

B.M.S COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



JAVA LAB REPORT

OBJECT ORIENTED JAVA PROGRAMMING

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

R Abhinav
1BM22CS211

Submitted to:

Dr. Seema Patil
Assistant Professor
BMS College of Engineering

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) // Print Hello world!

class Hello {

 public static void main(String[] args) {

 System.out.println("Hello, World!");

}

}

Output:

Hello, World!

2) // Calculate area of rectangle

class area {

 public static void main(String[] args) {

 int length = Integer.parseInt(args[0]);

 int breadth = Integer.parseInt(args[1]);

 System.out.println("Length: " + length);

 System.out.println("Breadth: " + breadth);

 System.out.println("Area is " + length * breadth);

}

}

Output: java area 10 8
Length = 10 | Area is 80
Breadth = 8

3) Scanner class methods

```
import java.util.Scanner;  
  
class Input {  
    public static void main(String[] args) {  
        String a, int b;  
        Scanner input = new Scanner(System.in);  
        System.out.print("Enter String:");  
        a = input.nextLine();  
        System.out.println("String is "+a);  
  
        System.out.print("Enter Integer:");  
        b = input.nextInt();  
        System.out.print("Integer is "+b);  
    }  
}
```

Output:

Enter String: Abhimanyu

String is Abhimanyu

Enter Integer: 20

Integer is 20

4) // Arrays

class array

```
public static void main(String[] args){  
    int a[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};  
    System.out.println("July has " + a[6] + " days");  
}
```

Output:

July has 31 days.

5) // Factorial

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

class factorial

```
public static void main(String[] args){  
    Scanner input = new Scanner(System.in);  
    System.out.print("Enter a number: ");  
    int num = input.nextInt();  
    int fact = 1;  
    for(int i = 1; i <= num; i++){  
        fact *= fact * i;  
    }
```

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

6) // Palindrome

```
import java.util.Scanner;  
class palin {  
    public static void main(String[] args) {  
        int num, copy, rev = 0;  
        Scanner in = new Scanner(System.in);  
        num = in.nextInt();  
        copy = num;  
  
        while (num != 0) {  
            rev = (rev * 10) + (num % 10);  
            num = num / 10;  
        }  
  
System.out.println("  
        if (copy == rev) {  
            System.out.println("It is a palindrome");  
        }  
        else {  
            System.out.println("It is not a palindrome");  
        }  
    }  
}
```

Output:

Enter number: 1234

It is not a palindrome.

//Quadratic equation roots

import java.util.Scanner;

class quad{

int a,b,c;

double r1,r2,d;

void coeff()

Scanner s = new Scanner (System.in);

a = s.nextInt();

b = s.nextInt();

c = s.nextInt();

void compute(){

while(a==0)

if(a==0){

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

}

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

if(d==0){

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

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

System.out.println("Roots are "+r1);

}

else if(d>0){

```
r1 = (-b) + Math.sqrt(b*b - 4*a*c)/(2*a);
```

```
r2 = (-b) - Math.sqrt(b*b - 4*a*c)/(2*a);
```

```
System.out.println("Root 1 = " + r1 + " Root 2 = " + r2);
```

```
}
```

```
else if (d < 0)
```

```
{
```

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

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

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

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

```
System.out.println("Root 1 = " + r1 + " - i " + r2);
```

```
}
```

```
{
```

```
}
```

```
class Quadratic Main
```

```
{
```

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

```
{
```

```
Quadratic qv = new Quadratic();
```

```
qv.gtd();
```

```
qv.computa();
```

```
}
```

```
}
```

Output:

Enter the coefficients of a,b,c : 1 2 1

Roots are real and equal

Root 1 = Root 2 = -1.0

Enter the co-efficients of a,b,c : 1 -5 6

Roots are real and distinct

Root 1 = -2.0

Root 2 = -3.0

Enter the co-efficient of a,b,c : 1 3 4

Roots are imaginary

Root 1 = -1 + i 1.3228

Root 2 = -1 - i 1.3228

Enter the co-efficient of a,b,c : 0 2 3

Not a quadratic equation

Enter a non zero value for a : 1 -7 16

Root 1 = -2.00

Root 2 = -5.00

Lab - 2

SGPA Calculator

//Program to read marks and calculate SGPA

```
import java.util.Scanner;  
  
class Student {  
    String USN, name;  
    int n;  
    Student () {  
        String subject[] = new String[10];  
        int credits[] = new int[10];  
        int marks[] = new int[10];  
    }  
  
    void read() {  
        Scanner input = new Scanner(System.in);  
        System.out.println("Enter your name:");  
        name = input.nextLine();  
        System.out.println("Enter number of subjects:");  
        n = input.nextInt();  
        for (int i = 0; i < n; i++) {  
            System.out.println("Enter subject:");  
            System.out.println("Enter marks:");  
            subject[i] = input.nextLine();  
            System.out.println("Enter marks:");  
            marks[i] = input.nextInt();  
        }  
    }  
}
```

```
System.out.println("Enter credits:");
credits[i] = input.nextInt();
```

```
void display() {
```

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

```
    System.out.println("USN: " + num);
```

```
    System.out.println("Subject\tCredits\tMarks");
```

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

```
        System.out.println(subject[i] + credits[i] + mark[i]);
```

```
}
```

```
}
```

```
int grade(int a) {
```

```
    while (a % 10 != 0) {
```

```
        a = a / 10;
```

```
}
```

```
    if ((a + 1) > 5) {
```

```
        return (a + 1);
```

```
}
```

```
    else if (@+1 == 5) {
```

```
        return a';
```

```
}
```

```
    else { return 0; }
```

```
}
```

```
void calculateSGPA() {
    int total_credits = 0;
    int marks_credits = 0;
    for (int i = 0; i < n; i++) {
        total_credits += credits[i];
        marks_credits += grad(marks[i]) * credits[i];
    }
    System.out.println("SGPA is " + (marks_credits / total_credits));
}

class marks {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.read();
        s1.display();
        s1.calculateSGPA();
    }
}
```

Output

Enter your name: Abhimar

Enter your USN: 1BM22CS211

Enter number of subjects: 4

-- -- -- --
Enter subject: Math

Enter credits: 4

Enter marks: 96

-- -- -- --
Enter subject: Java

Enter credits: 4

Enter marks: 88

Enter subject: Web

Enter credits: 3

Enter marks: 89

Enter subject: English

Enter credits: 1

Enter marks: 98

-- -- -- -- -- -- --
Name:

Name: Abhinav
VSN: IBM22 (S21)

Subject	Credits	Marks
Math	4	96
Java	4	88
Web	3	89
English	1	98
—	—	—
SGPA is	9.41666666	

~~Left~~
~~19.12.24~~

Lab-3

/*Program to read and print book details.

~~class~~ ~~title~~

public

```
import java.util.Scanner;
class books {
    String author, book;
    int price, numPages;
    books(String book, String author, int price, int numPages) {
        this.book = book;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString() {
        return "Book Name:" + this.name + "in Author:" + this.author +
               "in price:" + this.price + "\nPage number" + this.numPages;
    }
}
```

```
class lib{  
    String author, book;  
    int price, numPages;  
    int numBooks;  
    Scanner input = new Scanner(System.in);  
    numBooks = input.nextInt();  
    books b[] = new books[numBooks];  
  
    for(int i=0; i<numBooks; i++){  
        input.nextLine();  
        System.out.println("Enter book name:");  
        book = input.nextLine();  
  
        System.out.println("Enter author name:");  
        author = input.nextLine();  
  
        System.out.println("Enter Price:");  
        price = input.nextInt();  
  
        System.out.println("Enter Page number:");  
        numPages = input.nextInt();  
  
        b[i] = new books(book, author, price, numPages);  
    }  
  
    for(int i=0; i<numPages; i++){  
        String a = b[i].toString();  
        System.out.println(a);  
    }  
}
```

Output:

Book 1

Enter book name: Harry Potter

Enter author: JK Rowling

Enter the Price: 799

Enter page No.: 889

Book 2

Enter book name: Poohbumps

Enter author: RL Nime

Enter Price: 450

Enter number of pages: 236

Book 1 details

Name:

Book Name: Harry Potter

Author: JK Rowling

Price: 799

Page count: 889

Book 2 Details

Book Name: Poohbumps

Author: RL Nime

Price: 450

Page count: 236

4
26/12/23.

Lab-4

2/01/2024

```
import java.util.Scanner;  
class inputScanner{  
    void rec(rectangle ab){  
        System.out.println("Enter the dimensions  
        of the rectangle (Length  
        and Breadth):");  
        ab.a = input.nextInt();  
        ab.b = input.nextInt();  
    }  
    void tri(triangle ab){  
        Scanner x = new Scanner(System.in);  
        ab.a = input.nextInt();  
        ab.b = input.nextInt();  
    }  
}  
abstract class shape extends InputScanner{  
    int a, b;  
    abstract void printArea();  
}
```

class rectangle extends shape {

rectangle() {

rec(this);

}

void printArea() {

System.out.println("Area of Rectangle
= "+(double)(a*b));

}

}

class triangle extends shape {

triangle() { tri(this); }

void printArea() {

System.out.println("Area of Triangle = "+(0.5*a*

)

}

class circle extends shape {

circle() { cir(this); }

void printArea() {

System.out.println("Area of Circle = "+(3.14*a*a))

}

}

```
class calc  
public static void main(String [] args){  
    rectangle r = new rectangle();  
    triangle t = new triangle();  
    circle c = new circle();  
    r.printArea();  
    t.printArea();  
    c.printArea();  
}
```

}

Output:

Enter dimensions of rectangle (Length and ~~height~~ ^{breadth}):

45

Enter dimensions of ~~triangle~~ ^{triangle} (or base and height):

89

Enter radius of circle:

6

Area of Rectangle = 20.0

Area of triangle = 36.0

Area of Circle = 113.039

8
21/1/24

```
import java.util.Scanner;
import
class Account {
    String name;
    int acc_no;
    boolean current;
    double balance = 0;
    int main_balance = 0;
    Scanner sc = new Scanner(System.in);

    Account() {
        if (this.getClass() == CurrentAcc.class) {
            current = true;
        } else {
            current = false;
        }
        name = sc.next();
        System.out.print("Enter account no.: ");
        acc_no = sc.nextInt();
    }

    void deposit() {
        System.out.print("Enter deposit amt: ");
        balance = sc.nextDouble();
    }
}
```

```
void withdraw() {
    System.out.print("Enter deposit amount: ");
    if (amount > this.balance) {
        System.out.println("Not enough balance");
    } else {
        this.balance -= amount;
    }
}

void showBalance() {
    System.out.print("Balance = " + balance);
}

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

class SavingAcc extends Account {
 void compound(int t, int r) {
 balance = balance * Math.pow((1 + (double)r / 1000), t);
 System.out.print("Balance after given rate : ");
 }
}

Class Bank {

public static void main(String args[]) {
 SavingAcc john = new SavingAcc();
 CurrentAcc mith = new CurrentAcc();
 Account ref = null;
 System.out.println("---- Menu ----")

System.out.println("1. Deposit \n2. Withdraw
3. Compute Interest
4. Display account
details
5. Create Cheque
6. Exit \n Choice : ")

choice = sc.nextInt();

System.out.println("Enter account no. : ")

acc = sc.nextInt();

if (acc == 1)

ref = john;

else ref = mith;

```
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  
            mith.compound(1, 5);  
    } else if (choice == 4){  
        mith.change();  
    }  
}
```

```
3  
System.out.println("Enter account no.");  
acc = sc.nextInt();  
System.out.println(acc);  
3  
3
```

Output:

Enter account name: John
Enter account number: 1

--- MENU ---

1. Deposit
2. Withdraw
3. Compute Interest for Saving Account
4. Display account details
5. Exit

choice: 1

Enter amount: 2000

Amount added! Balance: 2000

choice: 2

Enter amount: 180

Amount withdrawn! Balance 2820

choice: 4

Account name: John
Account number: 1
Balance: 2820
CJS 09.01.24

Output from 1 to

- ① Constructor with one parameter : abcdef.
 Constructor with $\text{startindex} = 1$ and $\text{numchars} = 5$.
 b cde

Constructing s2 from s1 ; bcde
 string from ansi ; bcd

- ② Demonstrating String lengths, literal and concatenation.

Length of abcdef is : 7

Length of abc using literal : 3

Concatenation of strings.

- ③ ToString demonstration

characts is 72.0 bpm and S02 level is 98.2%

- ④ getChars demonstration

Original string = Welcome to Bmsce college.

- ⑤ Compare String

Bmce equals Bmce \rightarrow true

Bmce equals College \rightarrow false

Bmce equals BMSCE \rightarrow false

Bmce equals BMSCE \rightarrow true

Region Matches

substr1: Welcome to Bruce College

substr2: Bruce

true

Q9
Bruce starts with B and ends with T

true

false

Q11

Sort the given array:

apple ball cat dog cut free gun hen ice jug
kite lift man net orange parrot queen ring
star tree umbrella van watch Xmas match

Q10

Bmre.equals("Bruce") : true

Bmre == Bruce : false

Q12 Sorting 10 to 1

1 2 3 4 5 6 7 8 9 10

Q13 Replacing string.

Result: ~~This is a test. This is~~

This was a test. This was too.

This was a test. This was too.

This is a test. This was too.

This is a test. This was too.

This is a test. This is too.

⑭ Concatenation

hello + world = hello world.

hello.concat(world) = hello world.

⑮ Replace

"College".replace("Commegc") = Commegc

⑯ Initial : Hello World

Final : Hello world

~~SPS~~
16.01.24

Lab 7

```
IE/Student.java
package CIE;
import java.util.Scanner;
public class Student {
    protected String un= new String();
    protected String name= new String();
    protected int rnm;
    public void inputStudentDetails() {
        Scanner input= new Scanner(System.in);
        System.out.print("Enter un:");
        un= input.nextLine();
        System.out.print("Enter name:");
        name= input.nextLine();
    }
}
```

```
public void displayDetails() {
    System.out.println("Name:" + name + "USN:" + un
        + "\nRm:" + rnm)
}
```

```
// CIE/internal.java.  
package CIE;  
  
public class internal extends student {  
    protected int marks[] = new int[5];  
  
    public void input(CIEMarks()) {  
        Scanner input = new Scanner(System.in);  
  
        for (int i = 0; i < 5; i++) {  
            System.out.println("Enter marks of Subject" + (i + 1));  
            marks[i] = input.nextInt();  
        }  
    }  
}
```

SEE /~~internal~~ external.java

```
package SEE;  
import CIE.internal;  
import java.util.Scanner;  
  
public class external extends internal {  
    protected int mark[],  
    protected int fmarks[];  
  
    public external () {  
        this.marks = new int[5];  
        this.fmarks = new int[5];  
    }  
}
```

```
public void inputSEEmarks() {
    Scanner s = new Scanner (System.in);
    for (int i=0; i<5; i++) {
        System.out.print("Subject " + (i+1) + " marks: ");
        marks[i] = s.nextInt();
    }
}
```

```
public void calculateFinalMarks() {
    for (int i=0; i<5; i++) {
        finalMarks[i] = marks[i]/100 * super.marks[i];
    }
}
```

```
public void displayFinalMark() {
    displayStudentDetails();
    for (int i=0; i<5; i++) {
        System.out.println("Final Marks[" + i + "]");
    }
}
```

SEE/main.java

```
import SEE.externals;  
class main{  
    public static void main(String args[]){  
        int num=2;  
        external finalMarks[] = new external(num);  
        for(int i=0;i<num;i++){  
            final Main() = new externals();  
            final Marks[i].inputStudentDetails();  
            System.out.println("Enter CIE marks");  
            final Marks[i].inputCIEmarks();  
        }  
        System.out.println("Displaying data:\n");  
        for(int i=0;i<num;i++){  
            final Marks[i].calculateMarks();  
            final Marks[i].displayFinalMark();  
        }  
    }  
}
```

Output:

Student 1

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

Student 2

Enter marks of
 subject 1 : 30
 subject 2 : 70
 subject 3 : 60
 subject 4 : 80
 subject 5 : 90

Student 1

Enter marks of
 subject 1 : 20
 subject 2 : 40
 subject 3 : 30
 subject 4 : 10
 subject 5 : 0

8
23/01/2021

Student 2

Enter subject mark of
 subject 1 : 70
 subject 2 : 40
 subject 3 : 20
 subject 4 : 80
 subject 5 : 10

CIE:

Subj subg 1 marks : 30
 sub 2 marks : 50
 sub 3 marks : 40
 sub 4 marks : 20
 sub 5 marks : 10

SEE:

sub 1 marks : 30
 sub 2 marks : 20
 sub 3 marks : 60
 sub 4 marks : 80
 sub 5 marks : 90

CIE:

sub 1 marks : 20
 sub 2 marks : 40
 sub 3 marks : 30
 sub 4 marks : 10
 sub 5 marks : 0

SEE:

mb 1 marks : 70
mb 2 marks : 40
mb 3 marks : 20
mb 4 marks : 80
mb 5 marks : 10

02 shows 1 show
02 shows 2 show
03 shows 3 show
03 shows p show
01 shows 2 show

02 shows 1 show
03 shows 3 show
02 shows & show
03 shows & show
03 shows 2 show

03 shows 1 show
04 shows 3 show
03 shows & show
01 shows p show
01 shows 2 show

Lab 8

// Main.Father.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 (Father < 0) {
            throw new WrongAge("Age cannot be < 0");
        }
    }
}
```

```
class Non extends Father {
    int NonAge;
    Non() throws WrongAge {
        super();
        System.out.println("Enter non age");
        NonAge = sc.nextInt();
        if (FatherAge < NonAge) {
            throw new WrongAge("Age cannot be ");
        }
    }
}
```

```
void display() {
    System.out.println("Non Age = " + NonAge);
}
}
```

```
}
```

```
class Father Main{
    public static void main (String [] args) {
        try {

```

```
            Father father = new Father(),

```

```
            Non non = new Non();

```

```
            father.display();

```

```
            non.display();

```

```
        } catch (WrongAge e) {

```

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

Output:

1) Enter father age: 10

10

Enter son age:

5

Father Age: 10

Son age: 5

2) Enter father age

-1

Age cannot be < 0

3) Enter father age:

14

Enter son age:

16

Age cannot be greater for son.

8
06/02/2021

Lab-9

```
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

chat program - 10

```
import java.lang.*;  
  
class A {  
    int n;  
    boolean valueSet = false;  
    synchronized int get() {  
        while (!valueSet) {  
            try {  
                System.out.println("\nConsumer waiting\n");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("Interrupted Exception");  
            }  
        }  
        valueSet = false;  
        System.out.println("\nIntimate Producer\n");  
        notif();  
        return n;  
    }  
}
```

```
synchronized void put(int n){  
    while (valueSet){  
        try {  
            System.out.println("In Producer waiting");  
            wait();  
        } catch (InterruptedException e) {  
            System.out.println("Interrupted Exception caught");  
        }  
        this.n = n;  
        valueSet = true;  
        notify();  
    }  
}
```

```
class Consumer implements Runnable{  
    Queue<Integer> q;  
    Consumer(Q q){  
        this.q = q;  
        new Thread(this).start();  
    }  
    public void run(){  
        int i = 0;  
        while (i < 15) {  
            int r = q.get();  
            System.out.println("Consumed : " + r);  
            i++;  
        }  
    }  
}
```

```
class PC_Fixed {
    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:

Main thread entered A.foo
Racing thread entered B.bar
Main thread trying to call B.start()
Inside A.start
Back in main thread
Inside A.start
Back in other thread.

8
6 | 0 | 2 | 2n

Lab 10

1) AwingDemo():

The constructor initializes

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

```
class AwingDemo{
```

```
AwingDemo(){
```

```
    JFrame jfrm = new JFrame("Divide App");
    jfrm.setSize(275, 150);
    jfrm.setLayout(new FlowLayout());
```

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

```
JLabel plab = new JLabel("Enter the divisor and  
dividend:");
```

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

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

```
JLabel atab = new JLabel();
```

```
JLabel amulab = new JLabel();
```

```
jfm.add(cw);
jfm.add(jtf);
jfm.add(ljf);
jfm.add(button);
jfm.add(alab);
jfm.add(blab);
jfm.add(amlab);
```

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

```
try {
    int a = Integer.parseInt(tf1.getText());
    int b = Integer.parseInt(tf2.getText());
    int ans = a/b;
    alab.setText("nA = " + a);
    blab.setText("nB = " + b);
    amlab.setText("nAns = " + ans);
}
```

```
} catch (NumberFormatException e) {
    alab.setText("Text: " + "");
    blab.setText("Text: " + "a");
    amlab.setText("Text: " + " ");
    cw.setText("Text: " + "Enter only Integers!");
}
```

catch (ArithmeticalException e) {

abab -> Tcat (Text: "u").

bab -> Tcat (Text: "u").

amab -> Tcat (Text: "u").

er -> Tcat (Text: "B should be NON zero")

}

}

});

System.out.println(b: true);

}

public static void main(String args[]) {

Arithmetics invokeLater(new Runnable() {

public void run() {

new ArithExample();

});

}

}

Output

Divide A/b - □ X

Enter the divisor and dividend

Calculate

→ Report on functions used:

1) `MyingDemo()`:

This constructor initializes a JFrame with a size of 275x150 and sets the layout to Flow Layout and handles the window closing event.

2) `ajtf.addActionListener(l)`:

Adds the action listener to the JTextField ajtf, allowing it to respond to user input events.

3) `bjtf.addActionListener(l)`:

Adds the ActionListener to the JTextField bjtf, enabling it to handle user input events.

4) `button.addActionListener()`:

"calculate" button's ActionListener performs division based on user-input integers, updates JLabels with relevant information.

7) `JLabel jlabs`:

Displays the instruction label

"Enter the divisor and dividend".

8) `JTextField ajtf, bjtf`:

Text fields for user input of divisor and dividend, each set with a size of 8 characters.

- 9) JButton button.
Represents the "Calculate" button, triggering the division calculation
- 10) ~~SwingUtilities.invokeLater()~~:
Ensures that swing components are accessed on the event dispatching thread for thread safety
- ~~JDe
JPS
20.02.23~~

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 getCoefficients() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the coefficients of a, b, c:");
        a = scanner.nextInt();
        b = scanner.nextInt();
        c = scanner.nextInt();
        scanner.close();
    }

    void computeRoots() {
        while (a == 0) {
            System.out.println("Not a quadratic equation");
            System.out.println("Enter a non-zero value for a:");
            Scanner scanner = new Scanner(System.in);
            a = scanner.nextInt();
            scanner.close();
        }

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

        if (d == 0) {
            r1 = -b / (2.0 * a);
            System.out.println("Roots are real and equal");
            System.out.println("Root1 = Root2 = " + r1);
        } else if (d > 0) {
            r1 = (-b + Math.sqrt(d)) / (2.0 * a);
            r2 = (-b - Math.sqrt(d)) / (2.0 * 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 / (2.0 * a);
        r2 = Math.sqrt(-d) / (2.0 * a);
        System.out.println("Root1 = " + r1 + " + i" + r2);
        System.out.println("Root2 = " + r1 + " - i" + r2);
    }
}

class QuadraticMain {
    public static void main(String args[]) {
        Quadratic quadraticEquation = new Quadratic();
        quadraticEquation.getCoefficients();
        quadraticEquation.computeRoots();
    }
}

```

Output:

```

PS C:\Users\rabhi\OneDrive\Desktop\Java> cd "c:\Users\rabhi\OneDrive\Desktop\Java\lab\"  

; if ($?) { javac QuadraticMain.java } ; if ($?) { java QuadraticMain }  

Enter the coefficients of a, b, c:  

1 2 3  

Roots are imaginary  

Root1 = -1.0 + i1.4142135623730951  

Root2 = -1.0 - i1.4142135623730951  

PS C:\Users\rabhi\OneDrive\Desktop\Java\lab>

```

Lab 2

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

```
import java.util.Scanner;

class student{
    String usn,name;
    int n;
    String subject[]={};
    int credits[]={};
    int marks[]={};

    void read(){
        Scanner input= new Scanner(System.in);

        System.out.print("Enter your name:");
        name=input.nextLine();

        System.out.print("\n");
        System.out.print("Enter your USN:");
        usn=input.nextLine();
        System.out.print("\n");

        System.out.print("Enter number of subjects:");
        n=input.nextInt();

        input.nextLine();

        System.out.print("\n");
        for(int i=0;i<n;i++){
            System.out.println("-----");
            System.out.print("Enter subject:");
            subject[i]=input.nextLine();

            System.out.print("\n");
        }
    }

    void display(){
        System.out.print("-----");
        System.out.print("Name: "+name);
        System.out.print("USN: "+usn);
        System.out.print("Number of subjects: "+n);
        System.out.print("Subject: "+subject[0]);
        System.out.print("Marks: "+marks[0]);
        System.out.print("SGPA: "+(float)credits[0]/marks[0]);
    }
}
```

```
        System.out.print("Enter credits:");
        credits[i]=input.nextInt();

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

        System.out.print("Enter Marks:");
        marks[i]=input.nextInt();

        System.out.print("\n");
        System.out.println("-----");
        input.nextLine();

    }

}

void display(){

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

        System.out.println("Name:"+name);
        System.out.println("USN:"+usn+"\n");

        System.out.println("Subject\tCredits\tMarks");
        System.out.println("-----");
        for(int i=0;i<n;i++){

System.out.println(subject[i]+\t"+credits[i]+\t"+marks[i]);
        }

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

int grade(int a){

    while(a/10!=0){
```

```
        a=a/10;
    }
    if((a+1)>5){
        return (a+1);
    }
    else if((a+1)==5){
        return a;
    }
    else{return 0;}
}

void calculate(){

    int total_credits=0;
    int marks_credits=0;

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

        total_credits+=credits[i];
        marks_credits+=grade(marks[i])*credits[i];

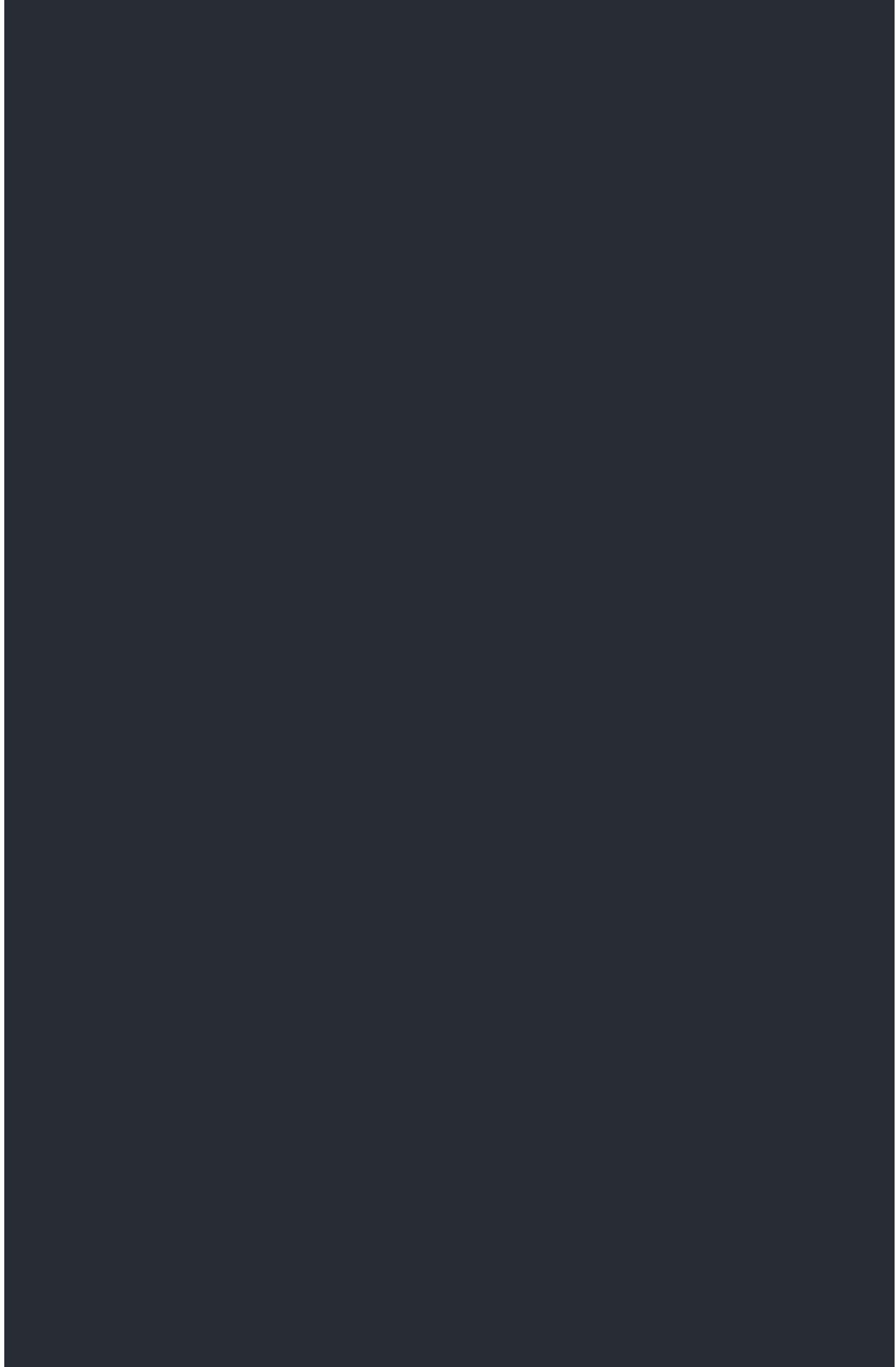
    }

    System.out.println("SGPA is
"+(double)marks_credits/total_credits);
}
}

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

        student calc= new student();
        calc.read();
        calc.display();
        calc.calculate();

    }
}
```



```

-----
Name:Abhinav
USN:1BM22CS211

Subject Credits Marks
-----
Math      4      98
Java      4      89
DBMS      3      99
COA       3      88
Kannada   1      98
-----
SGPA is 9.533333333333333
PS C:\Users\bmsce\Desktop\1BM22CS211\Java>

```

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 books{
    String author,book;
    int price,numPages;

    books(String book,String author, int price, int numPages){
        this.book=book;
        this.author=author;
        this.price=price;
        this.numPages=numPages;
    }

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

}

class lib{

```

```
public static void main(String[] args){
    String author,book;
    int price,numPages;

    int num_book;

    Scanner input=new Scanner(System.in);

    System.out.print("Enter number of books:");
    num_book=input.nextInt();
    System.out.print("\n");

    books b[]={new books[num_book]};

    for(int i=0;i<num_book;i++){
        System.out.println("-----\nBook "+(i+1)+"\n-----");
        input.nextLine();
        System.out.print("Enter the name of book:");
        book=input.nextLine();

        System.out.print("Enter Authors name:");
        author=input.nextLine();

        System.out.print("Enter the price:");
        price=input.nextInt();

        System.out.print("Enter the number of pages:");
        numPages=input.nextInt();

        b[i]=new books(book,author,price,numPages);
    }

    for(int i=0;i<num_book;i++){
        System.out.println("-----\nBook "+(i+1)+" details\n-----");
    }
}
```

```
        String a=b[i].toString();
        System.out.println(a);

    }

}
```

Output:

```
-----  
Book 1  
-----  
Enter the name of book:Harry Potter  
Enter Authors name:JK Rowling  
Enter the price:799  
Enter the number of pages:889  
-----  
Book 2  
-----  
Enter the name of book:Percy Jackson  
Enter Authors name:Rick Riordan  
Enter the price:689  
Enter the number of pages:459  
-----  
Book 3  
-----  
Enter the name of book:Goosebumps  
Enter Authors name:RL Stine  
Enter the price:450  
Enter the number of pages:236  
-----  
Book 1 details  
-----  
Book Name:Harry Potter  
Author:JK Rowling  
Price:799  
Number of pages:889  
-----  
Book 2 details  
-----  
Book Name:Percy Jackson  
Author:Rick Riordan  
Price:689  
Number of pages:459  
-----  
Book 3 details  
-----  
Book Name:Goosebumps  
Author:RL Stine  
Price:450  
Number of pages:236
```

Lab 4

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

```
import java.util.Scanner;

class inputScanner{
    void rec(rectangle ab){
        Scanner input=new Scanner(System.in);
        System.out.println("Enter the dimensions of the
rectangle(Length and Breadth):");
        ab.a=input.nextInt();
        ab.b=input.nextInt();
    }

    void tri(triangle ab){

        Scanner input=new Scanner(System.in);
        System.out.println("Enter the dimensions of the triangle(base
and height):");
        ab.a=input.nextInt();
        ab.b=input.nextInt();
    }

    void cir(circle ab){
        Scanner input=new Scanner(System.in);
        System.out.println("Enter the dimension of the
circle(radius):");
        ab.a=input.nextInt();
    }

}

abstract class shape extends inputScanner{
    int a,b;
    abstract void printArea();
}
```

```
class rectangle extends shape{
    rectangle(){
        rec(this);
    }
    void printArea(){
        System.out.println("Area of Rectangle = "+(double)(a*b));
    }
}

class triangle extends shape{

    triangle(){tri(this);}
    void printArea(){
        System.out.println("Area of Triangle = "+(0.5*a*b));
    }
}

class circle extends shape{
    circle(){cir(this);}
    void printArea(){
        System.out.println("Area of Circle = "+(3.14*a*a));
    }
}

class calc{
    public static void main(String[] args){
        rectangle r=new rectangle();
        triangle t=new triangle();
        circle c=new circle();

        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

Output:

```
PS C:\Users\bmsce\Desktop\1BM22CS211\Java\Lab 4> java calc
Enter the dimensions of the rectangle(Length and Breadth):
4 5
Enter the dimensions of the triangle(base and height):
9 8
Enter the dimension of the circle(radius):
4
Area of Rectangle = 20.0
Area of Triangle = 36.0
Area of Circle = 50.24
PS C:\Users\bmsce\Desktop\1BM22CS211\Java\Lab 4>
```

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;

class Account {
    String name;
    int accountNumber;
    double balance = 0;
    Scanner scanner = new Scanner(System.in);

    Account(String accountType) {
        System.out.println("Creating a new " + accountType + " account");
        System.out.print("Enter name: ");
```

```
this.name = scanner.next();
System.out.print("Enter account number: ");
this.accountNumber = scanner.nextInt();
}

void deposit() {
    System.out.print("Enter deposit amount: ");
    balance += scanner.nextDouble();
    System.out.println("Deposit successful. Current balance: " +
balance);
}

void withdraw() {
    System.out.print("Enter withdrawal amount: ");
    double withdrawal = scanner.nextDouble();
    if (withdrawal <= balance) {
        balance -= withdrawal;
        System.out.println("Withdrawal successful. Current balance: " +
balance);
    } else {
        System.out.println("Insufficient funds for withdrawal");
    }
}

void displayBalance() {
    System.out.println("Current balance: " + balance);
}

void performMenuActions() {
    int choice;
    do {
        System.out.println("\n----- MENU -----");
        System.out.println("1. Deposit");
        System.out.println("2. Withdraw");
        System.out.println("3. Display Balance");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();

        switch (choice) {
            case 1:
                deposit();
                break;
        }
    }
}
```

```
        case 2:
            withdraw();
            break;
        case 3:
            displayBalance();
            break;
        case 4:
            System.out.println("Exiting the menu. Thank you!");
            break;
        default:
            System.out.println("Invalid choice. Please try
again.");
    }
}

class SavingsAccount extends Account {
    int interestRate = 5;

    SavingsAccount() {
        super("Savings");
    }

    void compoundInterest(int time) {
        balance *= Math.pow((1 + (interestRate / 100.0)), time);
        System.out.println("Compound interest applied. Current balance: " +
balance);
    }
}

class CurrentAccount extends Account {
    double overdraftLimit = -100;

    CurrentAccount() {
        super("Current");
    }

    void issueCheque() {
        System.out.print("Enter cheque amount: ");
        double chequeAmount = scanner.nextDouble();
        if (chequeAmount <= balance && (balance - chequeAmount) >=
overdraftLimit) {
```

```
        balance -= chequeAmount;
        System.out.println("Cheque issued successfully. Current
balance: " + balance);
    } else {
        System.out.println("Insufficient funds to issue the cheque");
    }
}

public class bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Welcome to the Banking App");

        SavingsAccount savingsAccount = new SavingsAccount();
        CurrentAccount currentAccount = new CurrentAccount();

        Account selectedAccount = null;

        System.out.println("\nSelect an account type:");
        System.out.println("1. Savings Account");
        System.out.println("2. Current Account");
        int accountTypeChoice = scanner.nextInt();

        if (accountTypeChoice == 1) {
            selectedAccount = savingsAccount;
        } else if (accountTypeChoice == 2) {
            selectedAccount = currentAccount;
        } else {
            System.out.println("Invalid choice. Exiting.");
            System.exit(0);
        }

        selectedAccount.performMenuActions();

        scanner.close();
    }
}
```

Output:

```
Welcome to the Banking App
Creating a new Savings account
Enter name: Smith
Enter account number: 1
Creating a new Current account
Enter name: John
Enter account number: 2

Select an account type:
1. Savings Account
2. Current Account
1

----- MENU -----
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 1
Enter deposit amount: 8000
Deposit successful. Current balance: 8000.0

----- MENU -----
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 2
Enter withdrawal amount: 700
Withdrawal successful. Current balance: 7300.0

----- MENU -----
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 3
Current balance: 7300.0
```

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.

CIE/Student.java

```
package CIE;

public class Student {
    public String usn;
    public String name;
    public int sem;

    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

// CIE/Internals.java
package CIE;

public class Internals extends Student {
    public int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks)
    {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}
```

SEE/External.java

```
package SEE;

import CIE.Student;

public class External extends Student {
    public int[] seeMarks;

    public External(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
}
```

FinalMarks.java

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

public class FinalMarks {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of students: ");
        int n = scanner.nextInt();

        CIE.Internals[] cieStudents = new CIE.Internals[n];
        SEE.External[] seeStudents = new SEE.External[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for CIE of student " + (i + 1));
            cieStudents[i] = new CIE.Internals(usn, name, sem, cieMarks);
        }

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

```
        System.out.println("Enter details for SEE of student " + (i + 1));

        seeStudents[i] = new SEE.External(usrn, name, sem, seeMarks);
    }

    System.out.println("\nFinal Marks of Students:");
    for (int i = 0; i < n; i++) {
        System.out.println("\nDetails of Student " + (i + 1));

    }
}
}
```

Output

```
PS C:\Users\joshi\OneDrive\Desktop> javac SGPA.java
PS C:\Users\joshi\OneDrive\Desktop> java SGPA
Enter student usn:
146
Enter student name:
Manjari
Enter marks in order of credits
97 97 95 89 87 90 92 98
Enter order of credits
4 4 3 3 3 1 1 1
SGPA is : 9.0
```

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 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 < 0 for a person");
        }
    }
    void display(){
        System.out.println("Father Age = " + FatherAge);
    }
}
class Son extends Father{
    int SonAge;
    Son() throws WrongAge{
        super();
        System.out.println("Enter son age: ");
```

```

SonAge = sc.nextInt();
if (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 agemain{
    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:

```

PS C:\Users\rabhi\OneDrive\Desktop\Java> cd "c:\Users\rabhi\OneDrive\Desktop\Java\lab\" ;
if ($?) { javac Main.java } ; if (?) { java Main }
Enter father's age: 54
Enter son's age: 21
Father's age: 54
Son's age: 21
PS C:\Users\rabhi\OneDrive\Desktop\Java\lab> cd "c:\Users\rabhi\OneDrive\Desktop\Java\lab \
\" ; if (?) { javac Main.java } ; if (?) { java Main }
Enter father's age: 21
Enter son's age: 54
Exception caught: WrongAge: Son's age cannot be greater than or equal to father's age
Exception caught: Son's age cannot be greater than or equal to father's age
PS C:\Users\rabhi\OneDrive\Desktop\Java\lab> []

```

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 bois
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;

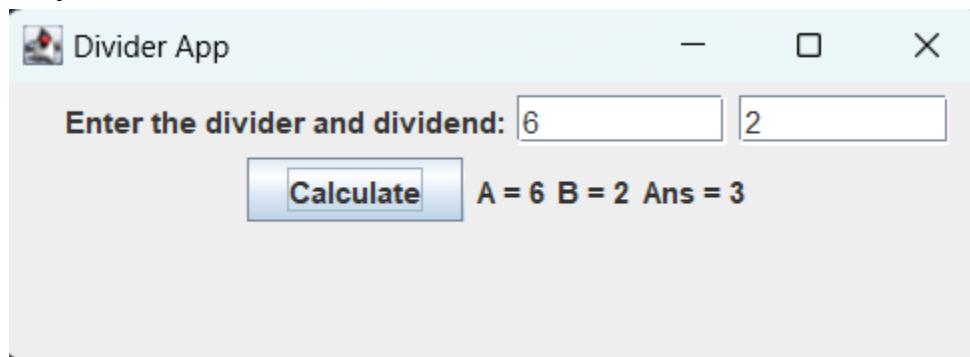
                alab.setText("\nA = " + a);
                blab.setText("\nB = " + b);
                anslab.setText("\nAns = " + ans);
            } catch (NumberFormatException e) {
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("Enter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

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

```

PS C:\Users\rabhi\OneDrive\Desktop\Java> cd "c:\"
ava Deadlock }
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
Inside A.last
RacingThread trying to call A.last()
Back in main thread
Inside A.last
Back in other thread
PS C:\Users\rabhi\OneDrive\Desktop\Java\lab>

```

```
procon.java
```

```
import java.lang.*;  
  
class Q {  
  
    int n;  
  
    boolean valueSet = false;  
  
    synchronized int get() {  
  
        while (!valueSet) {  
            try {  
                System.out.println("\nConsumer waiting\n");  
                wait();  
  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
            System.out.println("Got: " + n);  
            valueSet = false;  
            System.out.println("\nIntimate Producer\n");  
            notify();  
        }  
  
        return n;  
    }  
  
    synchronized void put(int n) {  
  
        while (valueSet) {  
            try {  
                System.out.println("\nProducer waiting\n");  
                wait();  
  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
            this.n = n;  
            valueSet = true;  
            System.out.println("Put: " + n);  
            System.out.println("\nIntimate Consumer\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 < 15) {
            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 < 15) {
            int r = q.get();
            System.out.println("consumed:" + r);
            i++;
        }
    }
}

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

```
    }
}
```

Output

```
PS C:\Users\rabhi\OneDrive\Desktop\Java\lab> javac procon.java
PS C:\Users\rabhi\OneDrive\Desktop\Java\lab> java procon
Put: 0

Intimate Consumer

Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer

Producer waiting

consumed:0
Got: 1

Intimate Producer

consumed:1
Put: 2

Intimate Consumer

Producer waiting

Got: 2

Intimate Producer

consumed:2
Put: 3

Intimate Consumer

Producer waiting

Got: 3

Intimate Producer

consumed:3
Put: 4

Intimate Consumer

Producer waiting

Got: 4

Intimate Producer

consumed:4
Put: 5

Intimate Consumer
```

```
Producer waiting
Got: 10

Intimate Producer
consumed:10
Put: 11

Intimate Consumer

Producer waiting
Got: 11

Intimate Producer
consumed:11
Put: 12

Intimate Consumer

Producer waiting
Got: 12

Intimate Producer
consumed:12
Put: 13

Intimate Consumer

Producer waiting
Got: 13

Intimate Producer
consumed:13
Put: 14

Intimate Consumer
Got: 14

Intimate Producer
consumed:14
PS C:\Users\rabhi\OneDrive\Desktop\Java\lab>
```