

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:

RANISHA GIRI (1BM22CS218)

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

INDEX

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

Q1:-

1) import java.util.*;
 class demo
 {
 public static void main (String args [])
 {
 System.out.println ("Hello World\n");
 }
 }

Output :-

Hello World.

2) class RectangleArea{

public static void main (String args []) {

int length , breadth ;

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

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

int area = length * breadth ;

System.out.println ("length of rectangle = " + length);

System.out.println ("breadth of rectangle = " + breadth);

System.out.println ("area of rectangle = " + area);

System.out.println ("Ranisha 1BM22CS218 ");

Output :-

jinae RectangleArea.jinae

jina Rectangle Area 105

length of rectangle = 10

breadth of rectangle = 5

area of rectangle = 50

Ranisha 1BM22CS218

3) Factorial:-

```

import java.util.*;
class factorial {
    public static void main (String args[]) {
        int fac = 1;
        System.out.println ("Enter a number :");
        Scanner sc = new Scanner (System.in);
        int n = sc.nextInt();
        for (int i = 1; i <= n; i++) {
            fac = fac * i;
        }
        System.out.println ("The factorial is :" + fac);
    }
}

```

Output :-

Enter a number:

6

The factorial is :

720

Ranisha 1Bm22C028

4) Palindrome:-

```

import java.util.*;
class palindrome {
    public static void main (String args[])
    {
        int n, t, rem, rev = 0;
        Scanner sc = new Scanner (System.in)

```

```
System.out.println ("Enter a 5 digit number :");
```

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

```
t = n;
```

```
while (t > 0) {
```

```
rem = t % 10;
```

```
rev = rev * 10 + rem;
```

```
t = t / 10;
```

```
}
```

```
if (rev == n) {
```

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

```
y
```

```
else {
```

```
System.out.println ("not palindrome");
```

```
}
```

```
y
```

```
.
```

Output :

Enter a 5 digit number:

12345

not palindrome

Ranisha 13M22CS216

5) Quadratic :-

```

import java.util.Scanner;
class Quadratic
{
    int a,b,c;
    double r1,r2,d;
    void get()
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the coefficients of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while (a == 0)
        {
            System.out.println ("Not a quadratic equation");
            System.out.println ("Enter a non zero value for a");
            Scanner s = new Scanner (System.in);
            a = s.nextInt();
        }
        d = b * b - 4 * a * c;
        if (a == 0)
        {
            r1 = (-b) / (2 * a);
            System.out.println ("Roots are real and equal");
            System.out.println ("Root1 = Root2 = " + r1);
        }
    }
}

```

```
- else if (d > 0)
```

{

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

```
    r2 = ((-b) - (math.sqrt(d))) / (double)(2*a);
```

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

```
    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 ("Root1 = " + r1 + " + i " + r2);
```

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

}

Y

Y

```
class QuadraticMain
```

{

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

{

```
        Quadratic q = new Quadratic();
```

```
        q.getd();
```

```
        q.compute();
```

Date

Y

Output :- Ranisha

1 BM22 CS218

Enter coefficient of a,b,c

$$a = 6$$

Roots are imaginary

$$\text{Root 1} = 0.0 + i 1.053268721$$

$$\text{Root 2} = 0.0 - i 1.053268721$$

Ranisha

1 BM22 CS218

Enter coefficient of a,b,c

$$1 \quad -2 \quad 1$$

Roots are real and equal

$$\text{Root 1} = \text{Root 2} = 1.0$$

Ranisha

1 BM22 CS218

Enter coefficient of a,b,c

$$1 \quad -3 \quad 2$$

Roots are real and distinct

$$\text{Root 1} = 2.0 \quad \text{Root 2} = 1.0$$

Additional programs in Operators:-

```
class modulus {
```

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

```
int x = 42;
```

```
double y = 42.25;
```

```
byte a = 64, b, b1;
```

```
int i;
```

```
System.out.println ("Ranisha Giri 18M22CS218");
```

```
i = a << 2;
```

```
b = (byte) (a << 2);
```

```
i = a << 2;
```

```
System.out.println ("Original value of a: " + a);
```

```
System.out.println ("i and b: " + i + " " + b);
```

```
int a1 = -1;
```

```
a1 = a1 >> 24;
```

```
System.out.println ("value of a1: " + a1);
```

```
a1 = -1;
```

```
a1 = a1 >>> 24;
```

```
System.out.println ("value of a1: " + a1);
```

```
y
```

Output:-

Ranisha Giri 18M22CS218

Original value of a : 64

i and b : 2560

value of a1 = -1

value of a1 : 255

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

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

 name = input.nextLine();

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

 usn = input.nextLine();

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

 System.out.println ("Enter " + subject[i]);

 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 ("Subject " + subject[i]);

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

}

void calculate SGPA () {

int total_credits = 0;

int marks_credits = 0;

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

{

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

}

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

Y
Y

Class marks {

public static (String I) args {

student s1 = new Student();

s1.read

s1.hi,play

11. calculate SGPA();

P
Y

Output :

Enter your name : Lanisha

Enter USN: 13M22CS218

Enter subject : Y

Enter subject : Math

Enter credit : 4

Enter marks : 96

Name : Lanisha
USN: 13M22CS 218

Subject	credit	marks
Math	4	96

SPPA is 9.4.

19/12/22

LAB 3 PROGRAMS:-

- Create class book containing : name, author, price, num-pages.
Include a constructor to set value for members. Include method
and string () method to display complete details of the book.
Develop a Java program

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

Class Book {

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

```
    public Book (String name, String author, int price, int numPages)
```

```
        this.name = name;
```

```
        this.author = author;
```

```
        this.price = price;
```

```
        this.numPages = numPages;
```

```
}
```

public String toString ()

```
    String bookDetails = "Book name :" + this.name + "\n";
```

```
    + "Author name :" + this.author + "\n";
```

```
    + "Price :" + this.price + "\n" + "Number of pages:"
```

```
    + this.numPages + "\n" + ;
```

```
    return bookDetails;
```

```
public class Main {
```

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

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

```
        System . out . print ( " Enter the number of books : " );
```

```
        int n = scanner . nextInt ();
```

```
        Book [] books = new Book [n];
```

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

```
        {
```

```
            System . out . println ( " Enter name of the book : - " );
```

```
            String name = scanner . next ();
```

```
            System . out . println ( " Enter author of book : \n " );
```

```
            String author = scanner . next ();
```

~~```
 System . out . println (" Enter price of book : ");
```~~~~```
            int price = scanner . nextInt ();
```~~

```
            System . out . println ( " Enter number of pages of  
                the book : " );
```

```
            int numPages = scanner . nextInt ();
```

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

```
        }
```

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

```
    {
```

```
        System . out . print ( " \n Book " + (i + 1) + " : \n " +  
            books [i] );
```

System.out.println ("Ranisha 15m22cs218");
})

OUTPUT:

Enter the number of books: 2

Enter name of the book: Harry Potter

Enter author of the book: JK Rowling

Enter the price of the book: 500

Enter the number of pages of the book: 600

Enter the name of the book: Jungle Book

Enter author of the book: Rukin Bond

Enter price of the book: 200

Enter the number of pages of the book: 300

Book details:

Book 1:

Book name: Harry Potter

Author name: JK Rowling

Price: 500

Number of pages: 600

Book 2:

Book Name: Jungle Book

Author name: Rukin Bond

Price: 200

Number of pages: 300

Ranisha Giri 15m22cs218

ADDITIONAL PROGRAM:-

```
class OverloadDemo {  
    void test() {  
        System.out.println ("No parameters");  
    }  
    void test (int a) {  
        System.out.println ("a:" + a);  
    }  
    void test (int a, int b) {  
        System.out.println ("a and b:" + a + " " + b);  
    }  
    double test (double a) {  
        System.out.println ("double a:" + a);  
        return a * a;  
    }  
}
```

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

```
        OverloadDemo ob = new OverloadDemo ();  
        double result;  
        ob.test ();  
        ob.test (10);  
        ob.test (10, 20);  
        result = ob.test (12.2);
```

```
        System.out.println ("ob.test (12.2) = " + result);  
    }
```

Output :-

no parameters

a: 10

a and b: 10 20

double a: 123.25

result ob.out(123.2): 15178.240000 ... 2

$\sqrt{6 \times 10^3}$

Lab 4 programs:

Java program to create an abstract class named Shape containing two integers and an empty method named printArea().

```
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 d2:");
```

```
            d1 = sc.nextInt();
```

```
}
```

```
else {
```

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

```
    d1 = sc.nextInt();
```

```
    d2 = sc.nextInt();
```

```
}
```

```
y
```

```
y
```

```
y
```

```
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);
```

```
}
```

```
y
```

```
class Rectangle extends Shape {
```

```

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

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

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

```

OUTPUT:-

Enter d1 and d2:

5

4

Enter d1 and d2:

5

3

Enter d1:

6

Area of rectangle is : 20.0

Area of triangle is : 7.5

Area of circle is 13.039999

Name: Ravinder Giri

USN: 18M22CS218

8/10/12
02/10/12

Lab 5 programs:

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

```
class account
```

```
{
```

```
String name;
```

```
int accno;
```

```
String type;
```

```
double balance;
```

```
account (String name, int accno, String type, double balance)
```

```
{
```

```
    this.name = name;
```

```
    this.accno = accno;
```

```
    this.type = type;
```

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

```
}
```

```
void deposit (double amount)
```

```
{
```

```
    balance = amount;
```

```
}
```

~~```
void withdraw (double amount)
```~~~~```
{
```~~~~```
 if ((balance - amount) >= 0)
```~~~~```
        balance = amount
```~~~~```
}
```~~~~```
System.out.println ("Insufficient balance, can't withdraw");
```~~~~```
}
```~~

void display()

{

System.out.println("Insufficient name: " + name + " account  
type: " + type " balance: " + balance);

}

class SavAcct extends Account

{

private static double rate = 5;

SavAcct (String name, int accno, double balance)

{

super (name, accno, "savings", balance);

}

void interest()

{

balance += "balance" \* rate / 100;

System.out.println("balance: " + balance);

}

class CurAcct extends Account

{

private double minBal = 500;

private double serviceCharge = 50;

CurAcct (String name, int accno, double balance)

{

super (name, accno, "current", balance);

}

void checkMin()

{

If (balance < minBal).

}

System.out.println ("balance is less than min balance , service charges imposed : " + servicecharges);

balance -= servicecharges;

System.out.println ("balance is : " + balance);

y

y

y

class account Main

{

public static void main (String args) .

{

Scanner s = new Scanner (System.in);

System.out.println ("Enter the name : ");

String name = s.next();

System.out.println ("enter the type (current / savings) : ");

String type = s.next();

System.out.println ("enter the account number : ");

int accno = s.nextInt();

System.out.println ("Enter the initial balance : ");

double balance = s.nextDouble();

int ch;

double balans = s.nextDouble(); amt1, amt2;

Account acc = new Account (name, accno, type, balance);

SavAcct sa = new SavAcct (name, accno, balance);

currAct ca = curAct (name, accno, balance);

while (true) .

{ if (acc.type.equals ("savings"))

System.out.println ("\nMenu\n 1. Deposit\n 2. Withdraw

2. Compute interest. 4. Exit");

System.out.println("enter the choice : ");

ch = s.nextInt();

switch(ch) {

}

case 1: System.out.println("enter amt : ");

amt1 = s.nextInt();

sa.deposit(amt1);

break;

case 2: System.out.println("enter amt : ");

amt2 = s.nextInt();

sa.withdraw(amt2);

break;

case 3: sa.interest();

break;

default: System.out.println("invalid");

break;

4.

5.

else

6.

System.out.println("1-deposit 2-withdraw 3-display")

System.out.println("enter choice");

ch = s.nextInt();

switch(ch) {

case 1: System.out.println("enter amt : ");

amt1 = s.nextInt();

sa.deposit(amt1);

break;

case 2: System.out.println("enter amt : ");

amt2 = s.nextInt();

on withdraw (cont 2);

a. check (n C);

break;

case 3 : cl : play();

break;

default : System.out.println("invalid");

break;

}

}

}

Enter customer name : Ranisha Giri

Enter account number : 1

Enter initial balance : 45000

Select an option :

1. Deposit to Savings Account
2. Compute and Deposit Interest for Savings Account
3. Withdraw from Savings Account
4. Deposit to Current Account
5. Withdraw from Current Account
6. Display Balances
7. Exit

8

9/01/29

Enter your choice : 2

Amount deposited + Current balance : 48375.0

Interest computed and deposited + Current balance : 48375

Select an option :

1. Deposit to Savings Account
2. Compute and Deposit Interest for Savings Account
3. Withdraw from Savings Account
4. Deposit to Current Account
5. Withdraw from Current Account

## 6. Display Balances:

o. Exit

Enter your choice : 3

Enter the amount to withdraw from Savings Account : 4500.

Amount withdrawn successfully. Current balance : 3375.

Select an option :

1. Deposit to Savings Account
2. Compute and deposit Interest for savings Account
3. withdraw from Savings Account
4. Deposit to Current Account
5. Withdraw from Current Account
6. Display balances
0. Exit .

Enter your choice : 3

Enter the amount to withdraw from Savings Account : 3000.

Amount withdrawn successfully. Current balance : 375.

Select an option :

1. Deposit to Savings Account
2. Compute and deposit Interest for Savings Account
3. withdraw from Savings Account
4. Deposit to Current Account
5. Withdraw from Current Account
6. Display balances
0. Exit .

Enter your choice : 6

Savings Account details :

Account type : Savings

Fiyoshi Customer Name : Ramsha

Account Number : 1

## UAS 6 PROGRAMS:-

Page No.

Date 16/01/24

Current Balance : 45100

Q) Outputs :-

i) String length : 13

Are string literals equal? true

FULL NAME: John Doe

iii) ToString() :-

String representation using to & toString(): Student { name = 'John', age = 20 }

toString(): Student { name = 'John', age = 20 }

i) Extracted string is : Bruce

ii) String converted to byte array: Ranisha

String converted to char array: Ranisha

iii) Bruce equals Bruce → true

public class StringComparisonDemo {

    public static void main (String [] args)

        String bruce = "Bruce";

        String sameString = "Bruce";

        boolean isEqual = bruce.equals (sameString);

        System.out.println ("bruce" + "equals" + sameString + "→" +  
                            isEqual);

String different string = "College";

is equal = brace.equals (different string);

System.out.println (brace + "equals" + different string + "→",  
is equal);

String case sensitive comparison = "Bmsee";

is equal = brace.equals (case sensitive comparison);

System.out.println (brace + "equals" + case sensitive comparison,  
"→" + is equal);

}

}

OUTPUT:-

Bmsee equals Bmsee → True

Bmee equals College → false

Bmee equals Bmse → false

Bmee equals ignore case Bmse → true

7) Substring is matched

8) Does string start with "Hello"? true ]

Does string start with "hi"? false ]

→ [if Original  
string = "Hello,  
world".

9) Does string end with "World"? true ]

Does string end with "Jain"? false ]

10) Create two strings with same content

String s1 = new String ("Kanisha");

String s2 = new String ("Kanisha");

Create two strings with same reference:

String s3 = "OJ";

String s4 = s3;

Output :

using equals (): true

using == : false

using equals (): true

using == : true

ii) string [] words = { "van", "watch", "ball", "cat", "xmas",  
 "yatch", "zee", "apple", "ice", "kite", "lift", "man",  
 "net", "orange", "dog", "ent", "free", "grin", "hen",  
 "parrot", "green", "ring", "star", "tree", "umbrella" };

Output :

apple

ball

cat

dog

ent

free

grin

hen

ice

ring

tree

umbrella

van

watch

xmas

yatch

zee

star

parrot

green

dog

tree

umbrella

van

watch

xmas

yatch

zee

12) Integer [] numbers = { 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 };

```
Array sort (numbers);
System.out.println("Sorted Numbers (Descending orders):");
for (int i = number.length - 1; i >= 0; i--) {
 System.out.println(numbers[i]);
}
```

↳

Output:-

Sorted numbers (Descending numbers):

10 9 8 7 6 5 4 3 2 1

13) Original → First string: This is an ~~o~~ test. This is, too  
 Best final string: This is a test. This is, too

14) String s1 = "Hello";  
 String s2 = " ~~an~~ world";

Output:-

Concatenated string: Ranidip in 01 lab Hello world.

15) Original: Welcome to BMSCE College of Engineering  
 Modified: Welcome to BMSCE commage of engineering

80  
16/1/20

LAB TO PROGRAMS:-Student.java

```
package cie;
public class Student {
 public String name, uni;
 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() {
 for (int i = 0; i < 5; i++) {
 marks[i] = sc.nextInt();
 }
 }
}
```

```
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();
}
```

```
System.out.println("Subject " + (i + 1) + " marks" +
```

```
marks[i]);
```

```
public void display() {
 for (int i = 0; i < 5; i++) {
 System.out.println("Subject " + (i + 1) + " marks" +
```

```
marks[i]);
```

```
}
```

```
y.
```

futurals - jaina

package see ';

Import cie . Student ;

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

public class External extends Student {

public int maxs () = new int [5];

```
public void input marks () {
```

```
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();`

5

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

~~for (int i = 0; i < s; i++) {~~

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

۶

Main.java

```
import cie.Student;
import cie.Internals;
import cie.Externals;
import java.util.Scanner;
```

```
class Main {
 public static void main (String args []) {
 int no = 2;
 Externals finalMarks [] = new Externals [no];
 Internals intMarks [] = new Internals [no];
 for (int i=0; i < no; i++) {
 finalMarks [i] = new Externals ();
 intMarks [i] = new Internals ();
 finalMarks [i].inputMarks ();
 intMarks [i].inputMarks ();
 }
 for (int i=0; i < no; i++) {
 System.out.println ("CIE:");
 intMarks [i].displayMarks ();
 System.out.println ("SEE:");
 finalMarks [i].displayMarks ();
 }
 }
}
```

OUTPUT:-

Enter subject 1 marks : 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 : 70  
Enter subject 2 marks : 40  
Enter subject 3 " : 20  
Enter subject " : 80  
Enter subject " : 10

CIE

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

SEE:-

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

CIE:-

Subject 1 marks : 70

Subject 2 marks : 40

Subject 3 marks : 20

Subject 4 marks : 20

Subject 5 marks : 10

STF:-

Subject 1 marks : 80

Subject 2 marks : 60

Subject 3 marks : 30

Subject 4 marks : 20

Subject 5 marks : 30

8/1/20  
29/01/20

Lab 7 program:-

import java.util.Scanner;

class wrongAge extends RuntimeException {

    public wrongAge() {

        super ("Age cannot be negative");

}

    public wrongAge (String message) {

        super (message);

}

}

class InputScanner {

    protected Scanner scanner;

    public InputScanner () {

        protected scanner = new Scanner (System.in);

    public int nextInt () {

        return scanner.nextInt();

}

class Father extends InputScanner {

    protected int fatherAge;

    public Father () {

        System.out.print ("Enter father's age : ");

        fatherAge = super.nextInt();

    if (fatherAge < 0) {

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

}

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

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

{

```
}
```

```
class Son extends Father {
```

```
 private int sonAge;
```

```
 public Son () {
```

```
 super ();
```

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

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

```
 if (sonAge > fatherAge) {
```

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

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

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

{

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

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

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

{

```
public class InheritanceException {
```

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

```
 try {
```

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

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

{

catch (wrong Age e) ?

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

↳

↳

INPUT:

Enter father's age: 20

(wrong input)

Enter son's age: 40

Son's age cannot be greater than father's age.

Enter father's age: -10

Age cannot be negative

Enter father's age: 45

Enter son's age: do

Father's age: 45

Son's age: 20

8/01/20  
30

Lab 8 programs:-

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

class B extends Thread {

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

```
 for (int i=0; i<3; i++) {
```

```
 System.out.println("BMS");
 Thread.sleep(1000);
```

}

```
 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("CSF");
```

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

}

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

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

}

};

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

```
b = new B();
```

```
c = new C();
```

```
b.start();
```

```
c.start();
```

4

Output :-

BMS

CSE

CSE

CSE

BMS

BMS

class A {

synchronized void f (B b)

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

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

```
try {
```

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

```
}
```

```
(catch {Exception e})
```

```
{
```

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

```
 System.out.println (name + " trying to call a.int()");
```

```
b.start();
```

)

```
void last()
```

{

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

}

}

```
class B {
```

```
 synchronized void bus (A a)
```

{

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

```
 System.out.println (name + " enter B.bus");
```

try

{

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

}

```
 catch (Exception e)
```

{

```
 System.out.println ("B interrupted");
```

}

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

```
 a.last();
```

}

```
void last ()
```

{

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

}

```
class Deadlock implements
```

```
 A a = new A ();
```

```
 B b = new B ();
```

```
 Deadlock ()
```

{

```

Thread.currentThread().setName("Main Thread");
Thread t = new Thread(this, "Racing Thread");
t.start();
a.for(b);
System.out.println("Back in main thread");
}

public void run() {
 b.bor(a);
 System.out.println("Back in main thread");
}

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

```

Output :-

Main Thread entered A.for

Racing Thread entered B.bor

Main Thread trying to call B.last()

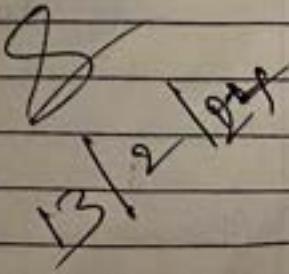
Inside A.last

Back in main thread

Racing thread trying to call A.last()

Inside A.last

Back in other thread



Lab 9 programs:-

Thread main:-

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

Class B extends Thread :-

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

```
try {
```

```
for (int i=0; i<3; i++) {
```

```
System.out.println ("BMS");
Thread.sleep (1000);
```

```
} 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 ("CE");
Thread.sleep (2000);
```

```
}
```

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

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

```
}
```

Class Threadmain :-

```
public class Threadmain {
```

```

}
Bb = new B();
C c = new C();
b.start();
c.start();
}

```

Deadlock Java :-

Class A {

```

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

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

```

Class B {

synchronized void bar(A a).

```

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

```

```

try {
 Thread.sleep(1000);
} catch (Exception e) {
 System.out.println("B interrupted");
}

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

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

Class Deadline implements Runnable {
 A a = new A();
 B b = new B();

 Deadlock() {
 Thread.currentThread().setName("Main Thread");
 Thread t = new Thread(this, "Racing Thread");
 t.start();
 a.foo(b);
 S.O.P("Break in thread");
 }

 new Deadlock();
}

```

Output :-

Main Thread entered A.foo

Racing Thread entered B.bar

Main Thread trying to call B.last()

Inside A.last

Break in Main Thread

Racing Thread trying to call A.last()

Inside A.last

Break in other thread

Pro. com Tarea

class A {

int n;

boolean value set = false;

synchronized int get () {

while (!valueset) {

try {

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

wait ();

} catch (InterruptedException e) {

S. O. Pm (e);

}

S. O. Pm (" : " + n);

value set = true;

S. O. Pm (" Tell producer ");

notify ();

y

return n;

y

class Producer implements Runnable

Q q;

Producer (Q q) {

this. q = q;

new Thread (this. "Producer"). start();

public void () {

int i = 0;

while (i < 3) {

q. put (i++);

y y

Class consumer implements Runnable &

Q q<sub>i</sub>

consumer (Q q) {

this.q = q<sub>i</sub>

new Thread (this, "Consumer").start();

y

public void run () {

int i = 0;

while (i < 3) {

int n = q.get();

System.out.println("Consumed : " + r);

i++;

}

Output:

Put : 1

get : 1

Put : 2

get : 2

Put : 3

get : 3

Name: Ramisha Singh (218)

8 / 2 / 2020  
13

Lab 10 programs:-

```
import java.awt.*;
import java.awt.event.*;
import java.awt.WindowEvent;
```

class SwingingDemo {

    SwingingDemo() {

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

```
 jfrm.setSize(275, 250);
```

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

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

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

```
 JTextField aJtf = new JTextField(8);
```

```
 JTextField bJtf = new JTextField(8);
```

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

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

```
 JLabel stats = new JLabel();
```

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

```
 JLabel anslab = new JLabel();
```

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

```
 jfrm.add(jlab);
```

```
 jfrm.add(aJtf);
```

```
 jfrm.add(bJtf);
```

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

```
 jfrm.add(stats);
```

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

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

```
Action listener 1 = new ActionListener() {
 public void actionPerformed (ActionEvent evt) {
 System.out.println ("Action event from a text field");
 }
};

aJTF.addActionListener (1);
bJTF.addActionListener (2);

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

 aLab.setText ("\nA = " + a);
 bLab.setText ("\nB = " + b);
 ansLab.setText ("\nAns = " + ans);
 }
 catch (NumberFormatException e) {
 aLab.setText ("");
 bLab.setText ("");
 ansLab.setText ("");
 err.setText ("Enter only integers!");
 }
 catch (ArithmaticException e) {
 aLab.setText ("");
 bLab.setText ("");
 ansLab.setText ("");
 err.setText ("B should be non zero");
 }
 }
});
```

11)

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

Output:-

Enter the divisor and dividend:

25 5 .

$$\text{Calculate } A = 25 \quad B = 5 \quad \text{and } A_{\text{avg}} = 5 \dots$$

JFrame: is a class in Java Swing that represents a top-level container for a GUI application.

`setSize()`: used to set size of a JFrame in pixels.

setLayout(): this method is used to layout manager for the JFrame.

`setDefaultCloseOperation()`: to set the default close operation for the JFrame.

Label : is a swing component used to display a non-editable text.

~~Textfield~~: swing component that allows the user to enter text and click a single line or text.

Add (frame): method used to add components to a container

`addActionListener()`: method used with components like buttons  
to register an action listener

`setText()`: used with components like JTextField or JLabel to set the text content

~~100% 100% 100%~~

# Lab 1 programs

BY RANISHA GIRI (1BM22CS218)

## 1)Quadratic:

```
import java.util.Scanner;
class Quadratic
{
 int a, b, c;
 double r1, r2, d;
 void getd()
 {
 Scanner s = new Scanner(System.in);
 System.out.println("Enter the coefficients of a,b,c");
 a = s.nextInt();
 b = s.nextInt();
 c = s.nextInt();
 }
 void compute()
 {
 while(a==0)
 {
 System.out.println("Not a quadratic equation");
 System.out.println("Enter a non zero value for a:");
 Scanner s = new Scanner(System.in);
 a = s.nextInt();
 }
 d = b*b-4*a*c;
 if(d==0)
 {
 r1 = (-b)/(2*a);
 System.out.println("Roots are real and equal");
 System.out.println("Root1 = Root2 = " + r1);
 }
 else if(d>0)
 {
 r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
 r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
 System.out.println("Roots are real and distinct");
 System.out.println("Root1 = " + r1 + " Root2 = " + r2);
 }
 }
}
```

```

 }
 else if(d<0)
 {
 System.out.println("Roots are imaginary");
 r1 = (-b)/(2*a);
 r2 = Math.sqrt(-d)/(2*a);
 System.out.println("Root1 = " + r1 + "+ i"+r2);
 System.out.println("Root1 = " + r1 + " - i"+r2);
 }

}

class QuadraticMain
{
 public static void main(String args[])
 {
 Quadratic q = new Quadratic();
 q.getd();
 q.compute();
 }
}

```

2)additional code for lab 1:

```

class Modulus {
public static void main(String args[]) {
int x = 42;
double y = 42.25;

byte a = 64, b,b1;
int i;
i = a << 2;
b = (byte) (a << 2);
i = a << 2;

System.out.println("Original value of a:" + a);
System.out.println(" and b:" + i + " " + b);

int a1=-1;
a1=a1>>24;
System.out.println("value of a1: " + a1);

a1=-1;

```

```
a1=a1>>>24;
System.out.println("value of a1: " + a1);

}
}
```

## LAB 2 PROGRAMS:

### SGPA CALCULATOR:

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

```
class student{
 String usn,name;
 int n;
 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.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");

 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();

 }
}

```

## LAB 3 PROGRAM:

### Book details:

```
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);

}
}

```

## LAB 4 PROGRAMS:

### Area of shapes:

```

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(" Ranisha Giri 1BM22CS218");
 Rectangle r = new Rectangle();
 Triangle tr = new Triangle();
 Circle c = new Circle();
 r.printArea();
 }
}

```

```

 tr.printArea();
 c.printArea();
 }
}

```

## LAB 5 PROGRAMS:

### Bank details:

```

import java.util.Scanner;
class account
{
 String name;
 int accno;
 String type;
 double balance;

 account(String name,int accno,String type,double balance)
 {
 this.name=name;
 this.accno=accno;
 this.type=type;
 this.balance=balance;
 }
 void deposit(double amount)
 {
 balance+=amount;
 }
 void withdraw(double amount)
 {
 if((balance-amount)>=0)
 {
 balance-=amount;
 }
 else
 {
 System.out.println("insufficient balance,cant withdraw");
 }
 }

 void display()
 {

```

```

 System.out.println("name:"+name+" accno:"+accno+" type:"+type+
balance:"+balance);
 }
}
class savAcct extends account
{

 private static double rate=5;
 savAcct(String name,int accno,double balance)
 {
 super(name,accno,"savings",balance);

 }

 void interest()
 {
 balance+=balance*(rate)/100;
 System.out.println("balance:"+balance);
 }

}

class curAcct extends account
{

 private double minBal=500;
 private double serviceCharges=50;

 curAcct(String name,int accno,double balance)
 {
 super(name,accno,"current",balance);

 }

 void checkmin()
 {

 if(balance<minBal)
 {
 System.out.println("balance is less than min balance,service charges
imposed:"+serviceCharges);
 balance-=serviceCharges;
 System.out.println("balance is:"+balance);
 }
 }
}

```

```

 }

 }

}

class accountMain
{
 public static void main(String a[])
 {
 Scanner s=new Scanner(System.in);
 System.out.println("enter the name :");
 String name=s.next();
 System.out.println("enter the type(current/savings):");
 String type=s.next();
 System.out.println("enter the account number:");
 int accno=s.nextInt();
 System.out.println("enter the intial balance:");
 double balance=s.nextDouble();
 int ch;
 double amount1,amount2;
 account acc=new account(name,accno,type,balance);
 savAcct sa=new savAcct(name,accno,balance);
 curAcct ca=new curAcct(name,accno,balance);
 while(true)
 {
 if(acc.type.equals("savings"))
 {
 System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute
interest 4.display 5.exit");
 System.out.println("enter the choice:");
 ch=s.nextInt();
 switch(ch)
 {
 case 1:System.out.println("enter the amount:");
 amount1=s.nextInt();
 sa.deposit(amount1);
 break;
 case 2:System.out.println("enter the amount:");
 amount2=s.nextInt();
 sa.withdraw(amount2);
 break;
 case 3:sa.interest();
 break;
 case 4:sa.display();
 }
 }
 }
 }
}
```

```
 break;
 case 5:System.exit(0);
 default:System.out.println("invalid input");
 break;
 }
}
else
{
 System.out.println("\nMenu\n1.deposit 2.withdraw 3.display
4.exit");
 System.out.println("enter the choice:");
 ch=s.nextInt();
 switch(ch)
 {
 case 1:System.out.println("enter the amount:");
 amount1=s.nextInt();
 ca.deposit(amount1);
 break;
 case 2:System.out.println("enter the amount:");
 amount2=s.nextInt();
 ca.withdraw(amount2);
 ca.checkmin();
 break;

 case 3:ca.display();
 break;
 case 4:System.exit(0);
 default:System.out.println("invalid input");
 break;
 }
}
}

}
```

# LAB 6 PROGRAMS:

## OUTPUT OF 20 PROGRAMS GIVEN BY PROF:

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

```

```

System.out.println("Original string: " + originalString);
System.out.println("Replaced string: " + replacedString);

/*14*/ System.out.println("Concatenation example: hello + world = " +
"hello".concat("world"));

/*15*/ System.out.println("Replacing ll with mm gives us: " + "College".replace("ll", "mm"));

/*16*/ System.out.println("Trimming example: " + " Hello World ".trim());

/*18*/
StringBuffer exp = new StringBuffer("Something");
exp.setLength(10);
char[] buffer = new char[4]; exp.getChars(0, 4, buffer, 0); String buf = new String(buffer);
System.out.println("Length: " + exp.length());
System.out.println(exp.charAt(0) + " " + buf + " " + exp.reverse() + " " + exp.substring(0, 3));
exp.reverse();
exp.append('c'); exp.insert(0, 'K'); exp.replace(0, 1, "L");
System.out.println(exp);
exp.deleteCharAt(0); exp.delete(0, 4);
System.out.println(exp);
}

}

```

## LAB 7 PROGRAM:

Students CIE & SEE marks:

Student.java:

```

package cie;

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

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]);
 }
 }
}
```

### External.java:

```
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]);
 }
 }
}
```

## 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();
 }
 }
}
```

## LAB 8 PROGRAM:

### Age exception program (father and son ages):

```
import java.util.Scanner;

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

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

```
}

class InputScanner {
 protected Scanner scanner;

 public InputScanner() {
 scanner = new Scanner(System.in);
 }

 public int nextInt() {
 return scanner.nextInt();
 }
}

class Father extends InputScanner {
 protected int fatherAge;

 public Father() {
 System.out.println("Enter father's age:");
 fatherAge = super.nextInt();

 if (fatherAge < 0) {
 throw new WrongAge("Age cannot be negative");
 }
 }

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

class Son extends Father {
 private int sonAge;

 public Son() {
 super();
 System.out.println("Enter son's age:");
 sonAge = super.nextInt();

 if (sonAge > fatherAge) {
 throw new WrongAge("Son's age cannot be greater than father's age");
 } else if (sonAge < 0) {
 throw new WrongAge("Age cannot be negative");
 }
 }
}
```

```

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

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

```

## LAB 9 PROGRAM:

### THREAD PROGRAM:

#### ThreadMain.java:

```

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();
 }
}

```

### Deadlock.java:

```

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

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

```

```

}
class B {

 synchronized void bar(A a) {
 String name = Thread.currentThread().getName();

```

```

 System.out.println(name + " entered B.bar");
 try {
 Thread.sleep(1000);
 } catch (Exception e) {
 System.out.println("B Interrupted");
 }

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

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

class Deadlock implements Runnable {
 A a = new A();
 B b = new B();

 Deadlock() {
 Thread.currentThread().setName("MainThread");
 Thread t = new Thread(this, "RacingThread");
 t.start();
 a.foo(b); // get lock on a in this thread.
 System.out.println("Back in main thread");
 }

 public void run() {
 b.bar(a); // get lock on b in other thread.
 System.out.println("Back in other thread");
 }
}

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

```

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);
 }
}
```

# LAB 10 PROGRAMS:

## Function program for a calculator:

```
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 (ArithmaticException e) {
 alab.setText("");
 blab.setText("");
 anslab.setText("");
 err.setText("B should be NON zero!");
 }
 }
 });
}

// display frame
jfrm.setVisible(true);
}

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

```

