

B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



LAB

REPORT

23CS3PCOOJ

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

Submitted by:
PRATHAM GANAPATHY
(1BM22CS206)

Department of Computer Science and Engineering,
B.M.S College of Engineering,
Bull Temple Road, Basavanagudi, Bangalore, 560
019 2023-2024.

INDEX

Sl. No.	Title	Date
1	Complete scanned observation Book	12/12/2023 - 20/02/2024
2	Lab 1	12/12/2023
3	Lab 2	19/12/2023
4	Lab 3	26/12/2023
5	Lab 4	02/01/2024
6	Lab 5	09/01/2024
7	Lab 6	16/01/2024
8	Lab 7	23/01/2024
9	Lab 8	30/01/2024
10	Lab 9	06/02/2024
11	Lab 10	20/02/2024

1 Quadratic code

```
import java.util.*;
```

```
public class quadratic
```

```
{
```

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

```
{
```

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

```
System.out.println ("Pratham Gangpathy IBM 22CS 206");
```

```
int a, b, c;
```

```
double r1, r2, d;
```

```
System.out.println ("Enter coefficients of a, b, c");
```

```
a = SC.nextInt();
```

```
b = SC.nextInt();
```

```
c = SC.nextInt();
```

```
while (a == 0)
```

```
{
```

```
System.out.println ("Not a quadratic equation");
```

```
System.out.println ("Enter a non zero value for a");
```

```
a = SC.nextInt();
```

```
}
```

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

```
if (d == 0)
```

```
{
```

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

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

```
System.out.println (" Root1 = Root2 = " + r1);
```

```
}
```

else if (d > 0)
}

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

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

System.out.println("Roots are real and distinct");
 System.out.println("Root1 = "+r1+" Root2 = "+r2);

}
 else if (d < 0)
{

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

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

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

System.out.println("Root1 = "+r1+"i "+r2+" Root2 = "+r1+"i "+r2);

}

}

Prathom Ganapathy

IBM22CS206

Enter coefficients of a, b, c

4 5 6

Roots are ~~tiny~~ imaginary

$$\text{Root1} = 0.0 + i 0.532687216470449$$

$$\text{Root2} = 0.0 - i 0.532687216470449$$

Prathom Ganapathy

IBM22CS206

Enter coefficients of a, b, c

1 -2 1

Roots are equal real and equal

$$\text{Root1} = \text{Root2} = 1.0$$

Prathom . Ganapathy

IBM22CS206

Enter coefficients of a, b, c

1 -3 2

Roots are real and distinct

$$\text{Root1} = 2.0 \quad \text{Root2} = 1.0$$

Lab Program 2

```
import java.util.*;
```

```
public class Subject {
    int Subjectmarks, credit, grade;
}
```

```
class Student
```

```
{
```

```
Subject subject[];
```

```
String name, usn;
```

```
double sgpa;
```

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

```
int n;
```

```
Student()
```

```
{
```

```
Subject = new Subject[10];
```

```
String name, usn;
```

```
double sgpa;
```

```
SC. System.out.println ("Enter no. of subject");
```

```
n = SC.nextInt();
```

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

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

```
SC.nextLine();
```

```
}
```

```
void has getStudentDetails()
```

```
System.out.println ("Enter name and USN");
```

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

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

```
}
```

{ void getMarks()

 System.out.println("\n");
 for (int i = 0; i < n; i++)

 System.out.println("Enter no. of credits: ");
 subject[i].credits = SC.nextInt();
 System.out.println("Enter marks obtained: ");
 subject[i].SubjectMarks = SC.nextInt();
 if (subject[i].SubjectMarks >= 90)
 subject[i].SubjectMarks = 100;
 else if

 int k = subject[i].SubjectMarks;

 if (k > 100),

 k = 100;

 else if (k < 40)

 k = 0;

 else if (k

 Student[i].grade = 0; }

 else if (k >= 40 && k <= 49)

}

 Student[i].grade = 4;

 else if (k >= 50 && k <= 59)

}

 Student[i].grade = 5;

}

 else if (k >= 60 && k <= 69)

}

 Student[i].grade = 6;

}

double computeSGPA() {

int off = 0, cred = 0;

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

off += (subject[i].grade * subject[i].credit);

cred += subject[i].credit;

}

}

class Student {

public static void main (String args[]) {

{

Student stud = new Student();

System.out.println ("Pratham");

stud.getStudentDetails();

stud.getMarks();

System.out.println ("Name");

System.out.println ("SGPA = " + stud.computeSGPA());

}

Name of student is -

Output: Enter name and USN

Pratham

18M22CS206

Enter no. of subjects : 2

8/12/23

26/12/23

Lab 3

```
import java.util.*;
```

```
class Book
```

{

```
String name, author;
```

```
int price, 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()
```

{

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

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

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

```
System.out.println("Number of pages: " + this.numPages);
```

```
return this.name + this.author + this.price + this.numPages;
```

}

```
} // end of class
```

```
public class BookMain
```

{

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

```
System.out.println("Prathom Gangpathy | 1BM22CS206);
```

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

```
int i, n, price, numPages;
```

```
String name, author;
```

```
System.out.println ("Enter number of books");
```

```
n = sc.nextInt();
```

```
Book b[] = new Book[n];
for (i=0; i<n; i++)
{
```

```
    System.out.println("Enter Name of the Book");
    name = SC.next();
```

```
    System.out.println("Enter name of the author");
    author = SC.next();
```

```
    System.out.println("Enter the price");
    price = SC.nextInt();
```

```
    System.out.println("Enter number of pages");
    numPages = SC.nextInt();
```

```
    b[i] = new Book(name, author, price, numPages);
```

```
}
```

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

```
    System.out.println("Book " + (i+1) + " details");
    b[i].toString();
}
```

```
System.out.println();
}
```

```
}
```

Output: Pratham Ganapathy 18M22CS206

Enter number of books
2

Enter name of author the book

b1

Enter name of the author
a1

Enter the price

10

Enter the number of pages
12

Enter name of the book

b2

Enter name of the author

a2

Enter the price

11

Enter number of pages

13

Book1 details :

Name : b1

Author : a1

Price : 10

Number of pages : 12

Book2 details :

Name : b2

Author : a1

Price : 11

Number of pages : 13

86
20/2/23

2/1/24 import java.util.*;

class InputScanner

{

int d1, d2;

Scanner SC = new Scanner (System.in);

InputScanner ()

{

if (this.getClass() == Circle.class)

{

System.out.println ("Enter d1: ")

d1 = SC.nextInt(); }

else

{

System.out.println ("Enter d1 and d2: ")

d1 = SC.nextInt();

d2 = SC.nextInt();

}

} abstract class Shape extends InputScanner

} abstract void printArea();

class Triangle extends Shape

{

void printArea()

{

System.out.println ("Area of triangle is: "+(double)(d1*d2)/2);

}

class Rectangle extends Shape

{ void printArea() {

} System.out.println("Area of rectangle is: "+(double)(d1*d2));

}

class Circle extends Shape

{ void printArea() {

} System.out.println("Area of circle is: "+(double)(3.14*d1*d1));

}

class AreaMain {

public static void main(String args[]) {

System.out.println("Pratham - IBM 22CS206");

Rectangle r = new Rectangle();

Triangle t = new Triangle();

Circle c = new Circle();

r.printArea();

t.printArea();

c.printArea();

}

}

Output: Pratham - IBM 22CS206

Enter d1 and d2:

1 2

Entered d1 and d2:

3 4

Enter d1:

3

Area of rectangle is : 2.0

Area of triangle is : 6.0

Area of circle is : 28.259998

8
20/12X

import java.util.*;
 import java.lang.*;

class Account

```
{  

  String name;  

  int acc-no;  

  boolean current;  

  double balance = 0;  

  int main-balance = 100;  

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

Account () {

```
if (this.getClass() == Current Ac. class)  

{ current = true; }
```

else

```
{ current = false; }
```

```
System.out.print ("Enter name: ");  

name = SC.next();
```

```
System.out.print ("Enter account number: ");  

acc-no = SC.nextInt();
```

}

void deposit()

```
{  

  System.out.print ("Enter deposit amount: ");  

  balance += SC.nextDouble();}
```

}

* void withdraw () {

```
System.out.print ("Enter withdraw amount: ");  

double withdraw = SC.nextDouble();
```

```
while (withdraw > balance) {
```

```
System.out.println ("Withdraw amount is greater");}
```

void withdraw (double withdraw) {

if (withdraw > balance)

System.out.println ("Withdraw amount greater than
balance");

} if (current ft balance < min_balance)

System.out.println ("Below min balance \$100,"

remaining money in account");

}

}

void showBalance () {

System.out.println ("Balance = " + balance);

}

}

class CurrentAcc extends Account {

void cheque () {

System.out.print ("Enter cheque amount: ");

double cheque = SC.nextDouble ();

withdraw (cheque);

System.out.println ("Cheque created...");

}

class SavingsAcc extends Account {

void compound (int t, int s) {

}

balance = balance * (Math.pow ((1 + (double)s / 100), t));

class Bank {

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

 savingAcc acc1 = new savingAccount();
 currentSmith = new currentAccount();

 Account ref = null;

 Scanner SC = new Scanner (System.in);

 int acc, choice;

 System.out.println ("----- Menu ----- \n");

 while (choice != 6)

 if (choice == 1)

 { ref.deposit (); }

 else if (choice == 2)

 if (acc == 1)

 { john.compound (1,5); }

 else

 { System.out.println ("Not savings"); }

 } else if (choice == 4)

 { ref.showBalance (); }

 else if (choice == 5)

 { if (acc == 2) { smith.cheque (); } }

 else

 { System.out.println ("Not current"); }

}

 System.out.println ("Enter acc.no.");

 acc = SC.nextInt ();

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

choice = SC.wrt Int(); } }

Output

~~Enter name : John~~

Enter account no. : 1

Enter new : Smith

Enter account no. : 2

--- MENU ---

1. Deposit

2. Withdraw

3. Com

4. Display

5. Change

6. Exit

choice : 1

Enter acc. no : 1

Enter deposit amount : 100

Enter acc. no : 2

Enter choice : 6

09/01/29

16/1/2024

- 1) String length : 4
- 2) Concatenated String : somebmse
- 3) Converting Integer to String: 55 → 55
- (+) 115 111 109 101 (5) BMSCE equals BMSCE : true
- BMSCE equals some : false
- BMSCE equals IC Bmsce : true

(6) true

7) true

8) false

9) Result: "BMSCE"

10. apple ball yates zee

11. 1 2 3 4 5 6 7 8 9 10

12. Initial string : This is a test. This is too

Final string : Thomas was a test. Thomas was too

13. Concatenated String : holloworld

14. Find string : camouflage

15. Find string : Hello friends

16 \ 01 / 2024

18 Length : 10

S Something gnithemor gn

L Something

thing c

19 Eagle fly method Eagle make sound method Hawk fly method Hawk makes sound

20 Enter a, b, c

3 4 5

Enter radius : 10

Area of triangle : 6.0

Perimeter of triangle : 12.0

Area of circle : 314.0

Perimeter of circle : 62.80004

16. Hello Friend

17. Enter the no of students;

2

Enter the name:

Prathom

Enter the register number:

206

enter the cgpa: 2.8

2.8

Enter the name:

Rishi

Enter the register number

222

Enter the cgpa:

8.4

Sorted by cgpa:

name: Rishi reg no: 222 sem: 3 cgpa: 8.4

name: Prathom reg no: 206 sem: 3 cgpa: 2.8

8

3/1/24

Student.java:

```
package cie;
public class Student {
    public String name, usn;
    public int sem;
}
```

Internal.java:

```
package cie;
import java.util.*;
public class Internal extends Student {
    public int marks[] = new int [5];
    public void InputMarks()
}
```

Scanner SC = new Scanner (System.in);
 for (int i=0; i<5; i++)

System.out.println("Enter subject " + (i+1) + " mark ");
 marks[i] = SC.nextInt();

public void displayMarks()

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

System.out.println("Subject " + (i+1) + " marks: " + marks[i]);

External.java:

package see;

```
import cie.Student;
import java.util.*;
public class External extends Student {
    public int marks[] = new int [5];
    public void inputMarks()
```

```
Scanner SC = new Scanner (System.in);
for (int i=0; i<5; i++)
```

```
    System.out.println ("Enter marks for subject " + (i+1) + " marks:");
    marks[i] = SC.nextInt();
```

{

}

```
public void displayMarks()
```

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

```
    System.out.println ("Subject " + (i+1) + " marks: " + marks[i]);
```

}

}

Main.java :

```
import cie.Student;
import cie.Internal;
import cie.External;
import java.util.*;
```

class Main {

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

```
    int no = 2;
```

```
    External finalmarks[] = new External [no];
```

```
    Internal intmarks[] = new Internal [no];
```

```
    for (int i = 0; i < no; i++)
```

```
        finalmarks[i] = new External ();
```

```
        intmarks[i] = new Internal ();
```

```
        finalmarks[i].inputMarks();
```

```
        intmarks[i].inputMarks();
```

}

```
    for (int i = 0; i < no; i++)
```

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

```
        finalmarks[i].displayMarks();
```

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

```
        finalmarks[i].displayMarks();
```

}

}

OUTPUT

Enter subject 1 marks : 30
Enter subject 2 marks : 50
Enter subject 3 marks : 40
Enter subject 4 marks : 20
Enter subject 5 marks : 10

Enter subject 1 marks : 80
Enter subject 2 marks : 20
Enter subject 3 marks : 60
Enter subject 4 marks : 90
Enter subject 5 marks : 90
Enter subject 1 marks : 70
Enter subject 2 marks : 40
Enter subject 3 marks : 20
Enter subject 4 marks : 80
Enter subject 5 marks : 10

CIE:

Subject 1 marks : 30
Subject 2 marks : 50
Subject 3 marks : 60
Subject 4 marks : 20
Subject 5 marks : 10

SEE:

Subject 1 marks : 30
Subject 2 marks : 70
Subject 3 marks : 60
Subject 4 marks : 80
Subject 5 marks : 90

CIE:

Subject 1 marks : 70

Subject 2 marks : 40

Subject 3 marks : 20

Subject 4 marks : 20

Subject 5 marks : 10

SEE:

Subject 1 marks : 80

Subject 2 marks : 60

Subject 3 marks : 30

Subject 4 marks : 20

Subject 5 marks : 30

86
23/01/2021

30/1/24

CLASSMATE

Date _____

Page _____

```
import java.util.*;
```

```
class WrongAge extends Exception
```

```
{ public WrongAge()
```

```
{ super ("Age Error"); }
```

```
{ public WrongAge (String message)
```

```
{ super (message); }
```

```
}
```

```
class InputScanner
```

```
{
```

```
public static int getIntInput (String prompt)
```

```
{ Scanner SC = new Scanner (System.in);  
System.out.println (prompt);  
return SC.nextInt(); }
```

```
class Father extends InputScanner
```

```
{
```

```
public int fatherAge ;
```

```
public Father () throws WrongAge
```

```
{ fatherAge = getIntInput ("Enter Father's Age");  
if (fatherAge < 0) }
```

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

class Son extends Father

private int sonAge;

public Son() throws WrongAge

super();

sonAge = getIntInput ("Enter Son's Age");

if (sonAge < getInt > fatherAge)

if (sonAge > super.fatherAge)

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

else if (sonAge < 0)

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

else if (sonAge == super.fatherAge)

throw new WrongAge ("Son age cannot be equal to father's age");

public void display()

super.display();

System.out.println ("Son's Age = " + sonAge);

public class ExceptionHandling

public static void main (String args [])

try

son son = new Son();

son.display();

} catch (WrongAge e)

} system.out.println("Error: " + e.getMessage());

}

}

OUTPUT Enter Father's age: 45

Enter Son's age: 12

Father's age: 45

Son's age: 12

Enter Father's age: -2

Error: Age cannot be negative

Enter Father's age: 19

Enter Son's age: 20

Son's age cannot be greater than father.

8
29/01/24

6/2/24

```
import java.util.*;
import java.io.*;
```

```
class B extends Thread {
    public void run() {
        try {
            for(int i = 0; i < 3; i++) {
                System.out.println("BMS");
                Thread.sleep(10000);
            }
        } catch(InterruptedException e) {
            System.out.println(e);
        }
    }
}
```

```
class C extends Thread {
    public void run() {
        try {
            for(int i = 0; i < 3; i++) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch(InterruptedException e) {
            System.out.println(e);
        }
    }
}
```

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

B b = new B();

C c = new C();

b. start();

c. start();

{

{

OUTPUT BMS

CSE

CSE

CSE

BMS

BMS

x

x

—

class A {

synchronized void foo(B b)

{ String name = Thread.currentThread().getName();

System.out.println(name + " entered A.foo");

try {

Thread.sleep(1000);

}

catch (Exception e)

{

System.out.println("A interrupted");

System.out.println(name + " trying to call B.last()");

B.last();

}

void last()

{

System.out.println("Inside A.last");

class B {

{ synchronized void bar (A a)

{ String name = Thread.currentThread().getName();
 System.out.println(name + " entered B.bar");
 try {

} Thread.sleep(1000);

{ catch (Exception e)

} System.out.println("B Interrupted");

} System.out.println(name + " trying to call A.last()");
 a.last();

void last ()

{ System.out.println("Inside A.last");

}

class Deadlock implements Runnable {

A a = new A();

B b = new B();

Deadlock ()

{

Thread.currentThread().setName("Main Thread");

Thread t = new Thread(this, "Racing Thread");

t.start();

a.foo(b); // get lock on a in this thread

System.out.println("Back in main thread"); }

public void run()

{
 b.bar(a);
 System.out.println("Back in main thread");
}

public static void main(String args[])

{
 new Deadlock();
}

OUTPUT

Main Thread entered A.foo

Racing Thread enters B.bar

Main Thread trying to call B.last()

Inside A.last

Back in main Thread

Racing Thread trying to call A.last()

Inside A.last

Back in other thread.

S
Q6 (or) m

Proton

class Q {

int n;

boolean valueSet = false;

synchronized int get() {

while (!valueSet) {

try {

System.out.println("Consumer Wait,");

} catch (Exception e) {

System.out.println("Got: " + n);

valueSet = true;

System.out.println("Tell producer");

notify();

return n;

}

synchronized void put(int n)

while (valueSet) {

try {

System.out.println("Producer waiting");

} catch (InterruptedException e) {

System.out.println(e);

}

this.n = n;

valueSet = true;

System.out.println("Put: " + n);

System.out.println("Tell 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 < 3) {
        q.put(i++);
    }
}
```

class Consumer implements Runnable {

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

public void run() {
    int i = 0;
    while(i < 3) {
        int n = q.get();
        System.out.println("Consumed: " + n);
        i++;
    }
}
```

& class Procon {

```
public static void main(String args[]) {
    Q q = new Q();
    new Producer(q);
    new Consumer(q);
}
```

0010101

Put : 1

Got : 1

Put : 2

Got : 2

Put : 3

Got : 3

8/21m
13/21m

Da
Pa

20-2-24

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

```
class SwingDemo
{
```

```
    SwingDemo()
    {
```

```
        JFrame jfrm = new JFrame("Dividers App");
        jfrm.setSize(275, 150);
```

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

```
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        JLabel jLab = new JLabel("Enter the numbers and divide it");
```

```
        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 onelab = new JLabel();
```

```
        jfrm.add(err);
```

```
        jfrm.add(alab);
```

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

```
        jfrm.add(button);
```

```
        jfrm.add(onelab);
```

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

```
        jfrm.add(err);
```

```
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);

 anlab.setText("\nAns = " + ans);

}

 } catch (NumberFormatException e) {

 alab.setText("::");

 blab.setText("::");

 anlab.setText("::");

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

}

 } catch (ArithmaticException e) {

 alab.setText("");

 blab.setText("");

 anlab.setText("");

 err.setText("B should be NON zero!");

 } ifrm.setVisible(true);

public static void main(String args[]) {

```
swingUtilities.invokeLater(new Runnable() {
    public void run() {
        new SwingDemo();
    }
});
```

OUTPUT:

Enter the divisor and dividend

[120 | 30]

calculator A=120 B=30 Ans = 4

Classes Used:

JFrame: Top-level container for Swing components

JLabel: A non-editable text label to display info

JTextField: A single-line text field for user input

JButton: A clickable button that triggers action

FlowLayout: A layout manager that arranges components in horizontal flow

Action Listener: An interface for handling action events (like button click)

Action Event: An event object representing an action

swingUtilities: A utility class for working with

Swing components on the Event Dispatch Thread (EDT)

Function:

new JFrame(title): specifies title

setSize(width, height): set frame size

setLayout(new FlowLayout()): sets layout manager to flowlayout

setVisible(true): makes frame visible

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE): Terminates program when closed

JFrame
JPanel
JLabel
JText
JText
JText
JText
JText

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 discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
class Quadratic{
    int a, b, c;
    double r1, r2, d;
    void getd(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter coeffs a, b, c: ");
        a = sc.nextInt(); b = sc.nextInt(); c = sc.nextInt();
    }
    void compute(){
        while(a == 0){
            System.out.println("Invalid coeff, enter new one: ");
            Scanner sc = new Scanner(System.in);
            a = sc.nextInt();
        }
        d = b*b - 4*a*c;
        if (d == 0){
            r1 = -b/(2*a);
            System.out.println("Roots are real and equal");
            System.out.println("r1 = r2 = " + r1);
        }
        else if (d > 0){
            r1 = (-b + Math.sqrt(d))/(2*a);
            r2 = (-b - Math.sqrt(d))/(2*a);
            System.out.println("Roots are real and distinct");
            System.out.println("r1 = " + r1 + "; " + "r2 = " +
r2);
        }
        else if (d < 0){
            r1 = -b/(2*a);
            r2 = Math.sqrt(-d)/(2*a);
            System.out.println("Roots are imaginary");
            System.out.println("r1 = " + r1 + " + i" + r2 + "r2 =
" + r1 + " - i" + r2);
        }
    }
}
class QuadraticMain{
    public static void main(String args[]){
        Quadratic q = new Quadratic();
```

```
q.getd();
q.compute();
}
}
```

Output:

```
PS C:\Users\bmsce\Desktop\1BM22CS206> javac .\quadratic.java
PS C:\Users\bmsce\Desktop\1BM22CS206> java quadratic
Pratham Ganapathy      1BM22CS206
Enter coefficients of a,b,c
4 5 6
Roots are imaginary
Root1 = 0.0+i1.0532687216470449 Root2 = 0.0-i1.0532687216470449
PS C:\Users\bmsce\Desktop\1BM22CS206> java quadratic
Pratham Ganapathy      1BM22CS206
Enter coefficients of a,b,c
1 -2 1
Roots are real and equal
Root1 = Root2 = 1.0
PS C:\Users\bmsce\Desktop\1BM22CS206> java quadratic
Pratham Ganapathy      1BM22CS206
Enter coefficients of a,b,c
1 -3 2
Roots are real and distinct
Root1 = 2.0  Root2= 1.0
PS C:\Users\bmsce\Desktop\1BM22CS206> |
```

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 Subject{
int SubjectMarks; int credits; int grade;
}
class Student{
Subject subject[];
String name, usn;
double sgpa;
Scanner sc;
int n;
Student(){
subject = new Subject[10];
sc = new Scanner(System.in);
```

```
System.out.println("Enter no. of subjects: ");
n = sc.nextInt();
for(int i = 0; i < 9; i++){
subject[i] = new Subject();
}
sc.nextLine();
}
void getStudentDetails(){
System.out.println("Enter name: ");
name = sc.nextLine();
System.out.println("Enter usn: ");
usn = sc.nextLine();
}
void getMarks(){
System.out.println("\n");
for(int i = 0; i < n; i++){
System.out.println("Enter no. of credits: ");
subject[i].credits = sc.nextInt();
System.out.println("Enter marks obtained: ");
subject[i].SubjectMarks = sc.nextInt();
System.out.println("\n");
if (subject[i].SubjectMarks > 100)
subject[i].SubjectMarks = 100;
else if (subject[i].SubjectMarks < 40)
subject[i].SubjectMarks = 0;
subject[i].grade = (subject[i].SubjectMarks / 10) + 1;
if (subject[i].grade == 11) subject[i].grade = 10;
if (subject[i].SubjectMarks >= 40 &&
subject[i].SubjectMarks < 50) subject[i].grade = 4;
else if (subject[i].SubjectMarks >= 50 &&
subject[i].SubjectMarks < 55) subject[i].grade = 5;
else if (subject[i].SubjectMarks >= 55 &&
subject[i].SubjectMarks < 60) subject[i].grade = 6;
}
}
double computeSGPA(){
int effective = 0, credits = 0;
for(int i = 0; i < n; i++){
effective += (subject[i].grade * subject[i].credits);
credits += subject[i].credits;
}
sgpa = effective/credits;
return sgpa;
}
```

```
}

class StudentMain{
public static void main(String args[]){
Student student = new Student();
System.out.println("Pratham Ganapathy 1BM22CS206");
student.getStudentDetails();
student.getMarks();
System.out.println("Name of student is: " + student.name);
System.out.println("USN of student is: " + student.usn);
System.out.println("SGPA of student is: " +
student.computeSGPA());
}
}
```

Output:

```
Pratham Ganapathy 1BM22CS206
Enter name and USN
Pranav
1BM22CS203
Enter marks and credits
50
3
Enter marks and credits
60 2
Enter marks and credits
50 1
Name: Pranav      USN: 1BM22CS203
SGPA= 5 .6666666666666667

C:\Users\admin\Desktop\BMS\Sem3\00J\Lab>
```

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.

```
import java.util.Scanner;
class Book{
String name, author;
int price, no_pages;
public Book(String name, String author, int price, int no_pages){
this.name = name;
this.author = author;
```

```
this.price = price;
this.no_pages = no_pages;
}
public String toString(){
System.out.println("Name: " + this.name);
System.out.println("Author: " + this.author);
System.out.println("Price: " + this.price);
System.out.println("Pages: " + this.no_pages);
return this.name + this.author + this.price + this.no_pages;
}
}
class BookMain{
public static void main(String args[]){
System.out.println("Pratham Ganapathy 1BM22CS206");
Book books[] = new Book[10];
Scanner sc = new Scanner(System.in);
System.out.println("Enter no. of book objects: ");
int n = sc.nextInt();
sc.nextLine();
for(int i = 0; i < n; i++){
String name, author;
int price, no_pages;
System.out.println("Enter name: ");
name = sc.next();
System.out.println("Enter author: ");
author = sc.next();
System.out.println("Enter price: ");
price = sc.nextInt();
System.out.println("Enter no. of pages: ");
no_pages = sc.nextInt();
books[i] = new Book(name, author, price, no_pages);
}
System.out.println("\n");
for(int i = 0; i < n; i++){
System.out.println("Book " + (i+1) + " Details:\n");
books[i].toString();
System.out.println("\n");
}
}
}
```

Output:

```
Pratham Ganapathy      1BM22CS206
Enter number of bookd
2
Enter name of the book
b1
Enter name of the author
a1
Enter the price
50
Enter number of pages
20
Enter name of the book
b2
Enter name of the author
a2
Enter the price
100
Enter number of pages
257
Book1 details
Name: b1
Author: a1
Price: 50
Number of pages: 20

Book2 details
Name: b2
Author: a2
Price: 100
Number of pages: 257
```

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 classShape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;
class InputScanner{
int d1, d2;
Scanner sc = new Scanner(System.in);
```

```
InputScanner(){
if(this.getClass() == Circle.class){
System.out.println("Enter d1: ");
d1 = sc.nextInt();
}
else{
System.out.println("Enter d1 and d2: ");
d1 = sc.nextInt();
d2 = sc.nextInt();
}
}
}

abstract class Shape extends InputScanner{
abstract void printArea();
}

class Triangle extends Shape{
void printArea(){
System.out.println("Area of triangle is: " +
(double)(d1*d2)/2);
}
}

class Rectangle extends Shape{
void printArea(){
System.out.println("Area of rectangle is: " +
(double)(d1*d2));
}
}

class Circle extends Shape{
void printArea(){
System.out.println("Area of circle: " +
(double)(3.14*d1*d1));
}
}

class AreaMain{
public static void main(String args[]){
Rectangle r = new Rectangle();
Triangle tr = new Triangle();
Circle c = new Circle();
r.printArea();
tr.printArea();
c.printArea();
}
}
```

Output:

```
Enter d1 and d2:  
5 6  
Enter d1 and d2:  
5 2  
Enter d1:  
5  
Area of rectangle is: 30.0  
Area of triangle is: 5.0  
Area of circle: 78.5
```

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;  
import java.lang.*;  
class Account {  
    String name;  
    int acc_no;  
    boolean current;  
    double balance = 0;  
    int min_balance = 100;  
    Scanner sc = new Scanner(System.in);  
    Account(){  
        if(this.getClass() == CurrentAcc.class){  
            current = true;  
        } else {  
            current = false;  
        }  
        System.out.print("Enter name: ");  
        name = sc.next();  
        System.out.print("Enter account no.: ");
```

```
acc_no = sc.nextInt();
}
void deposit() {
System.out.print("Enter deposit amount: ");
balance += sc.nextDouble();
}
void withdraw() {
System.out.print("Enter withdraw amount: ");
double withdraw = sc.nextDouble();
while (withdraw > balance) {
System.out.print("Withdraw amount greater than
balance, enter new amount: ");
withdraw = sc.nextDouble();
}
balance -= withdraw;
if (current && balance < min_balance) {
System.out.println("Below min balance of 100, removing
remaining money in account");
balance = 0;
}
}
void withdraw(double withdraw) {
if (withdraw > balance) {
System.out.println("Withdraw amount greater than
balance");
}
if (current && balance < min_balance) {
System.out.println("Below min balance of 100, removing
remaining money in account");
balance = 0;
}
}
void showBalance() {
System.out.print("balance = " + balance);
}
}

class CurrentAcc extends Account {
void cheque(){
System.out.print("Enter cheque amount: ");
double cheque = sc.nextDouble();
withdraw(cheque);
System.out.println("Cheque created...");
}
}
```

```
class SavingsAcc extends Account {  
    void compound(int t, int r) {  
        balance = balance * (Math.pow((1 + ((double) r / 100)), t));  
        System.out.print("Balance after given rate and time = " +  
            balance);  
    }  
}  
  
class Bank {  
    public static void main(String args[]) {  
        SavingsAcc john = new SavingsAcc();  
        CurrentAcc smith = new CurrentAcc();  
        Account ref = null;  
        Scanner sc = new Scanner(System.in);  
        int acc, choice;  
        System.out.println("-----MENU-----\n");  
        System.out.println(  
            "1.Deposit\n2.Withdraw\n3.Compute interest for  
            Savings Acc\n4.Display account details\n5. Create  
            cheque\n6.Exit\nChoice:");  
        choice = sc.nextInt();  
        System.out.println("Enter account no.: ");  
        acc = sc.nextInt();  
        if (acc == 1) {  
            ref = john;  
        } else {  
            ref = smith;  
        }  
        while (choice != 6) {  
            if (choice == 1) {  
                ref.deposit();  
            } else if (choice == 2) {  
                ref.withdraw();  
            } else if (choice == 3) {  
                if (acc == 1) {  
                    john.compound(1, 5);  
                } else {  
                    System.out.println("Not a savings  
                        account");  
                }  
            } else if (choice == 4) {  
                ref.showBalance();  
            } else if (choice == 5) {  
                if (acc == 2) {  
                    smith.cheque();  
                }  
            }  
        }  
    }  
}
```

```
} else {
System.out.println("Not a current
account");
}
}
System.out.println("Enter account no.: ");
acc = sc.nextInt();
System.out.println("-----MENU-----\n");
System.out.println(
"1.Deposit\n2.Withdraw\n3.Compute intrest
for Savings Acc\n4.Display account details\n5. Create
cheque\n6.Exit\nChoice:");
choice = sc.nextInt();
}
}
}
```

Output:

```
Enter name: a
Enter account no.: 1
Enter name: b
Enter account no.: 2
-----MENU-----

1.Deposit
2.Withdraw
3.Compute intrest for Savings Acc
4.Display account details
5. Create cheque
6.Exit
Choice:
1
Enter account no.:
1
Enter deposit amount: 100
Enter account no.:
1
```

Lab 6

**Demonstrate the utilization of String and StringBuffer functions as well as the
usage of abstract classes**

```
import java.util.*;
class StringMain {
```

```

public static void main(String args[]) {
/* 1 */ char arr[] = { 'B', 'M', 'S', 'C', 'E' };
String s1 = new String(arr);
String s2 = new String("bmsce");
String s3 = new String(s2);
/* 2 */ String s4 = "some";
System.out.println("String length: " + s4.length() + "\n" +
"Concatenated string: " + s4.concat(s2));
/* 3 */ int d = 55;
String sd = Integer.toString(d);
System.out.println("Converting Integer to string: " + d + "
-> " + sd);
/* 4 */ char res[] = new char[20];
String str = new String("Welcome to BMSCE College");
str.getChars(10, 16, res, 0);
/* 5 */ byte byte_arr[] = s4.getBytes();
for (int i = 0; i < 4; i++) {
System.out.print(byte_arr[i] + " ");
}
/* 6 */ System.out.println("BMSCE equals BMSCE: " +
s1.equals("BMSCE"));
System.out.println("BMSCE equals some: " + s1.equals(s4));
System.out.println("BMSCE equalsIC Bmsce: " +
s1.equalsIgnoreCase(s2));
/* 7 */ System.out.println(str.regionMatches(11, "BMSCE
College", 0, 11));
/* 8 */ System.out.println(str.startsWith("Welcome"));
/* 9 */ System.out.println(str.endsWith("College"));
/* 10 */ String s5 = new String("BMSCE");
System.out.println("Reference equal b/w s1 and s5 (==): " +
(s1 == s5));
System.out.println("Value equal b/w s1 and s5 (equals()): " +
+ s1.equals(s5));
/* 11 */ String str_arr[] = { "van", "watch", "ball", "cat",
"xmas", "yatch", "zee", "apple", "ice", "jug",
"kite", "lift", "man", "net", "orange", "dog",
"ent", "free", "gun", "hen", "parrot", "queen", "ring",
"star", "tree", "umbrella" };
for (int i = 0; i < str_arr.length; i++) {
for (int j = i + 1; j < str_arr.length; j++) {
if (str_arr[i].compareTo(str_arr[j]) > 0) {
String temp;
temp = str_arr[i];
str_arr[i] = str_arr[j];
str_arr[j] = temp;
}
}
}
}

```

```

str_arr[j] = temp;
}
}
}
for (int i = 0; i < str_arr.length; i++) {
System.out.print(str_arr[i] + " ");
}
/*12*/ String num_arr[] = {"1", "4", "3", "2", "5"};
for (int i = 0; i < num_arr.length-1; i++) {
for (int j = i + 1; j < num_arr.length; j++) {
if (num_arr[i].compareTo(num_arr[j]) > 0) {
String temp;
temp = num_arr[i];
num_arr[i] = num_arr[j];
num_arr[j] = temp;
}
}
}
System.out.println("\n");
for (int i = 0; i < num_arr.length; i++) {
System.out.print(num_arr[i] + " ");
}
System.err.println("\n");
/*13*/ String originalString = "This is a test. This is,
too.";
String replacedString = new String("");
int beginIndex = 0;
int indexOfis = originalString.indexOf("is");
while (indexOfis != -1) {
replacedString += originalString.substring(beginIndex,
indexOfis);
System.out.println(replacedString);
replacedString += "was";
System.out.println(replacedString);
beginIndex = indexOfis+2;
indexOfis = originalString.indexOf("is", indexOfis + 2);
if (indexOfis == -1) replacedString +=
originalString.substring(beginIndex);
}
// System.out.println(originalString.substring(indexOfis+2));
System.out.println("Original string: " + originalString);
System.out.println("Replaced string: " + replacedString);
/*14*/ System.out.println("Concatenation example: hello + world
= " + "hello".concat("world")));

```

```

/*15*/ System.out.println("Replacing ll with mm gives us: " +
"College".replace("ll", "mm"));
/*16*/ System.out.println("Trimming example: " + " Hello World
.trim());
/*18*/
StringBuffer exp = new StringBuffer("Something");
exp.setLength(10);
char[] buffer = new char[4]; exp.getChars(0, 4, buffer, 0);
String buf = new String(buffer);
System.out.println("Length: " + exp.length());
System.out.println(exp.charAt(0) + " " + buf + " " +
exp.reverse() + " " + exp.substring(0, 3)); exp.reverse();
exp.append('c'); exp.insert(0, 'K'); exp.replace(0, 1, "L");
System.out.println(exp);
exp.deleteCharAt(0); exp.delete(0, 4);
System.out.println(exp);
}

}

ShapeMain.java
import java.util.Scanner;
import java.lang.Math;
class InputScanner{
int d1, d2, d3;
Scanner sc = new Scanner(System.in);
InputScanner(){
if(this.getClass() == Circle.class){
System.out.println("Enter d1: ");
d1 = sc.nextInt();
}
else{
System.out.println("Enter a, b, c: ");
d1 = sc.nextInt();
d2 = sc.nextInt();
d3 = sc.nextInt();
}
}
}

abstract class Shape extends InputScanner{
abstract void calculateArea();
abstract void calculatePerimeter();
}

class Triangle extends Shape{
void calculateArea(){
double s = (d1+d2+d3)/2;

```

```
System.out.println("Area of triangle is: " +
(double)Math.sqrt(s*(s-d1)*(s-d2)*(s-d3)));
}
void calculatePerimeter(){
System.out.println("Perimeter of triangle is: " +
(double)(d1+d2+d3));
}
}
class Circle extends Shape{
void calculateArea(){
System.out.println("Area of circle: " +
(double)(3.14*d1*d1));
}
void calculatePerimeter(){
System.out.println("Perimeter of circle: " +
(double)(3.14*2*d1));
}
}
class ShapeMain{
public static void main(String args[]){
System.out.println("Pranav Y - 1BM22CS204");
Triangle tr = new Triangle();
Circle c = new Circle();
tr.calculateArea(); tr.calculatePerimeter();
c.calculateArea(); tr.calculatePerimeter();
}
}
BirdMain.java
abstract class Bird{
abstract void fly();
abstract void makeSound();
}
class Eagle extends Bird{
void fly(){
System.out.println("Eagle fly method");
}
void makeSound() {
System.out.println("Eagle sound method");
}
}
class Hawk extends Bird{
void fly() {
System.out.println("Hawk fly method");
}
}
```

```
void makeSound() {  
    System.out.println("Hawk sound method");  
}  
}  
  
class BirdMain {  
    public static void main(String[] args) {  
        Eagle e = new Eagle(); Hawk h = new Hawk();  
        e.fly();h.fly();e.makeSound();h.makeSound();  
    }  
}
```

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 see;
import cie.Student;
import java.util.Scanner;
public class Externals extends Student{
    public int marks[] = new int[5];
    public void inputMarks() {
        Scanner sc = new Scanner(System.in);
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter subject " + (i + 1) + " marks: ");
            marks[i] = sc.nextInt();
        }
    }
    public void displayMarks() {
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i + 1) + " marks: " + marks[i]);
        }
    }
}
Internals.java
package cie;
import java.util.Scanner;
public class Internals extends Student {
    public int marks[] = new int[5];
    public void inputMarks() {
        Scanner sc = new Scanner(System.in);
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter subject " + (i + 1) + " marks: ");
            marks[i] = sc.nextInt();
        }
    }
    public void displayMarks() {
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i + 1) + " marks: " +
```

```
marks[i]);
}
}
}
Student.java
package cie;
public class Student {
public String name, usn;
public int sem;
}
Main.java
import cie.Student;
import cie.Internals;
import see.Externals;
import java.util.Scanner;
class Main{
public static void main(String args[]){
int no = 2;
Externals finalmarks[] = new Externals[no];
Internals intmarks[] = new Internals[no];
for (int i = 0; i < no; i++){
finalmarks[i] = new Externals();
intmarks[i] = new Internals();
finalmarks[i].inputMarks();
intmarks[i].inputMarks();
}
for(int i = 0; i < no; i++){
System.out.println("CIE: ");
intmarks[i].displayMarks();
System.out.println("SEE: ");
finalmarks[i].displayMarks();
}
}
}
```

Output:

```
Enter subject 1 marks:  
10  
Enter subject 2 marks:  
10  
Enter subject 3 marks:  
10  
Enter subject 4 marks:  
10  
Enter subject 5 marks:  
10  
Enter subject 1 marks:  
10  
Enter subject 2 marks:  
10  
Enter subject 3 marks:  
10  
Enter subject 4 marks:  
10  
Enter subject 5 marks:  
10  
Enter subject 1 marks:  
10  
Enter subject 2 marks:  
10  
Enter subject 3 marks:  
10  
Enter subject 4 marks:  
10  
Enter subject 5 marks:  
10  
Enter subject 1 marks:  
10  
Enter subject 2 marks:  
10  
Enter subject 3 marks:  
10  
Enter subject 4 marks:  
101  
Enter subject 5 marks:  
10  
CIE:  
Subject 1 marks: 10  
Subject 2 marks: 10  
Subject 3 marks: 10  
Subject 4 marks: 10  
Subject 5 marks: 10  
SEE:  
Subject 1 marks: 10  
Subject 2 marks: 10  
Subject 3 marks: 10  
Subject 4 marks: 10  
Subject 5 marks: 10  
CIE:  
Subject 1 marks: 10  
Subject 2 marks: 10  
Subject 3 marks: 10
```

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 25 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;  
import java.lang.Exception;  
class WrongAge extends Exception{  
WrongAge(String s){  
super(s);  
}  
}  
class InputScanner {
```

```
Scanner sc = new Scanner(System.in);
int FatherAge, SonAge;
InputScanner(){
if (this.getClass() == Father.class){
System.out.println("Enter father age: ");
FatherAge = sc.nextInt();
} else if (this.getClass() == Son.class){
System.out.println("Enter son age: ");
SonAge = sc.nextInt();
}
}
}

class Father extends InputScanner{
int FatherAge;
Father() throws WrongAge{
FatherAge = super.FatherAge;
// System.out.println("Father Age: " + FatherAge);
if (FatherAge < 0){
throw new WrongAge("Age cannot be < 0 for a person");
}
}
void display(){
System.out.println("Father Age = " + FatherAge);
}
}

class Son extends Father{
int SonAge;
Son(Father f) throws WrongAge{
SonAge = super.SonAge;
// System.out.println("Enter son age: ");
// SonAge = sc.nextInt();
// System.out.println("Father Age: " + f.FatherAge + "; Son Age:
" + SonAge);
if (f.FatherAge < SonAge){
throw new WrongAge("Age cannot be greater for son");
} else if (SonAge < 0){
throw new WrongAge("Age cannot be < 0 for a person");
}
}
void display(){
System.out.println("Son Age = " + SonAge);
}
}

class FatherMain{
```

```
public static void main(String[] args) {  
    try{  
        Father father = new Father();  
        Son son = new Son(father);  
        father.display();  
        son.display();  
    }  
    catch (WrongAge e){  
        System.out.println(e.getMessage());  
    }  
}
```

Output:

```
Enter father age:  
20  
Enter son age:  
10  
Father Age = 20  
Son Age = 10
```

```
Enter father age:  
15  
Enter son age:  
16  
Age cannot be greater for son
```

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.

```
import java.io.*;
class B extends Thread{
public void run(){
try{
for(int i = 0; i < 3; i++){
System.out.println("BMS"); Thread.sleep(10000);
}
} catch (InterruptedException e){
System.out.println(e);
}
}
}
}
}
class C extends Thread{
public void run(){
try{
for(int i = 0; i < 3; i++){
System.out.println("CSE"); Thread.sleep(2000);
}
} catch (InterruptedException e){
System.out.println(e);
}
}
}
}
}
class ThreadMain{
public static void main(String args[]){
B b = new B();
C c = new C();
b.start(); c.start();
}
}
```

Output:

```
BMS
CSE
CSE
CSE
BMS
BMS
```

Lab 9

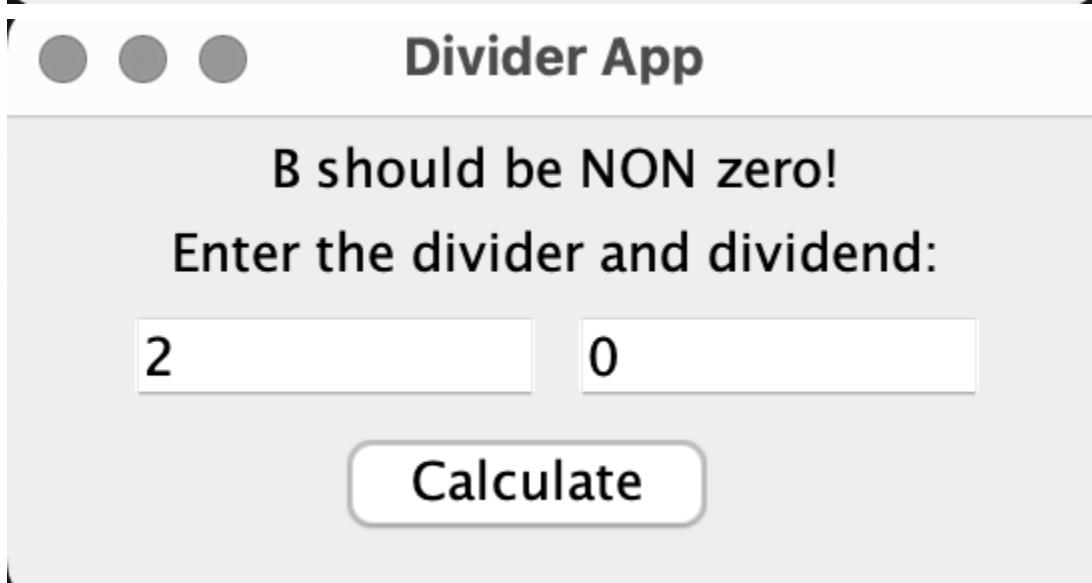
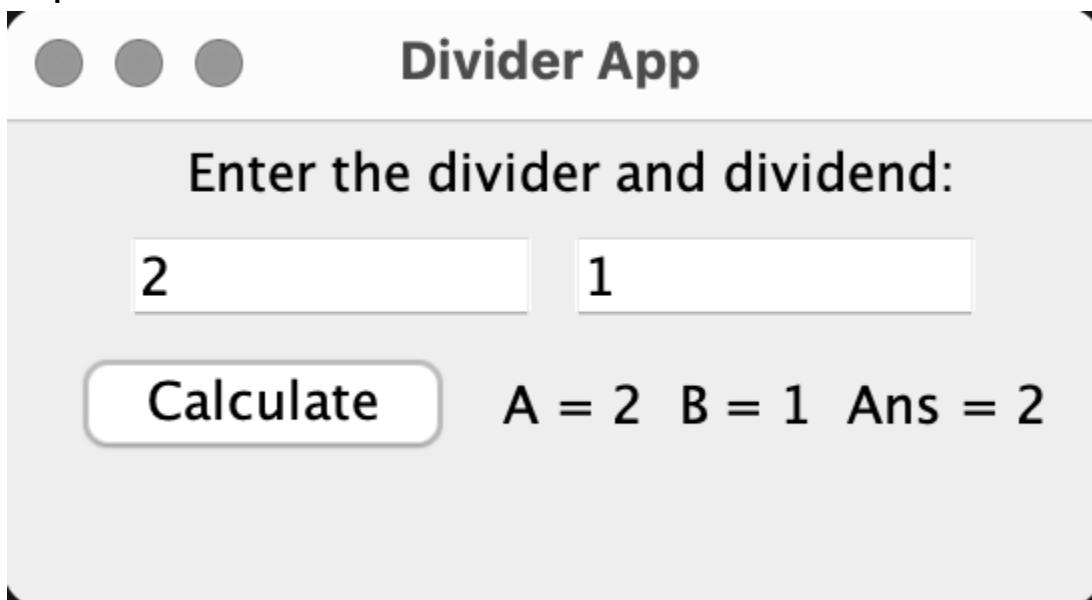
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo {
    SwingDemo() {
        // create jframe container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // text label
        JLabel jlab = new JLabel("Enter the divider and dividend:");
        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        // calc button
        JButton button = new JButton("Calculate");
        // labels
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
        // add in order
        jfrm.add(err); // to display error
        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!");
}
}
});
// display frame
jfrm.setVisible(true);
}

public static void main(String args[]) {
// create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable() {
public void run() {
new SwingDemo();
}
});
}
}
```

Output:



Lab 10

Demonstrate Inter process Communication and deadlock.

IPC

```
ProCon.java
class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while (!valueSet)
            try {
                wait();
            }
```

```
        } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
        System.out.println("Got: " + n);
        valueSet = false;
        notify();
        return n;
    }
    synchronized void put(int n) {
        while (valueSet)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        this.n = n;
        valueSet = true;
        System.out.println("Put: " + n);
        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 < 5) {
            q.put(i++);
        }
    }
}
class Consumer implements Runnable {
    Q q;
    Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }
    public void run() {
        int i = 0;
        while (i < 5) {
            int r = q.get();
        }
    }
}
```

```
i++;
}
}
}
}

class ProCon {
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: 0
Got: 0
Put: 1
Got: 1
Put: 2
Got: 2
Put: 3
Got: 3
Put: 4
Got: 4
```

Deadlock.java

```
class A {
synchronized void foo(B b) {
String name = Thread.currentThread().getName();
System.out.println(name + " entered A.foo");
try {
Thread.sleep(1000);
} catch (Exception e) {
System.out.println("A Interrupted");
}
System.out.println(name + " trying to call B.last()");
b.last();
}
void last() {
System.out.println("Inside A.last");
}
}

class B {
synchronized void bar(A a) {
```

```

String name = Thread.currentThread().getName();
System.out.println(name + " entered B.bar");
try {
    Thread.sleep(1000);
} catch (Exception e) {
    System.out.println("B Interrupted");
}
System.out.println(name + " trying to call A.last()");
a.last();
}
void last() {
    System.out.println("Inside A.last");
}
}

class Deadlock implements Runnable {
A a = new A();
B b = new B();
Deadlock() {
    Thread.currentThread().setName("MainThread");
    Thread t = new Thread(this, "RacingThread");
    t.start();
    a.foo(b); // get lock on a in this thread.
    System.out.println("Back in main thread");
}
public void run() {
    b.bar(a); // get lock on b in other thread.
    System.out.println("Back in other thread");
}
}

public static void main(String args[]) {
new Deadlock();
}
}

```

Output:

```

MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
MainThread trying to call B.last()
Inside A.last
Inside A.last
Back in main thread
Back in other thread

```