (2 * a);$$

System.out.printf ("r1 = %.3f\n", r1);

} System.out.printf ("r2 = %.3f\n", r2);

else

{

$$r_1 = -b / (2 * a);$$

$$r_2 = \text{Math.abs}(\text{Math.sqrt}(-D) / (2 * a));$$

System.out.printf("r₁ = %.3f + %.3f
i(n", r₁, r₂);

System.out.printf("r₂ = %.3f - %.3f
i(n", r₁, r₂);

3

{

}

Output:

Case 1: Enter a, b, & c:

$$r_1 = -1.000 + 1.732i$$

$$r_2 = -1.000 - 1.732i$$

Case 2: Enter a, b & c: 1 3 1

$$r_1 = -0.382$$

$$r_2 = -2.618$$

Name → Priyanshu Kumar
USN → IBM22CS210

classmate

Date _____
Page _____

LAB 2

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

```
class Subject {  
    int getSubjectMarks;  
    int credits;  
    int grade;  
}
```

```
class Student {  
    Scanner s = new Scanner(System.in);  
    Subject subject[] = new Subject[8];
```

```
String name, ron;  
double sgpa;
```

```
Student ()
```

```
{
```

```
for (int i=0; i<8; i++)
```

```
{
```

```
    subject[i] = new Subject();
```

```
}
```

```
void getStudentDetails ()
```

```
{
```

```
System.out.print("Enter student's name:  
name = s.nextLine();")
```

System.out.print("Enter student usn :");
usn = s.nextLine();

{

void getMarks()

{

{ for (int i=0; i<8; i++)

System.out.println("Subject" + (i+1));

System.out.print("Enter marks :");

subject[i].subjectMarks = s.nextInt();

System.out.print("Enter credits :");

subject[i].credits = s.nextInt();

int t = subject[i].subject / 10;

if (t < 4) {

t = 0;

{

t = t + 1;

if (t > 10) {

t = 10;

{

subject[i].credits = t;

{

void computeSGPA()
{
 int tg = 0;
 int tc = 0;
 for(int i=0; i<8; i++)
 {
 tc += subject[i].credits;
 tg += subject[i].grade *
 subject[i].credits;
 }
}

sgpa = tg / (tc + 1.0);
System.out.println("Your SGPA is " + sgpa);
}

public class StudentDemo {

 public static void main(String args[]){

 Student s1 = new Student();

 s1.getStudentDetails();

 s1.getMarks();

 s1.computeSGPA();

Output:

Enter student's name: Priyanshu
Enter student's usn: 18M22CS210

Subject 1

Enter marks: 81

Enter credits: 4

Subject 2:

Enter marks: 77

Enter credits: 3

Subject 3:

Enter marks: 62

Enter credits: 1

Subject 4:

Enter marks: 89

Enter credits: 4

.

.

.

Subject 8:

Enter marks: 100

Enter credits: 3

Your SGPA is 8.25.

26-12-03

Name → Priyanshu Kumar
USN → IBM22CS21D

classmate

Date _____

Page _____

LAB 3

→ import java.util.Scanner;

class Book

{

```
String name;  
String author;  
int price;  
int numPages;
```

Books (String name, String author, int
price, int numPages)

{

```
this.name = name;  
this.author = author;  
this.price = price;  
this.numPages = numPages;
```

}

public String toString()

{

String v = " ";

v += "Book name:" + this.name + "\n";
v += "Author name:" + this.author + "\n";
v += "Price :" + this.price + "\n";
v += "Number of pages:" + this.numPages
+ "\n";

return v;

}

public class Main

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

wint n;

Books b []);

String name, author;

unit price, numPages,

```
Scanner s = new Scanner(System.in);
System.out.print("Enter the number
of books:");
```

$\gamma = s.\text{nextInt}();$

`b = new Books[n];`

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

System.out.println("E in
S1 with i?");

Enter "Details of Book" + (i+1) + "m");

```
System.out.print("Name:");
```

`Name = s.next();`

```
System.out.print("Author:");  
author = s.next();
```

System.out.print("P₂₁₅₀ : ");

price = s.nextInt();

[1] System.out.print("Number of pages:");
numPages = s.nextInt();

b[i] = new Books(name, author,
price, numPages);
{

System.out.println("In displaying
book details: ");

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

~~System.out.println(b[i].toString());~~

{

}

{

Output :

Enter the number of books: 2

Enter details of Book 1

Name: Shyam

Author: Summer

Price: 100

Number of pages: 120

Enter details of Book 2:

Name : Peace

Author : Aryan

Price 20

Number of pages: 55

Displaying book details:

Book name: Shyam

Author name: Summer

Price: 100

Number of pages: 120

Book name: ~~Peace~~

Author name: Aryan

Price: 20

Number of pages: 55

8

26/12/23

Name: Priyanshu Kumar
USN: IBM22CS210

classmate

Date _____

Page _____

LAB 4

```
import java.util.Scanner;  
import java.util.Math;  
  
class InputScanner  
{  
    Scanner s;  
    InputScanner() {  
        s = new Scanner(System.in);  
    }
```

```
    public int takeInput(String m) {  
        System.out.println(m);  
        return s.nextInt();  
    }  
}
```

```
abstract class Shape  
{
```

```
    int dim1, dim2;
```

```
    Shape(int dim1, int dim2) {  
        this.dim1 = dim1;  
        this.dim2 = dim2;  
    }
```

```
    public abstract void printArea();
```

```
}
```

U8A1

class Rectangle extends Shape
{
 Rectangle (int length, int breadth)
 super (length, breadth);
}

public void printArea () {
 System.out.println ("Area of
rectangle = " + 1.0 *
 dim1 * dim2);
}

class Triangle extends Shape
{

Triangle (int base, int height) {
 super (base, height);
}

public void printArea () {
 System.out.println ("Area of triangle
= " + 0.5 * dim1 * dim2);
}

{

class Circle extends Shape

{ Circle(int radius) {

super(radius, radius); }

public void printArea () {

System.out.println ("Area of circle = "
+ 3.1415 * dim1* dim2);

{ }

public class ShapeDemo

{ public static void main (String args[])

InputScanner ic = new InputScanner();

Shape shape1, shape2, shape3;

int d1, d2;

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

d1 = ic.takeInput ("Enter length : ");

d2 = ic.takeInput ("Enter breadth : ");

shape1 = new Rectangle (d1, d2);

shape1.printArea ();

System.out.println ("Square : ");

d1 = ic.takeInput ("Enter side : ");

shape2 = new Square (d1, d1);

shape2.printArea ();

```
System.out.println("Side :");
d1 = i.c.takeInput("Enter radius :");
Shape3 = new Circle(d1, d1);
shape3.printArea();
```

{

3

~~Output:~~~~Rectangle~~~~Enter length : 2~~~~Enter breadth : 3~~~~Area of rectangle = 6.0~~~~Q.S.~~

LAB 5

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

```
abstract class Account {
```

```
    String customerName;
```

```
    long accountNumber;
```

```
    String accountType;
```

```
    double balance;
```

```
    boolean updated = false;
```

```
    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) {
```

```
        updated = false;
```

```
        balance += amount;
```

```
        System.out.println ("Deposit of ₹ " +
```

```
                           amount + " successful.");
```

```
}
```

~~public void withdraw (double amount) {~~ ~~update = false;~~ ~~if (amount <= balance) {~~ ~~balance -= amount;~~ ~~System.out.println ("Withdrawal~~~~of ₹ " + amount + " successful.");~~

```
    } else {  
        System.out.println ("Insufficient  
funds: withdrawal not allowed");  
    }  
}
```

```
public void displayBalance() {  
    update();  
    System.out.println ("Account Balance  
is " + balance);  
}
```

```
public abstract void update();
```

```
class Current extends Account {  
    double minBalance;  
    double serviceCharge;
```

```
    public Current (String customersName,  
        long accountNumber, double balance)  
    {  
        this.minBalance = 500;  
        this.serviceCharge = 10;  
    }
```

```
    public void update() {  
        if (!update) {  
            if (balance < minBalance) {  
                balance -= serviceCharge;  
                System.out.println ("Service  
charge of " + serviceCharge +  
                    " imposed for falling below  
minimum balance.");  
                updated = true;  
            }  
        }  
    }
```

```
        }  
    }  
    System.out.println("Account updated.");  
}
```

```
class SavAcct extends Account {  
    double interestRate;  
    double time;
```

```
    public SavAcct (String customerName,  
                    long accountNumber, double balance)  
    {  
        super (customerName, accountNumber,  
              "Savings", balance);  
        this.interestRate = 0.05;  
    }
```

```
    public void update() {  
        System.out.println ("Updated  
                           account");  
    }
```

```
    public void calculateCI (double r,  
                            int n, int t) {  
        balance = balance * Math.pow(1+r/n,  
                                     n*t);  
    }
```

public classes Bank Demo E

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

Cur Acct current Account = new
Cur Acct ("Priyanashu", 12
1234567 (89, 1000);

SavAcct savings Account = new
SavAcct ("Ajit", 987654321,
5000);

```
currentAccount.deposit(200);  
currentAccount.displayBalance();  
currentAccount.withdraw(200);  
currentAccount.displayBalance();
```

~~savings Account. deposit(1000);
savings Account. calculateCI (0.05, 2, 3);
savings Account. displayBalance();
savings Account. withdraw (850);
savings Account. displayBalance();~~

Orbit

Deposit of ₦ 200.0 successful.

Amount Balance: ₦ 1200.0

Withdrawal of ₦ 200.0 successful.

Account Balance: ₦ 1000.0

Deposit of ₦ 1000.0 successful.

Account Balance: ₦ 6958.160...

Withdrawal of ₦ 800.0 successful

Account Balance: ₦ 6158.160...

✓ 1
X 2
X 3
01.01.24

16/1/24

Name: Priyanshu Kumar
USN: 1BM22CS210

CLASSMATE

Date _____

Page _____

LAB - 6

Outputs

- String 1: Hello, World!
String 2: Ap

1.) Using String Literal

String 1: Hello, world!

Using new keyword

String 2: Apple

Using char array

String 3: Orange

Using byte array with character encoding

String 4: Hello

Substring of an existing string

String 5: Java

Using StringBuilder

String 6: Hello

234) Program 2

String 1: Hello

String 2: World

Length of String 1: 5

Concatenated String: Hello, World

Program 3

Using toString(): This is an example

Program 4
Extracted String: Bmsice

3.

5.) Bytes:

65

98

114

97

99

97

100

98

97

114

97

Character Array:

A

b

n

a

c

g

d

a

b

n

a

6. Bmsce equals Bmsce → true
Bmsce equals College → false
Bmsce equals BMSCe → false
Bmsce equals IgnoreCase BMSCe → true

7. Substring is not matched

8,
9,10) Program 8

Mango startsWith Man → true

Program 9

Orange endsWith range → true

Program 10

String 1: Apple

String 2: Apple

String 1.equals String 2 → true

String 1 == String 2 → false

11)

16.01.2021

Name: Priyanshu Kumar
USN: 1BM22CS210

23-1-21

classmate

Date _____

Page _____

LAB 7

CIE/Internals.java

package CIE;

import java.util.Scanner;

public class Internals extends Student {
protected int marks[] = new int[5];

public void inputCIEMarks () {

Scanner sc = new Scanner(System.in);

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

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

System.out.println("Enter marks
for Course" + (i+1) + ":");

marks[i] = sc.nextInt();

}

sc.close();

}

CIE/Student.java

package CIE;

import java.util.Scanner;

public class Student {

protected String usn = new String();

protected String name = new String();

protected int msem;

public void inputStudentsDetails() {

Scanner sc = new Scanner(System.in);

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

usn = sc.nextLine();

```
System.out.println("Enter your Name :");
name = sc.nextLine();
System.out.println("Enter your semesters");
sem = sc.nextInt();
sc.close();
```

```
public void displayStudentDetails() {
    System.out.println("USN : " + usn);
    System.out.println("Name : " + name);
    System.out.println("Sem : " + sem);
```

~~SEE/Externals.java~~

```
package SEE;
import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
    protected int marks[];
    protected int finalMarks[];

    public Externals() {
        marks = new int[5];
        finalMarks = new int[5];
    }
```

```
public void inputSEEMarks() {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter SEE marks");
```

```
for(int i=0; i<5; i++) {  
    System.out.println("Course " + (i+1) + ":" );  
    marks[i] = sc.nextInt();  
}  
sc.close();
```

```
public void calculateFinalMarks() {  
    for(int i=0; i<5; i++) {  
        finalMarks[i] = marks[i]/2 + super_marks  
    }  
}
```

```
public void displayFinalMarks() {  
    displayStudentDetails();  
    for(int i=0; i<5; i++) {  
        System.out.println("Course " + (i+1)  
            + ":" + finalMarks[i]);  
    }  
}
```

~~SEE Main.java~~
import SEE.Externals;

```
public class Main {  
    public static void main(String args[]) {  
        int numofStudents = 2;  
        Externals finalMarks[] = new  
        Externals[int numofStudent];  
  
        for(int i=0; i<numofStudents; i++) {
```

finalMarks[i] = new Externals();
finalMarks[i].inputStudentDetails();
System.out.println("Enter CIE marks");
finalMarks[i].inputCIEMarks();
finalMarks[i].inputSEEMarks();

System.out.println("Displaying data: ");

for (int i = 0; i < numStudents; i++)
finalMarks[i].calculateFinalMarks();
finalMarks[i].displayFinalMarks();

Output

Enter your USN : IBM22CS200

Enter your Name: Priyanshu Kumar

Enter your Semester: 2

Enter CIE Marks

Enter marks for Course 1: 45

Enter marks for Course 2: 37

Enter marks for Course 3: 47

:

8
25.07.2021

Name: Priyanshu Kumar
USN: 1BM22CS210

classmate

Date _____

Page _____

Lab-8

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

```
class WrongAge extends Exception {
```

```
    public WrongAge(String message) {  
        super(message);  
    }
```

```
}
```

```
extends InputScanner
```

```
class Father {
```

```
    private int fatherAge;
```

```
    public Father() throws WrongAge {  
        super();
```

P Scanner s = new Scanner(System.in);
System.out.print("Enter father's age:");
fatherAge = s.nextInt();

```
super.s.close();
```

```
if(fatherAge < 0)
```

```
throw new WrongAge("Age cannot  
be negative")
```

```
}
```

```
public void display() {
```

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

```
}
```

```
class Son extends Father {
```

```
private int sonAge;
```

```
public Son() throws WrongAge {  
    super();
```

super.s

```
Scanner s = new Scanner (System. in);  
System.out.print ("Enter son's age : ");
```

```
sonAge = super.s.nextInt();
```

```
super.s.close();
```

```
if (sonAge >= super.fatherAge)
```

```
throw new WrongAge ("Son's age  
cannot be greater than father's  
age");
```

```
else if (sonAge < 0)
```

```
throw new WrongAge ("Age cannot  
be negative);
```

```
public void display () {
```

```
super.display ();
```

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

```
public class Main {
```

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

```
try {
```

```
Son son = new Son ();
```

```
son.display ();
```

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

```
System.out.println ("Exception : " +  
e.getMessage());
```

class InputScanner {
Scanner s;

InputScanner () {

}

}

Output:

(Case 1) Enter son's age: father's age: -10

Exception: Age cannot be negative.

(Case 2) Enter father's age: 40

Enter son's age: 50

Exception: Son's age cannot be greater than father's age

8

30/01/24

~~6-24-24~~

Lab 9

Date _____
Page _____

→ class CollegeThread extends Thread {

@Override

```
public void run() {
```

```
    while (true) {
```

```
        System.out.println("BMS College  
Engineering");
```

```
    try {
```

```
        Thread.sleep(1000);
```

```
} catch (InterruptedException e) {
```

```
    e.printStackTrace();
```

```
}
```

```
}
```

class DepartmentThread extends Thread {

@Override

```
public void run() {
```

```
    while (true) {
```

```
        System.out.println("CSE");
```

```
    try {
```

```
        Thread.sleep(2000);
```

```
} catch (InterruptedException e) {
```

```
    e.printStackTrace();
```

```
}
```

```
}
```

```
}
```

public class Main {

public static void main(String[] args) {

CollegeThread collegeThread = new
collegeThread();

DepartmentThread departmentThread =
new DepartmentThread();

collegeThread.start();

{ departmentThread.start();

}

Output:

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

CSE

CSE

CSE

6/2/2021

Lab 10

classmate

Date _____

Page _____

(10)(a)

```
→ class Q {  
    int n;  
    boolean valueSet = false;  
  
    synchronized int get() {  
        while (!valueSet) {  
            try {  
                System.out.println("nConsumer  
Waiting\n");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("Interrupted-  
Exception caught");  
            }  
        }  
        System.out.println("Got : " + n);  
        valueSet = false;  
        System.out.println("nNotify Producer\n");  
        notifies;  
        return n;  
    }  
}
```

```
synchronized void put (int n) {  
    while (valueSet) {  
        try {  
            System.out.println("nProducer  
waiting\n");  
            wait();  
        } catch (InterruptedException e) {  
            System.out.println("Interrupted-  
Exception caught");  
        }  
    }  
}
```

```
synchronized void put (int n) {  
    while (valueSet) {  
        try {  
            System.out.println("nProducer  
waiting\n");  
            wait();  
        } catch (InterruptedException e) {  
            System.out.println("Interrupted-  
Exception caught");  
        }  
    }  
}
```

```
this.n = n;  
valueset = true;  
System.out.println("Put: " + n);  
System.out.println("In Notify consumer");  
notify();  
}  
}
```

```
class Producer implements Runnable {  
    Q q;
```

```
    Producer(Q q) {
```

```
        this.q = q;  
        new Thread(this, "Producer").start();  
    }
```

```
    public void run() {
```

```
        int i = 0;
```

```
        while (i < 15) {
```

```
            q.put(i++);  
        }
```

```
    }
```

~~```
class Consumer implements Runnable {
```~~~~```
    Q q;
```~~~~```
 Consumer(Q q) {
```~~~~```
        this.q = q;
```~~~~```
 new Thread(this, "Consumer").start();
 }
```~~

(19)

```
public void run() {
 int i = 0;
 while (i < 15) {
 int x = q.get();
 System.out.println("Consumed " + x);
 i++;
 }
}
```

```
public class PCFixed {
 public static void main(String args) {
 Q q = new Q();
 new Producer(q);
 new Consumer(q);
 System.out.println("Press Control-C to stop.");
 }
}
```

Output:

Put: 1

Get: 1

Put: 2

Get: 2

(10) (b)

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

→ public class DeadlockExample {

    public static void main (String [] args) {  
        Object lock1 = new Object ();

        Object lock2 = new Object (),

        Thread thread1 = new Thread () → {

            synchronized (lock1) {

                System.out.println ("Thread  
                    1 acquired lock1");

            try {

                Thread.sleep (100);

            } catch (InterruptedException e) {

                e.printStackTrace ();

        }

            synchronized (lock2) {

                System.out.println ("Thread 1  
                    acquired lock2");

    }

}

},

    Thread thread2 = new Thread () → {

        synchronized (lock2) {

            System.out.println ("Thread 2  
                    acquired lock2");

        try {

            Thread.sleep (100);

        } catch (InterruptedException e) {

                e.printStackTrace ();

synchronized (lock1) {

System.out.println("Thread

2 acquired lock2");

3;

thread1.start();

thread2.start();

3;

6

12

13

Output:

Thread 1 acquired lock1.

Thread 2 acquired lock2.

## Lab 11

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

9) 

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

`class swingDemo {`

`swingDemo() {`

`JFrame jfrm = new JFrame ("Divides App");`

`jfrm. setSize (275, 150);`

`jfrm. setLayout (new FlowLayout());`

`jfrm. setDefaultCloseOperation(JFrame.`

`EXIT_ON_CLOSE);`

`JLabel jlab = new JLabel ("Enter the`

`divisor and dividend:");`

`JTextField atjtf = new JTextField (8);`

`JTextField bjtff = new JTextField (8);`

`JButton button = new JButton ("Calculate");`

~~`JLabel err = new JLabel ();`~~

~~`JLabel alab = new JLabel ();`~~

~~`JLabel blab = new JLabel ();`~~

~~`JLabel anstab = new JLabel ();`~~

`jfrm. add (err);`

`jfrm. add (jlab);`

`jfrm. add (atjtf);`

`jfrm. add (bjtff);`

`jfrm. add (button);`

`jfrm. add (alab);`

11 det Page

```
jfrm.add(blab);
jfrm.add(anslab);

ActionListener l = new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
 System.out.println("Action event from a text field");
 }
};
```

```
ajtf.addActionListener(l);
bjtf.addActionListener(l);
```

```
button.addActionListener(new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
```

```
 try {
 int a = Integer.parseInt(ajtf.getText());
 int b = Integer.parseInt(bjtf.getText());
```

```
 int ans = a / b;
```

```
 alab.setText("mA = " + a);
```

```
 blab.setText("nB = " + b);
```

```
 anslab.setText("nAns = " + ans);
```

```
 err.setText("");
```

```
} catch (NumberFormatException e) {
```

```
 alab.setText(" ");
```

```
 blab.setText("");
```

```
 anslab.setText("");
```

```
 err.setText("Enter Only Integers!");
```

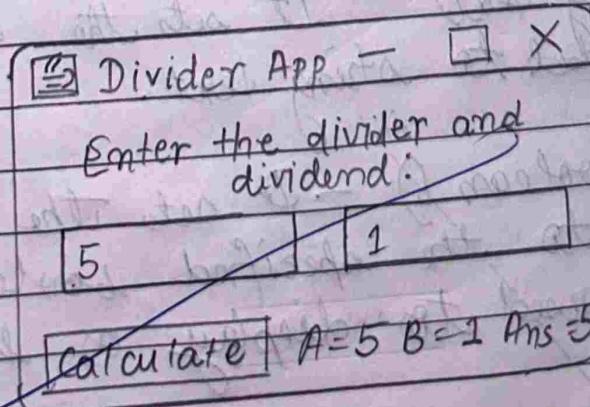
```
}
```

```
catch (ArithmeticException e) {
 alab.setText(" ");
 blab.setText(" ");
 anslab.setText(" ");
 err.setText("B should be non-zero!");
}

jform.setVisible(true);
```

```
public static void main (String args[])
{
 SwingUtilities.invokeLater (new Runnable()
 {
 public void run()
 {
 new SwingDemo();
 }
 });
}
```

Output:



Here is a report on the methods used in the program:

- 1) actionPerformed(ActionEvent evt) - An ActionListener interface method implemented to handle action events triggered by text fields and buttons. It contains logic to respond to user input events.
- 2) run() - A Runnable interface method implemented to execute code within the SwingUtilities.invokeLater() method. It instantiates a new SwingDemo object, allowing the GUI to be created & displayed.
- 3) addActionListener(ActionListener l) - A method of JTextField & JButton classes. It registers an ActionListener to receive action events when the user interacts with the associated GUI component.
- 4) setText(String text) - It sets the text content of the label to string text.
- 5) setVisible(boolean b) - It sets the visibility of the frame to the specified boolean value b. It is used to display or hide the GUI frame.
- 6) setLayout(LayoutManager manager) - A method of the Container class. It sets the layout

manages for the container. It determines how components are arranged with the container.

- 7) setDefaultCloseOperation (int operation) - A method of the JFrame class. It sets the default close operation of the frame. It determines what happens when user closes the window.

JSD  
JFS  
no. 2 m

## Lab 1

```
import java.util.Scanner;

class QuadraticEquation {

 public static void main(String[] args) {
 Scanner s = new Scanner(System.in);
 double a, b, c;
 double D;
 double r1, r2;

 System.out.print("Enter a, b, & c: ");
 a = s.nextDouble();
 b = s.nextDouble();
 c = s.nextDouble();

 D = b * b - 4 * a * c;

 if (D >= 0) {
 r1 = (-b + Math.sqrt(D)) / (2 * a);
 r2 = (-b - Math.sqrt(D)) / (2 * a);
 System.out.println("r1 = " + r1);
 System.out.println("r2 = " + r2);
 } else {
 r1 = -b / (2 * a);
 r2 = Math.sqrt(-D) / (2 * a);
 System.out.println("r1 = " + r1 + "+" + r2 + "i");
 System.out.println("r2 = " + r1 + "-" + r2 + "i");
 }
 }
}
```

## Lab 2

```
import java.util.Scanner;

class Subject {
 String subjectMarks;
 int credits;
 int grade;
}

class Student {
 Scanner s = new Scanner(System.in);
 Subject[] subject = new Subject[8];
 String name, usn;
 double sgpa;

 Student() {
 for (int i = 0; i < 8; i++) {
 subject[i] = new Subject();
 }
 }

 void getStudentDetail() {
 System.out.print("Enter student name: ");
 name = s.nextLine();
 System.out.print("Enter student usn: ");
 usn = s.nextLine();
 }

 void getMarks() {
 for (int i = 0; i < 8; i++) {
 System.out.print("Subject" + (i + 1));
 System.out.print("Enter marks: ");
 subject[i].marks = s.nextInt();
 }
 }
}
```

```

 System.out.print("Enter credits: ");
 subject[i].credit = s.nextInt();

 int t = subject[i].subject/10;
 if(t < 4)
 t = 0;
 t = t + 1;
 if(t > 10)
 t = 10;
 subject[i].credits = t;
 }

}

void computeSGPA() {
 int tg = 0;
 double tc = 0;
 for (int i = 0; i < 8; i++) {
 tc += subject[i].credits;
 tg += subjects[i].grade * subjects[i].credit;
 }
 sgpa = tg / (tc * 1.0);
 System.out.println("Your SGPA is: " + sgpa);
}

}

public class StudentDemo {
 public static void main(String[] args) {
 Student s1 = new Student();
 s1.getStudentDetail();
 s1.getMarks();
 s1.computeSGPA();
 }
}

```

## Lab 3

```
import java.util.Scanner;

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

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

 public String toString() {
 String v = "";
 v += "Book name: " + this.name + "\n" +
 "Author name: " + this.author + "\n" +
 "Price: " + this.price + "\n" +
 "Number of pages: " + this.numPages;
 return v;
 }
}

public class Main {
 public static void main(String[] args) {
 int n;
 Book[] b;
 String name, author;
 int price, numPages;
```

```
Scanner s = new Scanner(System.in);
System.out.print("Enter the number of books: ");
n = s.nextInt();
b = new Book[n];

for (int i = 0; i < n; i++) {
 System.out.println("Enter details of Book " + (i+1));
 System.out.print("Name: ");
 name = s.next();
 System.out.print("Author: ");
 author = s.next();
 System.out.print("Price: ");
 price = s.nextInt();
 System.out.print("Number of pages: ");
 numPages = s.nextInt();
 b[i] = new Book(name, author, price, numPages);
}

System.out.println("\nDisplaying book details: ");
for (int i = 0; i < n; i++) {
 System.out.println(b[i].toString());
}
}
```

## Lab 4

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

class InputScanner {
 Scanner s;
 InputScanner() {
 s = new Scanner(System.in);
 }
 public int takeInput(String m) {
 System.out.println(m);
 return s.nextInt();
 }
}

abstract class Shape {
 int dim1, dim2;

 Shape(int dim1, int dim2) {
 this.dim1 = dim1;
 this.dim2 = dim2;
 }

 public abstract void printArea();
}

class Rectangle extends Shape {
 Rectangle(int length, int breadth) {
 super(length, breadth);
 }

 public void printArea() {
```

```
 System.out.println("Area of rectangle = " + (1.0 * dim1 *
dim2));
 }
}

class Triangle extends Shape {
 Triangle(int base, int height) {
 super(base, height);
 }

 public void printArea() {
 System.out.println("Area of triangle = " + (0.5 * dim1 *
dim2));
 }
}

class Circle extends Shape {
 Circle(int radius) {
 super(radius, radius);
 }

 public void printArea() {
 System.out.println("Area of circle = " + (3.1415 * dim1 *
dim2));
 }
}

public class Main {
 public static void main(String[] args) {
 InputScanner ic = new InputScanner();
 Shape shape1, shape2, shape3;
 int d1, d2;

 System.out.println("Rectangle");
 d1 = ic.takeInput("Enter length:");
 d2 = ic.takeInput("Enter width:");
 shape1 = new Rectangle(d1, d2);
 shape1.printArea();
 }
}
```

```
d2 = ic.takeInput("Enter breadth:");
shape1 = new Rectangle(d1, d2);
shape1.printArea();

System.out.println("Triangle");
d1 = ic.takeInput("Enter base:");
d2 = ic.takeInput("Enter height:");
shape2 = new Triangle(d1, d2);
shape2.printArea();

System.out.println("Circle");
d1 = ic.takeInput("Enter radius:");
shape3 = new Circle(d1, d1);
shape3.printArea();
}

}
```

## Lab 5

```
import java.util.Scanner;

abstract class Account {
 String customerName;
 long accountNumber;
 String accountType;
 double balance;
 boolean updated = false;

 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) {
 updated = false;
 balance += amount;
 System.out.println("Deposit of ₹ " + amount + " successful.");
 }

 public void withdraw(double amount) {
 updated = false;
 if (amount <= balance) {
 balance -= amount;
 System.out.println("Withdrawal of ₹ " + amount + "
successful.");
 } else {
 System.out.println("Insufficient funds. Withdrawal not
allowed.");
 }
 }
}
```

```
 }

 }

public void displayBalance() {
 update();
 System.out.println("Account Balance: ₹ " + balance);
}

public abstract void update();

}

class CurAcct extends Account {
 double minBalance;
 double serviceCharge;

 public CurAcct(String customerName, long accountNumber, double
balance) {
 super(customerName, accountNumber, "Current", balance);
 this.minBalance = 500;
 this.serviceCharge = 10;
 }

 public void update() {
 if (!updated) {
 if (balance < minBalance) {
 balance -= serviceCharge;
 System.out.println("Service charge of ₹ " + serviceCharge
+ " imposed for falling below minimum balance.");
 }
 updated = true;
 }
 }
}
```

```
class SavAcct extends Account {
 double interestRate;

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

 public void update() {
 double interest = balance * interestRate;
 balance += interest;
 System.out.println("Interest of ₹ " + interest + " computed
and deposited. Updated balance: ₹ " + balance);
 }
}

public class BankDemo {
 public static void main(String[] args) {
 CurAcct currentAccount = new CurAcct("Priyanshu", 123456789,
1000);
 SavAcct savingsAccount = new SavAcct("Ajit", 987654321, 5000);

 currentAccount.deposit(200);
 currentAccount.displayBalance();
 currentAccount.withdraw(200);
 currentAccount.displayBalance();

 savingsAccount.deposit(1000);
 savingsAccount.displayBalance();
 savingsAccount.withdraw(800);
 savingsAccount.displayBalance();
 }
}
```

## Lab 6

```
public class Program1 {
 public static void main(String[] args) {

 System.out.println("Using String Literal");
 String str1 = "Hello, World!";
 System.out.println("String 1: " + str1);

 System.out.println("Using new keyword");
 String str2 = new String("Apple");
 System.out.println("String 2: " + str2);

 System.out.println("Using char array");
 char[] charArray = {'O', 'r', 'a', 'n', 'g', 'e'};
 String str3 = new String(charArray);
 System.out.println("String 3: " + str3);

 System.out.println("Using byte array with character
encoding");
 byte[] byteArray = {72, 101, 108, 108, 111}; // ASCII values
for "Hello"
 String str4 = new String(byteArray);
 System.out.println("String 4: " + str4);

 System.out.println("Substring of an existing string");
 String originalStr = "Java Programming";
 String str5 = new String(originalStr.substring(0, 4)); //
Extract "Java"
 System.out.println("String 5: " + str5);

 System.out.println("Using StringBuilder");
 StringBuilder stringBuilder = new StringBuilder("Hello");
 String str6 = new String(stringBuilder.toString());
```

```
 System.out.println("String 6: " + str6);

 }

}

public class Program234 {
 public static void main(String[] args) {

 String str1 = "Hello";
 String str2 = "World";
 int length = str1.length();

 System.out.println("Program 2");
 System.out.println("String 1: " + str1);
 System.out.println("String 2: " + str2);
 System.out.println("Length of String 1: " + length);
 String concatenatedString = str1.concat(", " + str2);
 System.out.println("Concatenated String: " +
concatenatedString);

 System.out.println("Program 3");
 String toStringExample = "This is an example.";
 System.out.println("Using toString(): " + toStringExample);

 System.out.println("Program 4");
 String originalString = "Welcome to Bmsce college";
 char[] targetArray = new char[5];
 originalString.getChars(11, 16, targetArray, 0);
 String extractedString = new String(targetArray);
 System.out.println("Extracted String: " + extractedString);

 }
}
```

```
public class Program5 {
 public static void main(String args[])
 {
 String str = "Abracadbara";
 byte[] ba = str.getBytes();
 char[] ca = str.toCharArray();

 System.out.println("Bytes:");
 for (int i = 0; i < ba.length; i++) {
 System.out.println(ba[i]);
 }

 System.out.println("\nCharacter Array:");
 for (int i = 0; i < ca.length; i++) {
 System.out.println(ca[i]);
 }
 }
}

public class Program6 {
 public static void main(String[] args)
 {
 String str1 = "Bmsce";
 String str2 = "College";
 String str3 = "BMSCE";
 System.out.println("Bmsce equals Bmsce -> " +
str1.equals(str1));
 System.out.println("Bmsce equals College -> " +
str1.equals(str2));
 System.out.println("Bmsce equals BMSCE -> " +
str1.equals(str3));
 System.out.println("Bmsce equalsIgnoreCase BMSCE -> " +
str1.equalsIgnoreCase(str3));
 }
}
```

```

}

public class Program7 {
 public static void main(String[] args) {
 System.out.println("Program 7");
 String str1 = "Welcome to Bmsce College of Engineering";
 String str2 = "Bmsce college";

 if (str1.regionMatches(true, 11, str1, 0, str2.length()))
 System.out.println("Substring is matched");
 } else {
 System.out.println("Substring is not matched");
 }
 }
}

public class Program8910 {
 public static void main(String[] args) {
 System.out.println("Program 8");
 System.out.println("Mango startsWith Man -> " +
 "Mango".startsWith("Man"));
 System.out.println("Program 9");
 System.out.println("Orange endsWith range -> " +
 "Orange".startsWith("range"));
 System.out.println("Program 10");
 String str1 = "Apple";
 String str2 = new String("Apple");
 System.out.println("String 1: " + str1);
 System.out.println("String 2: " + str2);
 System.out.println("String 1 equals String 2 -> " +
 str1.equals(str2));
 System.out.println("String 1 == String 2 -> " + (str1 == str2));
 }
}

```

```
}
```

## Lab 7

```
// CIE/Internals.java
```

```
package CIE;
```

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

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

```
 protected int marks[] = new int[5];
```

```
 public void inputCIEMarks() {
```

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

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

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

```
 System.out.println("Enter marks for Course: " + (i+1) +
```

```
":");
```

```
 marks[i] = sc.nextInt();
```

```
}
```

```
 sc.close();
```

```
}
```

```
}
```

```
// CIE/Student.java
```

```
package CIE;
```

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

```
public class Student {
```

```
 protected String usn = new String();
```

```
 protected String name = new String();
```

```
 protected int sem;
```

```
 public void inputStudentDetails() {
```

```

Scanner sc = new Scanner(System.in);
System.out.print("Enter your usn: ");
usn = sc.nextLine();
System.out.print("Enter your name: ");
name = sc.nextLine();
System.out.print("Enter your semester: ");
sem = sc.nextInt();
sc.close();
}

public void displayStudentDetails() {
 System.out.println("USN: " + usn);
 System.out.println("Name: " + name);
 System.out.println("Semester: " + sem);
}

```

```

// SEE/Externals.java
package SEE;

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

public class Externals extends Internals {
 protected int marks[];
 protected int finalMarks[];

 public Externals() {
 marks = new int[5];
 finalMarks = new int[5];
 }

 public void inputSEEMarks() {
 Scanner sc = new Scanner(System.in);

```

```

 System.out.println("Enter SEE marks:");
 for (int i = 0; i < 5; i++) {
 System.out.println("Course " + (i + 1) + ":");

 marks[i] = sc.nextInt();
 }

 sc.close();
 }

 public void calculateFinalMarks() {
 for (int i = 0; i < 5; i++) {
 finalMarks[i] = (marks[i] / 2) + super.marks[i];
 }
 }

 public void displayFinalMarks() {
 displayStudentDetails();
 for (int i = 0; i < 5; i++) {
 System.out.println("Course " + (i + 1) + ":" +
finalMarks[i]);
 }
 }
}

// Main.java
import SEE.Externals;

public class Main {
 public static void main(String[] args) {
 int numOfStudents = 2;
 External[] finalMarks = new External[numOfStudents];

 for (int i = 0; i < numOfStudents; i++) {
 finalMarks[i] = new External();
 finalMarks[i].inputStudentDetails();
 finalMarks[i].inputCIEMarks();
 }
 }
}

```

```
 finalMarks[i].inputSEEMarks();
}

System.out.println("Displaying data:");
for (int i = 0; i < numStudents; i++) {
 finalMarks[i].calculateFinalMarks();
 finalMarks[i].displayFinalMarks();
}
}
}
```

## Lab 8

```
import java.util.Scanner;

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

class Father extends InputScanner {
 private int fatherAge;

 public Father() throws WrongAge {
 super();
 super.s = new Scanner(System.in);
 System.out.print("Enter father's age: ");
 fatherAge = super.s.nextInt();
 super.s.close();
 if (fatherAge < 0) {
 throw new WrongAge("Age cannot be negative");
 }
 }

 public void display() {
 System.out.println("Father's age: " + fatherAge);
 }
}

class Son extends Father {
 private int sonAge;

 public Son() throws WrongAge {
 super();
 }
}
```

```

super.s = new Scanner(System.in);
System.out.print("Enter son's age: ");
sonAge = super.s.nextInt();
super.s.close();
if (sonAge >= super.fatherAge) {
 throw new WrongAge("Son's age cannot be greater than or
equal to father's age");
} else if (sonAge < 0) {
 throw new WrongAge("Age cannot be negative");
}
}

public void display() {
super.display();
System.out.println("Son's age: " + sonAge);
}
}

public class Main {
public static void main(String[] args) {
try {
Son son = new Son();
son.display();
} catch (WrongAge e) {
System.out.println("Exception: " + e.getMessage());
}
}
}

class InputScanner {
Scanner s;
public InputScanner() {
}
}

```

## Lab 9

```
class CollegeThread extends Thread {
 @Override
 public void run() {
 while (true) {
 System.out.println("BMS College of Engineering");
 try {
 Thread.sleep(10000);
 } catch (InterruptedException e) {
 e.printStackTrace();
 } } }
 }

class DepartmentThread extends Thread {
 @Override
 public void run() {
 while (true) {
 System.out.println("CSE");
 try {
 Thread.sleep(2000);
 } catch (InterruptedException e) {
 e.printStackTrace();
 } } }
 }

public class Main {
 public static void main(String[] args) {
 CollegeThread collegeThread = new CollegeThread();
 DepartmentThread departmentThread = new DepartmentThread();
 collegeThread.start();
 departmentThread.start();
 }
}
```

## Lab 10

```
// Inter-Process Communication

class Q {
 private int n;
 private boolean valueSet = false;

 synchronized int get() {
 while (!valueSet) {
 try {
 System.out.println("\nConsumer waiting\n");
 wait();
 } catch (InterruptedException e) {
 System.out.println("InterruptedException caught");
 }
 }

 System.out.println("Got: " + n);
 valueSet = false;
 System.out.println("\nNotify Producer\n");
 notify();
 return n;
 }

 synchronized void put(int n) {
 while (valueSet) {
 try {
 System.out.println("\nProducer waiting\n");
 wait();
 } catch (InterruptedException e) {
 System.out.println("InterruptedException caught");
 }
 }
 }
}
```

```
 this.n = n;
 valueSet = true;
 System.out.println("Put: " + n);
 System.out.println("\nNotify Consumer\n");
 notify();
}
}

class Producer implements Runnable {
 private Q q;

 Producer(Q q) {
 this.q = q;
 new Thread(this, "Producer").start();
 }

 public void run() {
 int i = 0;
 while (i < 15) {
 q.put(i++);
 }
 }
}

class Consumer implements Runnable {
 private Q q;

 Consumer(Q q) {
 this.q = q;
 new Thread(this, "Consumer").start();
 }

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

```

 while (i < 15) {
 int r = q.get();
 System.out.println("Consumed: " + r);
 i++;
 }
 }

}

public class PC {
 public static void main(String args[]) {
 Q q = new Q();
 new Producer(q);
 new Consumer(q);
 System.out.println("Press Control-C to stop.");
 }
}

// Deadlock
public class DeadlockExample {
 public static void main(String[] args) {
 Object lock1 = new Object();
 Object lock2 = new Object();

 Thread thread1 = new Thread(() -> {
 synchronized (lock1) {
 System.out.println("Thread 1 acquired lock1");

 try {
 Thread.sleep(100);
 } catch (InterruptedException e) {
 e.printStackTrace();
 }
 }

 synchronized (lock2) {
 System.out.println("Thread 1 acquired lock2");
 }
 });
 }
}

```

```
 }

 }

}) ;

Thread thread2 = new Thread(() -> {
 synchronized (lock2) {
 System.out.println("Thread 2 acquired lock2");

 try {
 Thread.sleep(100);
 } catch (InterruptedException e) {
 e.printStackTrace();
 }
 }

 synchronized (lock1) {
 System.out.println("Thread 2 acquired lock1");
 }
}

thread1.start();
thread2.start();
}
}
```

## Lab 11

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

class SwingDemo{
 SwingDemo () {
 JFrame jfrm = new JFrame("Divider App");
 jfrm.setSize(275, 150);
 jfrm.setLayout(new FlowLayout());
 jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

 JLabel jlab = new JLabel("Enter the divider and dividend:");

 JTextField ajtf = new JTextField(8);
 JTextField bjtf = new JTextField(8);

 JButton button = new JButton("Calculate");

 JLabel err = new JLabel();
 JLabel alab = new JLabel();
 JLabel blab = new JLabel();
 JLabel anslab = new JLabel();

 jfrm.add(err);
 jfrm.add(jlab);
 jfrm.add(ajtf);
 jfrm.add(bjtf);
 jfrm.add(button);
 jfrm.add(alab);
 jfrm.add(blab);
 jfrm.add(anslab);
```

```

ActionListener l = new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
 System.out.println("Action event from a text field");
 }
};

ajtf.addActionListener(l);
bjtf.addActionListener(l);

button.addActionListener(new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
 try{
 int a = Integer.parseInt(ajtf.getText());
 int b = Integer.parseInt(bjtf.getText());
 int ans = a/b;

 alab.setText("\nA = " + a);
 blab.setText("\nB = " + b);
 anslab.setText("\nAns = "+ ans);
 }
 catch(NumberFormatException e){
 alab.setText("");
 blab.setText("");
 anslab.setText("");
 err.setText("Enter Only Integers!");
 }
 catch(ArithmaticException e){
 alab.setText("");
 blab.setText("");
 anslab.setText("");
 err.setText("B should be NON zero!");
 }
 }
});

jfrm.setVisible(true);
}
);

```

```
}

public static void main(String args[]) {
 SwingUtilities.invokeLater(new Runnable() {
 public void run() {
 new SwingDemo();
 }
 });
}
```