

B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



LAB

REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

PRANAV Y

(1BM22CS204)

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560

0192023-2024.

INDEX

Sl.N o.	Title	Date
1	Complete scanned Observation Book	12/12/2023 - 20/02/2024
2	Lab 1	12/12/202 3
3	Lab 2	19/12/202 3
4	Lab 3	26/12/202 3
5	Lab 4	02/01/202 4
6	Lab 5	09/01/202 4
7	Lab 6	16/01/202 4
8	Lab 7	23/01/202 4
9	Lab 8	30/01/202 4
10	Lab 9	06/02/202 4
11	Lab 10	20/02/202 4

12/12/23

12/12/23

Date: / /

Lab program - 1

```

class Quadratic 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 coeff 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 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 +
                " + " + r2);
        } else if (d<0) {
            r1 = -b/(2*a);
            r2 = (Math.sqrt(-d)) / (2*a);
            System.out.println ("r1 = " + r1 + " + i" + r2 +
                " + " + r2 + " + " + r1 + " - i" + r2);
        }
    }
}

```

?

?

?

class Quadratic Main {

 public static void main (String args []) {

 Quadratic q = new Quadratic () ;

 q. getall () ;

 q. computel () ;

}

}

o/p

① Enter a, b, c

1 4 2, 1

Roots are equal

$\lambda_1 = \lambda_2 = -1.0$

② Enter a, b, c

1. -4 3 2

Roots are real and distinct

$\lambda_1 = -1.0 \quad \lambda_2 = 2.0$

③ Enter a, b, c

1. 1 1

Roots are imaginary

$\lambda_1 = 0.0 + i0.866 \quad \lambda_2 = 0.0 - i0.866$

2. Lab program - 2

```
import java.util.Scanner;  
class Subject {  
    int Subject Marks; int credit; int grade;  
}
```

```
class Student {
```

```
    Subject subject[7];
```

```
    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<n; 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].credit = sc.nextInt();

System.out.println ("Enter marks obtained: ");

Subject[i].SubjectMark = sc.nextInt();

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

if (Subject[i].SubjectMark > 100)

Subject[i].SubjectMark = 100;

else if (Subject[i].SubjectMark < 40)

Subject[i].SubjectMark = 0; 1/10

Subject[i].grade = (Subject[i].SubjectMark + 1); +1;

if (Subject[i].grade == 11)

Subject[i].grade = 10;

if (Subject[i].grade == 1)

Subject[i].grade = 0;

if (Subject[i].SubjectMark >= 40 &&

Subject[i].SubjectMark <= 50)

Subject[i].grade = 4;

else if (Subject[i].SubjectMark >= 50 &&

Subject[i].SubjectMark <= 58)

Subject[i].grade = 5;

else if (Subject[i].SubjectMark >= 58 &&

Subject[i].SubjectMark <= 60)

Subject[i].grade = 6;

}

}

double compute_SOPAC() {

int effective = 0, credit = 0;

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

effective += (Subject[i].grade * Subject[i].^{credit});

credit += Subject[i].credit;

}

sgpa = effective 1 credit;
return sgpa;

{

{

class Student Main {
public static void main (String args[]) {
class Student Main {
Student student = new Student();
System.out.println("Pranav Y - 1Bm22CS204");
student.getStudentDetails();
student.getMark();
System.out.println("Name of student is: " + student.
System.out.println("USN of student is: " + student.
System.out.println("SGPA of student is: " +
student.computeSGPA());

{

{

Q1) Enter no of subjects:

3

Pranav Y - 1Bm22CS204

Enter name:

Some Prana

Enter USN:

CS204

Enter no. of credits:

5

Enter marks obtained:

100

Enter no. of credits:

3

Enter marks obtained:

93

Enter no. of credits:

1

Enter marks obtained:

97

Name of Student ^{is}: Pranav

USN of Student is: CS204

SGPA of student is: 10.0

✓
19/12/23

3. Lab Program - 3

```
import java.util.Scanner;  
class Book {  
    String name, author;  
    int price, noPage;  
    public Book (String name, String author, int price)  
    {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.noPage = noPage;  
    }  
    public String toString () {  
        System.out.println ("Name: " + this.name);  
        System.out.println ("Author: " + this.author);  
        System.out.println ("Price: " + this.price);  
        System.out.println ("Page: " + this.noPage);  
        return this.name + this.author + this.price + this.noPage;  
    }  
}
```

class BookMain {

```
public static void main (String args []) {  
    System.out.println ("Pranav M - 1BM22CA204");  
    Book books [] = new Book [10];  
    Scanner sc = new Scanner (System.in);  
    System.out.println ("Enter no. of book object: ");  
    int n = sc.nextInt();  
    sc.nextLine();  
    for (int i = 0; i < n; i++) {  
        String name, author;  
        int price, noPage;  
        System.out.println ("Enter name: ");  
        name = sc.nextLine();  
    }  
}
```

```
System.out.println("Enter author: ");
author = sc.nextLine();
```

```
System.out.println("Enter price: ");
price = sc.nextInt();
```

```
System.out.println("Enter no. of pages: ");
noPages = sc.nextInt();
```

```
books[i] = new Book(name, author, price, noPages);
```

{
3}

```
System.out.println("\n");
```

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

```
System.out.println("Book " + (i+1) + " Details: ");
book[i].toString();
```

```
System.out.println("\n");
```

3

3

olp Pranav Y - IBM22CS204

Enter no. of book objects:

3

Enter name:

b1

Enter author:

a1

Enter price:

100

Enter no. of pages:

300

Enter name:

b2

Enter author:

a2

Enter Date:

20/0

Enter no. of pages:

100

Enter name:

b1

Enter Author:

a1

Enter Price:

100

Enter no. of pages:

10

~~Book 1 details~~

Name: b1

Author: a1

Price: 100

Page: 300

~~Book 2 details~~

Name: b2

Author: a2

Price: 200

Page: 100

~~Book 3 details~~

Name: b3

Author: a3

Price: 100

Page: 10

g
100 p.
100 p.

U. Lab Program - 4

```
import java.util.Scanner;  
class InputScanner {  
    int d1, d2;  
    Scanner sc = new Scanner(System.in);  
    InputScanner () {  
        if (this == Object.getConstructor().newInstance()) {  
            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: " + (d1 * d2) / 2);  
    }
```

```
}  
class Rectangle extends Shape {
```

```
    void printArea () {
```

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

```
}  
class Circle extends Shape {
```

```
    void printArea () {
```

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

```
}
```

class AreaMain {

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

o/p

Ex Phonix 4 - 1BM122E1204

Enter d1 and d2:

1

2

Enter d1 and d2:

3

4

Enter d1:

3

Area of rectangle is: 2.0

Area of triangle is: 6.0

Area of circle: 28.279998

Lab Program 1

```
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.getAccn() == CurrentAcc.accn) {  
    current = true;
```

```
} else {
```

```
current = false;
```

```
}
```

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

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

```
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 > balance,  
withdraw = sc.nextDouble();")
```

```
?
```

balance -> withdraw;

if (current & d balance < min-balance);

System.out.println("Below min balance");
balance = 0;

}

}

void showBalance();

void withdraw(); ~~withdraw~~ withdraw

System.out.print("balance = " + balance);

}

}

class CurrentAcc extends Account {

void cheque() {

System.out.println("Enter cheque amt.");

double cheque = sc.nextDouble();

withdraw(); balance -= cheque;

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

}

class SavingsAcc extends Account {

void compound(int t, double r) {

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

System.out.println("Balance after rate and time
+ balance");

}

class Bank {

public static void main(String args[]) {

System.out.println("Saving Acc joins new Saving Acc());

(* Phony - Current Acc Smith = new CurrentAcc();
B1221204); Account ref;

Scanner sc = new Scanner(System.in);

int acc, choice;

System.out.println(" --- MENU --- ");

System.out.println(" 1. Deposit 2. Withdraw 3.
Compound Interest 4. Display");

In S. Create cheq in G. Exit in Choice: ")

choice = sc. nextInt(1);

if ~~ba ==>~~ ac = sc. nextInt(1);

if (ac == 1) { ref = john; }

else { ref = smith; }

while (choice != 6) {

if (choice == 1) {

ref. deposit(1);

} else if (choice == 2) {

ref. withdraw(1);

} else if (choice == 3) {

if (ac == 1) {

john. compound(1, 5);

} else {

System.out.println("Not saving");

}

} else if (choice == 4) {

ref. showBalance();

} else if (choice == 5) {

if (ac == 2) {

smith. cheq(1);

} else {

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

}

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

acc = sc.nextInt();

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

choice = sc.nextInt();

{

}

Output:

Prana 4 - IBM 22 C1204

Enter name: John

Enter account no.: 1

Enter name: Smith

Enter account no.: 2

--- M.F.W ---

1. Deposit

2. Withdraw

3. Compute Interest

4. Display

5. Cheque

6. Exit

Choice: 1

Enter acc. no.: 1

Enter deposit amount: 100

Enter acc. no.: 2

Enter choice: 6

8/1/24
a/1

Lab 8: Lab Program - C

Q1 P 1)

1) String created with char array 20) ~~Enter a,b,c~~

~~2) String created with literal~~

String created with String

2) $55 = \text{some}$

String length = 4

Concatenated string = Somebrace

~~2 3 4 ✓~~

Enter ab^{radic}

10

Area of triangle: 6.0

Perimeter

Area of triangle: 12.0

3) Converting Integer to string: $55 \rightarrow 55$ Area of circle: 314.0

4) 115 1191 109 101

Perimeter of circle:

5) BMSCE equals BMSCE: true

62.8000u.

BMSCE equals some: ~~false~~x

BMSCE equals IL brace: true

6) ~~task~~ startWith ("Welcome") : true

7) endsWith ("College") : true

8) Reference equal blw st and st (==) : & false

Value equal blw st and st (equal(1)) : true

9) getChar (1) successful; Result = "BMSCE"

10) apple ball ... yatch zee

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

12) Initial String: This is a test. This is too ~~too~~

Final String: Thwas was a testy. Thwas wa too

13) Concatenated String: helloworld

14) Find String: Commage

15) Find String: Hello Friends

16) Length: 10

Something gnintemos gn

Something^{2y}

thing c

17) Eg: Eagle fly method Eagle makeSound method

Hawk fly method Hawk makeSound method

Lab Lab Program - 47→ Student.java:

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

{

→ Internal.java:

```
package cie;
import java.util.Scanner;
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) + " marks");
            marks[i] = sc.nextInt();
        }
    }
}
```

{

{

→ External.java:

```
package cie;
import cie.Student;
import java.util.Scanner;
```

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 subject " + (i+1) + "
```

```
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 cir.Student;
```

```
import cir.Internals;
```

```
import cir.External;
```

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

```
class Main {
```

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

```
int no = 2; External finalmarks[] = new External[no];
```

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

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

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

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

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

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

```
}
```

```
for (int i = 0; i < no; i++)  
    System.out.println("C.F.F: ");  
    int mark = i; displayMark(i);  
    System.out.println("S.E.E: ");  
    findMark(i).displayMark();
```

{

}

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: 30  
Enter Subject 2 Marks: 70  
Enter Subject 3 Marks: 60  
Enter Subject 4 Marks: 80  
Enter Subject 5 Marks: 90  
Enter Subject 1 Marks: 320  
Enter Subject 2 Marks: 40  
Enter Subject 3 Marks: 30  
Enter Subject 4 Marks: 10  
Enter Subject 5 Marks: 0  
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: CTF:

Subject 1 Mark: 30

Subject 2 Mark: 50

Subject 3 Mark: 40

Subject 4 Mark: 20

Subject 5 Mark: 10

SFF:

Subject 1 Mark: 30

Subject 2 Mark: 70

Subject 3 Mark: 60

Subject 4 Mark: 80

Subject 5 Mark: 90

CIE:

Subject 1 Mark: 20

Subject 2 Mark: 40

Subject 3 Mark: 30

Subject 4 Mark: 10

Subject 5 Mark: 0

SFF:

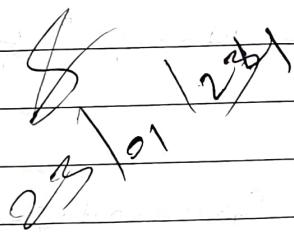
Subject 1 Mark: 70

Subject 2 Mark: 40

Subject 3 Mark: 20

Subject 4 Mark: 80

Subject 5 Mark: 10


01
01

Lab Program - 8

FatherMain.java

```
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 Age;
```

```
    InputScanner () {
```

```
        if (this.getClass () == Father.class) {
```

```
            System.out.println ("Enter father age:");
```

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

```
}
```

```
{
```

```
{
```

```
class Father extends InputScanner {
```

```
    int FatherAge;
```

```
    Father () throws WrongAge {
```

```
        FatherAge = Age;
```

```
        if (FatherAge < 0) {
```

```
            throw new WrongAge ("Age cannot be less than 0");
```

```
{
```

```
{
```

```
void display () { System.out.println ("Father Age: " + FatherAge);}
```

```
{
```

class Son extends Father {

int SonAge;

SonAge throws WrongAge {

super();

System.out.println("Enter son age: ");

SonAge = Scanner.next();

if (FatherAge < SonAge)

throw new WrongAge ("Age cannot be > father");

{ else if (SonAge < 0) }

throw new WrongAge ("Age cannot be < 0");

}

}

SonAge);

void display() { System.out.println("Son Age: " + }

}

class Father Main {

public static void main (String args[]) {

try {

Father father = new Father();

Son son = new Son();

father.display();

son.display();

} catch (WrongAge e) {

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

}

}

Output:

1) Enter father age:

10

2) Enter father age:

-1

Enter son age:

5

Age cannot be < 0

Father Age: 10

Son Age: 5

3) Enter father age:

14

Enter son age:

16

Age cannot be greater for son

6/01/24
30/01/24

Lab Program - 9

Thread Main . java

```
import java.lang.*;  
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");
```

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

```
                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();
```

```
}
```

```
}
```

old SNS

CSE

CSE

CSE

BMS

BMS

Lab Program - 10Procon.java

class Q {

int n;

boolean valueSet = false;

synchronized int get() {

while (!valueSet) {

try {

System.out.println("Consumer Waiting");

wait();

} catch (InterruptedException e) {

System.out.println(e);

}

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

valueSet = false;

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

notify();

}

return n;

}

synchronized void put(int n) {

while (valueSet) {

try {

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

wait();

} 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 a = q.get ();

System.out.println ("consumed: " + a);

i++;

}

}

}

class ProCon {

public static void main (String args []) {

Q q = new Q();

new Producer (q); new Consumer (q);

}

}

olp

Put: 1

Get: 1

Put: 2

Get: 2

Put: 3

Get: 3

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.lock());
 a.lock();

{
 void lock();}

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

}

{
 class Deadlock implements Runnable {

A a = new A();

B b = new B();

DeadLock() {

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

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

t.start();

a.foo(b);

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

}

public void run() {

b.bar(a);

System.out.println("Back in other");

}

public static void main(String args[]) {

new DeadLock();

}

{

Output:

Main Thread entering A.foo

Racing Thread entering B.bar

Main Thread trying to call B.lock()

Inside A.lock

Back in main thread

Racing Thread trying to call A.lock()

Inside A.lock

8/2/22
6/2/22

Lab Program - 9

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

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame ("Divide App");
        jfrm.setSize (275, 150);
        jfrm.setLayout (new FlowLayout (1));
        jfrm.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel ("Enter dividend : ");
        JTextField ajtf = new JTextField (8);
        JTextField bjtf = new JTextField (8);

        JButton button = new JButton ("Calculate");
        JLabel one = new JLabel ();
        JLabel alab = new JLabel ();
        JLabel blab = new JLabel ();
        JLabel anslab = new JLabel ();

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

        ActionListener l = new ActionListener () {
            public void actionPerformed (ActionEvent e) {
                System.out.println ("Action event for a " + e.getActionCommand ());
            }
        };
        ajtf.addActionListener (l);
        bjtf.addActionListener (l);
    }
}
```

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

int a = Integer.parseInt(tf1.getText());

int b = Integer.parseInt(tf2.getText());

int ans = b/a;

alab.setText("In A = " + a);

blab.setText("In B = " + b);

anslab.setText("In Ans = " + ans);

} catch (NumberFormatException e) {

alab.setText(" ");

blab.setText(" ");

anslab.setText(" ");

anslab.setText("Enter integer!");

} catch (ArithmaticException e) {

alab.setText(" ");

blab.setText(" ");

anslab.setText(" ");

anslab.setText("B should be != 0");

}

}

});

jfrm.setVisible(true);

});

public static void main (String args[]) {

SwingUtilities.invokeLater (new Runnable() {

public void run() {

new SwingDemo();

}

});

}

Output:

Divider App

Enter Dividend and divisor

2

1

Calculate

 $A = 2 \quad B = 1 \quad Ans = 2$

Class 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 actions.
- FlowLayout: A layout manager that arranges components horizontally.
like buttons
- ActionListener: An interface for handling action events.
- ActionEvent: An event object representing an action.
- SwingUtilities: A utility class for working with Swing components on the Event Dispatching Thread (EDT).

Functions:

- `new JFrame (text)`: Specifies title.
- `setSize (width, height)`: Sets size of frame.
- `setLayout (new FlowLayout ())`: Sets layout manager to FlowLayout.
- `setVisible (true)`: Makes frame visible.
- `setDefaultCloseOperation (JFrame.EXIT)`: Terminates program.
when close
- `new JLabel (text)`: Specifies text in label.
- `JTextField (size)`: Creates text field of specified size.
columns
- `new JButton (text)`: Creates button with given text.
- `addActionListener (ActionListener l)`: Attaches an Action Listener to a component to handle action events.
- ActionEvent actionPerformed (ActionEvent evt): Method called when an action event occurs.
- Invocable (Runnable r): Schedules a runnable task to be executed on EDT.

Additional:

~~new setError ()~~: Used to display error message on specific JLabel (JLabel object). ~~setText ()~~: Used to update labels with input value and calculate result.

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:

```
● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % javac Quadraticmain.java
● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % java QuadraticMain
Enter coeffs a, b, c:
1 2 1
Roots are real and equal
r1 = r2 = -1.0
```

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("Pranav Y - 1BM22CS204");
        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:

```
● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % javac StudentMain.java
● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % java StudentMain
Enter no. of subjects:
3
Pranav Y - 1BM22CS204
Enter name:
a
Enter usn:
22

Enter no. of credits:
4
Enter marks obtained:
98

Enter no. of credits:
3
Enter marks obtained:
97

Enter no. of credits:
1
Enter marks obtained:
95

Name of student is: a
USN of student is: 22
SGPA of student is: 10.0
```

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("Pranav Y - 1BM22CS204");
        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:

```

● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % javac BookMain.java
● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % java BookMain
Pranav Y - 1BM22CS204
Enter no. of book objects:
2
Enter name:
a
Enter author:
ab
Enter price:
10
Enter no. of pages:
15
Enter name:
b
Enter author:
ba
Enter price:
20
Enter no. of pages:
20

Book 1 Details:

Name: a
Author: ab
Price: 10
Pages: 15

Book 2 Details:

Name: b
Author: ba
Price: 20
Pages: 20

```

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[]){
        System.out.println("Pranav Y - 1BM22CS204");
        Rectangle r = new Rectangle();
        Triangle tr = new Triangle();
        Circle c = new Circle();
        r.printArea();
        tr.printArea();
        c.printArea();
    }
}
```

Output:

```
● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % javac AreaMain.java
● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % java AreaMain
Pranav Y - 1BM22CS204
Enter d1 and d2:
2 4
Enter d1 and d2:
10 2
Enter d1:
100
Area of rectangle is: 8.0
Area of triangle is: 10.0
Area of circle: 31400.0
```

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.nextLine();
        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 intrest 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:

```

● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % javac Bank.java
○ (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % java Bank
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;  
                }  
            }  
        }  
    }  
}
```

```

        }
    }
    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 index0fis = originalString.indexOf("is");
while (index0fis != -1) {
    replacedString += originalString.substring(beginIndex,
index0fis);
    System.out.println(replacedString);
    replacedString += "was";
    System.out.println(replacedString);
    beginIndex = index0fis+2;
    index0fis = originalString.indexOf("is", index0fis + 2);
    if (index0fis == -1) replacedString +=
originalString.substring(beginIndex);
}
// System.out.println(originalString.substring(index0fis+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:

```

(base) pranavyadlapati@Pranav-MacBook-Air Java_learn % javac cie/Student.java
(base) pranavyadlapati@Pranav-MacBook-Air Java_learn % javac see/Externals.java
(base) pranavyadlapati@Pranav-MacBook-Air Java_learn % javac Main.java
(base) pranavyadlapati@Pranav-MacBook-Air Java_learn % java Main
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:

```

● (base) pranavyadlapati@Pranavs-MacBook-Air Java_learn % javac FatherMain.java
● (base) pranavyadlapati@Pranavs-MacBook-Air Java_learn % java FatherMain
Enter father age:
20
Enter son age:
10
Father Age = 20
Son Age = 10
● (base) pranavyadlapati@Pranavs-MacBook-Air Java_learn % java FatherMain
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:

```
● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % javac ThreadMain.java
● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % java ThreadMain
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 bjtff = 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(bjtff);
        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 (ArithmException 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:



Divider App

Enter the divider and dividend:

Calculate

A = 2 B = 1 Ans = 2



Divider App

B should be NON zero!

Enter the divider and dividend:

Calculate

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:

```

● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % javac ProCon.java
● (base) pranavyadlapati@Pranav's-MacBook-Air Java_learn % java ProCon
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 B.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:

```

● (base) pranavyadlapati@Pranavs-MacBook-Air Java_learn % javac Deadlock.java
● (base) pranavyadlapati@Pranavs-MacBook-Air Java_learn % java Deadlock
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

```

