

B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



OBJECT ORIENTED JAVA PROGRAMMING LAB OBSERVATION

(23CS3PCOOJ)

Submitted by:

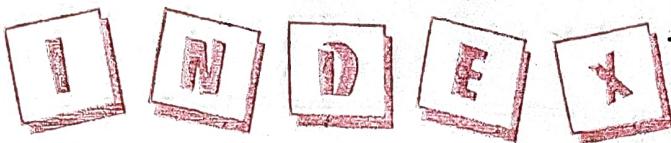
SHASHANK SP

1BM22CS256

Department of Computer Science and Engineering

B.M.S College of Engineering

Bull Temple Road, Basavanagudi, Bangalore 560 019



IBM22CS256

NAME: Ghashank-SP STD.: SEC.: 3E ROLL NO.: IBM22CS256.

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
01	18/12/23	Quadratic Equation	1-3	(9)
02	19/12/23	SGPA calculator	4-7	(9)
03	26/12/23	BOOK Details	8-11	(9) 27/12/2024
04	02/01/24	Abstract class	12-15	(10)
05	09/01/24	BANK ACCOUNT	16-21	(10) 8/16/2024
06	23/01/24	CIE & SEE Package	25-28	10 28/1/2024
07.	30/01/24	Exception Handling	30-32	10 31/1/2024
08,	06/02/24	multithreaded program	10 8/2/2024	
09	13/02/24	a) inter process communication b) deadlock	10 13/2/2024	
10	20/02/24	USER INTERFACE <u>(Gvisor APP)</u>	10 20/2/2024	

LAB-01

Quadratic equation

Import `java.util.Scanner`

public class Quadratic

{
 int a, b, c;

 double r1, r2, d;

 void get_d()

 {
 System.out.print("Enter the coefficients: ");

 a = nextInt();
 b = nextInt();
 c = nextInt();

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

 }

 void compute()

 {
 if (d < 0)
 System.out.println("No real roots");

 while (a == 0)

 {
 System.out.println("Input a quadratic eqn");

 System.out.print("Enter a non zero value: ");

 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 equal");

 }

```

} System.out.println("Root 1 = Root2 = " + x1);
} else if (d > 0) {
    r1 = ((-b) + (math.sqrt(d))) / (2 * a); // double
    r2 = (c - b) - (math.sqrt(d)) / (2 * a); // double
    System.out.println("Roots are real & distinct");
    System.out.println("Root 1 = " + 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("Root 1 = " + r1 + " + " + r2);
    System.out.println("Root 2 = " + r1 + " - " + r2);
}
}

```

class Quadratic Main

```

public static void main (String args[])
{
    quadratic q = new Quadratic ();
    q = get d();
    q = compute();
    System.out.println ("Shashank _");
}

```

}

Output

Enter the coefficients of a,b,c

1

2

1

Roots are real and equal

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

shashank - 13m2QCS256

Enter coefficients of a,b,c

1

3

2

Roots are real and distinct

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

shashank - 13m2QCS256

Enter coefficients of a,b,c

(2) Roots are real and distinct

$$10x^2 + 20x + 10 = 0 \Rightarrow x^2 + 2x + 1 = 0$$

$$3x^2 + 6x + 3 = 0 \Rightarrow x^2 + 2x + 1 = 0$$

Roots are imaginary.

$$\text{Root 1} = 0.1 + i 1.198578808$$

$$\text{Root 2} = 0.1 - i 1.198578808$$

shashank - 13m2QCS256

9

Next page for solving the next type

of question

(using imaginary numbers)

of solving the next type

10/12/23

LAB-08

4

import java.util.Scanner;

class Subject

{ int subject_marks;

int credits;

int grade;

}

class Student

{

Subject subject[8];

String name;

String usn;

double sgpa = 0;

Scanner s;

System.out.println("Enter student details");

{

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

subject = new Subject[8];

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

subject[i] = new Subject();

s = new Scanner(System.in);

void getStudentDetails()

{

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

name = s.next();

System.out.println("Enter your usn");

usn = s.next();

}

```
void get_marks()
```

```
{
```

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

```
        System.out.print("Enter marks for " +
```

```
        subject[i] : " + subject[i] + ":" );
```

```
    Subject[i].subjectMarks = scanner.nextInt();
```

```
    System.out.println("Enter credits " +
```

```
    subject[i].credits = scanner.nextInt();
```

```
    subject[i].grade = (subject[i].subjectMarks /
```

```
    subject[i].credits) * 10;
```

```
    if (subject[i].grade >= 9)
```

```
        subject[i].grade = 10;
```

```
    if (subject[i].grade <= 4)
```

```
        subject[i].grade = 0;
```

```
    } else if (subject[i].grade > 4 & subject[i].grade < 9)
```

```
        subject[i].grade = 5;
```

```
}
```

```
void compute_SGPA()
```

```
{
```

```
    int totalCredits = 0;
```

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

```
        totalCredits += subject[i].credits;
```

```
    SGPA = (subject[0].credits * subject[0].grade +
```

```
    subject[1].credits * subject[1].grade +
```

```
    subject[2].credits * subject[2].grade +
```

```
    subject[3].credits * subject[3].grade +
```

```
    subject[4].credits * subject[4].grade +
```

```
    subject[5].credits * subject[5].grade +
```

```
    subject[6].credits * subject[6].grade +
```

```
    subject[7].credits * subject[7].grade) / totalCredits;
```

```
}
```

```
class main
```

```
{
```

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

```
{
```

```
    Student st = new Student();
```

11: get student Details (7)

11. get marks (7)

11. compute SGPA (7)

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

System.out.println("USN: " + usn)

System.out.print("SGPA: " + sgpa)

System.out.println("CGPA: " + cgpa)

System.out.println("Percentage: " + percentage)

System.out.println("Grade: " + grade)

Output

Enter your name

shashank

Enter your usn

1BM22CS256

Enter marks of subject 1

94

Enter credits of subject 1

4

Enter Marks of subject 2

85

Enter credits of subject 2

4

Enter marks of subject 3

85

Enter credits of subject 3

3

Enter marks of subject 4

86

Enter credits of subject 4

3

Enter marks of subject 5

81

60 - 893

Enter credits of subject 5

Enter marks of subject 6

Enter credits of subject 6

Enter marks of subject 7

Enter credits of subject 7

Enter marks of subject 8

98

Enter credits of subject 8

1

NAME : shashank

USN : 13m8acs856

SGPA : 9.378999999999999

(a) student print (annex points) show
Shashank SGP 9.378999999999999 USN 13m8acs856

(i) print of profile diff

06/12/23

8

LAB - 03

- Q. Create a class book which contains four numbers : name, author, price, numPages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() that could display the complete details of the book. Develop a java program to create a book objects.

Import java.util.Scanner;

class Books {

String name;

String author;

int price;

int numPages;

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

this.name = name;

this.author = author;

this.price = price;

this.numPages = numPages;

}

public String toString()

String name, author, price, numPages;

name = "Book names" + this.name + "\n";

author = "Author names" + this.author + "\n";

Price = "Price" + this.price + " | n";
 numPages = "Number of Pages : " + this.numPages + " | n";

return name + author + Price + numPages;

class Main {

public static void main (String args[]) {

Scanner s = new Scanner (System.in);

int n;

String name, author;

int price, numPages;

System.out.print ("Enter number of

books : ");

n = s.nextInt();

Books b[];

b = new Books [n];

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

System.out.print ("Enter the name,
author, price, number of pages of the book");

name = s.next();

author = s.next();

price = s.nextInt();

numPages = s.nextInt();

b[i] = new Books (name, author, price,

numPages);

}

System.out.print ("The books details : | n");

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

```
System.out.println(t[i].toString());
```

```
Output : Enter the number of Books
```

2

Enter the name, author, price and number
of pages of the book :

java xyz 1200 100

xyz 1200

1200

100

Enter the name, author, Price and number
of Pages of the books :

c++ abc 1500 843

abc 1500

1500

843

~~The book details are as follows~~

Book name : java

Author name : xyz

Price : 1200

Number of pages : 100

Book name : c++

Author name : abc

Price : 1500

Number of pages : 843

Shashank P. SP

IBM&CS256

(g)

21/1/2024

Chandigarh University, Sector 42, Mohali

soft (f) 1200.00 each book

each book price is 1200.00

(Calculus) 1000.00 each book

each book price is 1000.00

(Geometry) 1000.00 each book

each book price is 1000.00

(Statistics) 1000.00 each book

each book price is 1000.00

(Computer) 1000.00 each book

each book price is 1000.00

(Computer) 1000.00 each book

each book price is 1000.00

(Computer) 1000.00 each book

each book price is 1000.00

(Computer) 1000.00 each book

each book price is 1000.00

LAB-04

* 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 Shape. Each one of the classes contain only the method printArea() that print the area of the given shape.

```

⇒ import java.util.*;
class InputScanner {
    protected Scanner s;
    public InputScanner() {
        s = new Scanner(System.in);
    }
    public double getDoubleInput(String message) {
        System.out.println(message);
        return s.nextDouble();
    }
    public void closeScanner() {
        s.close();
    }
}
abstract class Shape extends InputScanner {
}

```

Protected double dim1;

protected double dim2;

Shape();

{}
} Super();

Public abstract void paintArea();

Class Rectangle extends Shape

public Rectangle()

{
 dim1=50.0; dim2=30.0;
}

Super();

Public void paintArea()

{
 System.out.println("Painting rectangle");
}

dim1 = getDoubleInput("enter length of the
rectangle");
dim2 = getDoubleInput("enter breadth of the
rectangle");
double area = dim1 * dim2;

System.out.println("Area of the rectangle")

Class Triangle extends Shape

public Triangle()

Super();

Public void PrintArea ()

{
System.out.println("Enter side of the triangle") ;

dim1 = getDoubleInput ("enter base of the triangle");

dim2 = getDoubleInput ("enter height of the triangle");

double area = 0.5 * dim1 * dim2;

System.out.println ("area of the triangle
is " + area);

}

}.

class Circle extends Shape

{

Public Circle ()

{
System.out.println ("Enter radius of circle");

Super ();

public void printArea ()

{
System.out.println ("Enter radius of circle");

dim1 = getDoubleInput ("enter radius of the circle");

double area = Math.PI * dim1 * dim1;

System.out.println ("area of the circle is " + area);

}

}.

public static void main (String [] args)

Rectangle r = new Rectangle (2, 2);

Triangle t = new Triangle (2, 2);

A. PointArea(?)

B. PointArea(?)

C. PointArea(?)

D. closeScanner(?)

E. y-axis(?)

F. startAtOrigin(?)

G. endAtOrigin(?)

H. Output

Enter length of the rectangle

8

rectangle found

Enter breadth of the rectangle

3

area of the rectangle: 24.0

enter base of the triangle

7

triangle found

Enter height of the triangle

3

triangle found

area of the triangle: 10.5

enter radius of the circle

3

circle found

area of the circle: 28.274333882308138

shashankSP 201920181BM82CS256

21/2024 (0) 11:48:00 AM

LAB - OOPS [BANK ACCOUNT]

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. 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.

import java.util.Scanner;

```
class InputFormat {
    Scanner sc = new Scanner(System.in);
}
```

```
class Account extends InputFormat {
    String name;
    int acc_no;
    double balance;
```

```
void getDetails() {
```

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

```
    System.out.print("Enter accno.");
    accno = sc.nextInt();
```

```
}
```

```

20 void deposit () {
    System.out.print("Enter amount to deposit");
    double amt = nextDouble();
    balance += amt;
    System.out.println("Amount Deposited");
}

```

```
void withdraw ()
```

```

    System.out.print("Enter amount");
    double amt = nextDouble();
    if (balance >= amt) {
        balance -= amt;
    }

```

```
System.out.println("Amount withdrawn");
```

```
} else {
    System.out.println("Insufficient Balance");
}
```

```
void display ()
```

```

    System.out.println("Name: " + name);
    System.out.println("Account number:");
    System.out.println("Balance: " + balance);
}

```

class current extends Account

double minbal = 500;

double penalty = 100;

```
void withdraw ()
```

```
super.withdraw();
```

```
chargeBalance();
```

Print void check min Balance() {
if (balance < minInt) {

balance = penalty;

System.out.println("Penalty applicable for
low Balance");

}

}

class Saving extends Account {

double intRate = 0.04;

void computeInterest () {

double interest = balance * interestRate;
balance += interest;

System.out.println("Interest credited, " + interest);

}

class Bank extends Account {

public static void main (String [] args) {

Saving ob1 = new Savings();

Current ob2 = new Current();

ob1.getDetails();

ob2.getDetails();

int choice;

String acc;

System.out.println("menu");

System.out.println("1. Deposit ");

2. withdraw in 3 display Balance in

4. compute \$ interest (savings only)

5 exit();

do {

System.out.print("Enter your choice");
choice = sc.nextInt();

System.out.print("Enter the acc type");
acc_type = sc.next();

switch(choice) {

case 1:

if (acc.equals("Saving")) {

obj1.deposit();

class {

obj2.deposit();

break;

case 2:

if (acc.equals("Savings")) {

obj1.withdraw();

(else withdraw) else withdraw();

obj2.withdraw();

break;

case 3:

if (acc.equals("saving")) {

obj3.display();

else

obj2.display();

break;

case 4: withdrawal

obj1.computeInterest();

break;

case 5:

break;

defaults;

system : but part of C++ provided choice

{ while (option != 5); }

if (exit) { break; }

OUTPUT

Enter name Shashank

Enter acc no: 1

Enter name Sharmend

Enter acc no: 2

menu

1. deposit

2. withdraw

3. Display

4. Compute interest (Savings only),

5. Exit

Enter your choice: 1

Enter acc type: saving

Enter deposit amount: 6000

Amount deposited successfully.

Enter your choice: 1

Enter the acc type
current

Enter amount to withdraw: 4500

Amount withdraw successfully

Penalty applied for low balance

Enter your choice: 3

Enter the acc type
Current

name: shashank

Account number: 2

Balance 300.0

Enter your choice: 2

Enter the acc type: 1

Saving

Enter amount to withdraw: 200

In sufficient balance

Enter your choice: 4

Enter the acc type: 1

Saving

~~Interest calculated 4.0% () period 1~~

2024 Shashank-SP 11/08/2024 P9BM&2CS256

161

1. i) For 118 P.D. 2024 10.00

2. 100 amounts at 10.00 per unit

3. 100 amounts at 10.00 per unit

4. 100 amounts at 10.00 per unit

5. 100 amounts at 10.00 per unit

6. 100 amounts at 10.00 per unit

7. 100 amounts at 10.00 per unit

8. 100 amounts at 10.00 per unit

9. 100 amounts at 10.00 per unit

10. 100 amounts at 10.00 per unit

11. 100 amounts at 10.00 per unit

12. 100 amounts at 10.00 per unit

13. 100 amounts at 10.00 per unit

14. 100 amounts at 10.00 per unit

15. 100 amounts at 10.00 per unit

16. 100 amounts at 10.00 per unit

17. 100 amounts at 10.00 per unit

18. 100 amounts at 10.00 per unit

Strings

1)

String constructor of BMSCE

abcd :

String constructor using object : abcd

String constructor with index of chars : bed

String Constructor using ASCII values of characters

String constructor using ASCII : 65D

String concatenation : BMSCE + COLLEGE

2)

String length of chars : 3

String length of literals : 8

String concatenation : BMS : COLLEGE

3)

toString() method

2)

String using getBytes()

87 101 108 99 111 109 101

String using toCharArray()

w e l l o m e

4)

BMSCE equals BMsce : true

Bmsee equals college : false

Bmsce equals BMSCe : false

Bmsee equals BMSCE : true

5)

Substring is matched

6)

String using startsWith() : true

String using startsWith() : false

7)

String using endsWith() : true

String using endsWith() : false

8)

Hello == Hello : false

11) apple ball cat dog and tree queen hen rice
 jug kite wif man hot orange parrot queen
 ring star tree umbrella van watch & mca
 yacht zee

12) 1 2 3 4 5 6 7 8 9 10 11 12

13) This is a test, This is, tan

14) Hello world

15) incomerl

16) Hello friends

17) Student 1
 name: Grouind
 Reg no: 81
 Semester : 3
 CGPA : 9.2

Student 2
 name: v. nikhil
 Reg no: 88
 Semester 5
 CGPA : 9.2

18) class of 3rd
 BMSC
 ESMB

84

19. Eagle is flying over land
Eagle makes a sound
Hawk is meowing
Hawk is making a sound

20.

Genetics

Enter the elements to integer stack

1

2

3

4

5

Enter the elements to double stack

1.0

2.0

3.0

4.0

5.3

1

elements of stack 1

8

4

3

8

1

elements of stack 2

5.3

4.0

3.0

2.0

1.0.

Q6 Develop a Java program and create a CIE package consist of student and internal classes and SEE package consists of Extends and main class and display the student marks in internal & externally.

1) Student.java

```
import java.util.Scanner;
public class student
```

```
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
```

```
    public void InputStudentDetails()
```

```
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter USN");
        usn = scanner.nextLine();
```

```
        System.out.println("Enter name");
        name = scanner.nextLine();
```

```
        System.out.println("Enter Sem");
        sem = scanner.nextInt();
```

```
    public void displayStudentDetails()
```

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

~~23/07/2011~~

LAB-06

Q6

* Create a package

//internals.java

Package CIE;

Import java.util.Scanner;

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

Public void Input (CIE marks) {

Scanner s = new Scanner (System.in);

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

marks [i] = s.nextInt ();

}

(C) Create another input marks function.

Package SEE;

Import CIE (Internals);

Import java.util.Scanner;

Public class Externals extends Internals {

Protected int marks [];

Protected int final marks [];

Public Externals () {

marks [] = new int [5];

(C) Final marks = new int [5];

Public void Input SEE marks () {

Scanner s = new Scanner (System.in);

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

System.out.println ("subject " + i + " marks ");

marks [i] = Scanner.nextInt ();

}

public void calculateFinalMarks() {

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

finalMarks[i] = marks[i] / 2 + supermarks[i]

{}

public void displayFinalMarks() {

displayStudentDetails();

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

System.out.print("Subject " + (i + 1) + ":" + finalMarks[i]);

{}

import SEE.Externals;

class main {

public static void main (String args[]) {

int numOFStudents = 2;

Externals finalMarks[] = new Externals [numOfStudent];

finalMarks[i] = new Externals();

finalMarks[i].inputStudentDetails();

System.out.println("Displaying Data");

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

FinalMarks[i].calculateFinalMarks();

FinalMarks[i].displayFinalMarks();

{}

System.out.println("Shashank SP :
IBM&2CS256")

{}

Output

Enter student details:

IBM22CS256 Jitendra Patel

shashank

3 Subject marks pathik Patel

Enter CIE marks:

Subject 1: 540

Subject 2: 745

Subject 3: 640

Subject 4: 48

Subject 5: 44

Enter SEE marks:

Subject 1: 80 marks 332 %

Subject 2: 90 marks 22%

Subject 3: 86 marks 21%

Subject 4: 80 marks 20%

Subject 5: 84 marks 20%

(Average marks)

Shashank-SP IBM22CS256

1) Student registration (2) Attendance

(3) Subject marks pathik Patel

(4) CIE marks pathik Patel

(5) SEE marks pathik Patel

(6) Average marks pathik Patel

Pathik Patel (10) pathik Patel

Pathik Patel

① Write a program that demonstrates handling of exceptions in C. Create a base class called 'Father' & derived class named 'Son' which extends the base class. On Father class throws the exception wrong Age(). When the frontage is 0, on Son class implement a that class both father & son's age() throws own exception if son's age is \geq Father's age.

Import 'java.util.Scanner' class wrong Age extends Exception

Public WrongAge (String message)

super (message)

throws WrongAge ()

class InputScanner

{

protected Scanner getIn()

public InputScanner()

{} (empty) // constructor

System.out.println("Enter Father's age");

class Father extends InputScanner

protected int FatherAge();

public Father() throws WrongAge

System.out.println("Enter Father's age");

no

19/10/2021

if (FatherAge >= 5, nextInt());

if (FatherAge < 0)

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

public void display()

System.out.println ("Father's age:

current, " + FatherAge);

class son (Extends Father)

protected int SonAge;

public Son () throws WrongAge

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

SonAge = sc.nextInt();

if (SonAge >= FatherAge)

throws new WrongAge ("Son's age

can't be greater than or equal to
father's age");

if (SonAge < 0)

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

Public void display()

super.display();

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

}

Public class Main

Public static void main (String args[])

{

try

Son son = new Son();

son.display();

Catch (WrongAge e)

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

}

Output:-

enter Father's age:

85

enter Son's age:

25

Error: Son's age can't be greater than or equal to father's age.

enter Father's age;

-35

Error: Age cannot be negative

enter Father's age;

34

enter Son's age;

-9

Error: Age cannot be negative

enter Father's age;

35

enter Son's age;

13

Father's age: 35 | (Validation)

Son's age: 13

Shashank SP

1Bm20cs256

work: compare first you want
and it will be longer so

06/02/2014

LAB-08

Bafna Gold

Date:

Page:

- * 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.
- class displayMessage implements Runnable
 - String message;
 - int interval;

```
displayMessage (String message, int interval) {  
    this.message = message;  
    this.interval = interval;  
}
```

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

```
    while (true) {
```

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

```
        try {
```

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

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

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

```
        }
```

```
}
```

```
public class multith
```

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

```
    Thread t1 = new Thread (new displayMessage  
        ("BMS College of Engineering", 10));
```

```
    Thread t2 = new Thread (new displayMessage  
        ("CSE", 2));
```

part-1: start(1), connecting to server
part-2: start(2), connecting to server
part-3: stop, implementing the global server
part-4: separation from a client and
output

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE or (Computer Application)

CSE in programming language (C/C++)

CSE or MCA, M.Tech, M.Sc

BMS College of Engineering

Sg
6/20/2014

Shashank SP

IBM&CS256

13/08/24

LAB =

Bafna Gold

Date:

Page:

a) * Demonstrate inter process communication and deadlock:

class Q {

int n;

boolean valueSet = false;

synchronized int get () {

while (!valueSet)

try {

System.out.println ("All consumers waiting\n");

wait ();

} catch (InterruptedException e) {

System.out.println ("Interrupted Exception caught");

}

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

valueSet = true;

synchronized void put (int n) {

notify ();

return n;

}

synchronized void put (int n) {

while (!valueSet)

try {

System.out.println ("All Producers waiting\n");

wait ();

} catch (InterruptedException e) {

System.out.println ("Interrupted Exception caught");

}

this.n = n;

valueSet = true;

System.out.println("Put " + n);
System.out.println("In Intermediate Conditionality (?)");

{

{

class Producer implements Runnable {

Q q; // for too fast insertion
Producer(Q q) { this.q = q; }

{

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

{

public void run() {

while (true) {

{

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 (true) {

int x = q.get();

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

{}

{

{

class prefixed {

```
public static void main (String args[]) {
    Queue<String> q = new Queue<String>();
    new Producer(q);
    new Consumer(q);
    System.out.println ("Press control-c to stop.");
}
```

}

Output

Put: 0

Intimate consumer

Producer writing

Press control-c to stop

Not: 0

Intimate producer

Consumed: 0

Put: 1

Intimate consumer

producer writing

Not: 1

Intimate producer

Consumed: 1

put: 2

Intimate consumer

producer writing

Not: 2

Intimate producer

Consumed: 2

put: 3

Intimate consumer

producer writing

Part 3

Sharing mod)

Intimate producer (short product life cycle)

Consumer: 3 (C) demand & growth

Part 4 (C) market entry

Intimate consumer (C) demand & growth

Producer waiting selling characteristics

Part 4

Intimate Producer

Consumer: 4

Part 5

Intimate Consumer

Part 6 (Normalized) short life

Intimate Producer (short life)

Consumer: 5 (short life) & fast growth

On top

Starhawk Spring Cleaning IBM&CS256

Off balance

On top

Intimate short life

Intimate short life

Intimate short life

(Normalized)

Intimate short life

Normalized short life

Normalized short life

Normalized short life

Normalized short life

On top

Starhawk Spring Cleaning IBM&CS256

Normalized short life

b) class A {

synchronized void foo(B b) {

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

Thread.sleep(1000);

} catch (InterruptedException e) {

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

} finally {

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

b.last();

} // synchronized block ends here.

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 (InterruptedException e) {

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

void last()

System.out.println("inside B.last()");

}

}

to start (7)

action (8) "I get lock on a in this thread

System.out.println("Back in main

of thread")

public void run() {
 a.b.a(); // get lock on b in other thread
 inside p.last }

Back in main thread

Racing Thread trying to call p.last()

inside p.last (race condition, b is not yet back in other thread)

Back in other thread

public static void main (String args)

{
 new Thread (lock) () {
 public void run () {
 System.out.println("Main thread entered A, for
 racing thread to call B, back in main
 thread")
 System.out.println("Racing Thread trying to call B, last
 inside p.last")
 Back in main thread
 Racing Thread trying to call p.last()
 inside p.last
 Back in other thread

lock.lock();
 System.out.println("Racing Thread trying to call B, last
 inside p.last")
 Back in other thread

Shashank SP 13/2/2024 11:26 AM 256

13/2/2024

20/02/24

LAB

USER INTERFACE

Division Main.java

Import java.awt.*;

Import java.awt.event.*;

public class DivisionMain extends Frame
implements ActionListener {

TextField num1, num2;

Button dResult;

Label outResult;

String out = " ";

double resultnum;

int flag = 0;

public DivisionMain() {

setLayout(new FlowLayout());

dResult = new Button("RESULT");

label1 = new Label("number1");

label2 = new Label("number2");

num1 = new TextField();

num2 = new TextField();

outResult = new Label("Result");

add(number1);

add(num1);

add(number2);

add(num2);

add(dResult);

add(outResult);

num1.addWindowListener(this);
addwindowListener(new windowAdapter)

{
public void windowClosing(WindowEvent we)

{System.exit(0);}

}

} for adding window

public void actionPerformed(ActionEvent e)

{int n1, n2;

try

if (e.getSource() == dResults)

n1 = Integer.parseInt(num1.getText());

n2 = Integer.parseInt(num2.getText());

if (n2 == 0)

throw new ArithmeticException();

out.println(" " + n2);

result = num1 / n2;

out.println(" " + result.toString());

System.out.println(result);

}

catch : (numberFormatException e1)

flag = 1;

out = "Number Format Exception!"
repaint();

catch : (ArithmaticException e2)

flag = 1;

out = "divide by 0 Exception!"
repaint();

public void paint (Graphics g)

dm = new Dimension(800,400)

dm.setSize(new Dimension(800,400))

dm.setTitle("Division of Integers")

dm.setVisible(true);

Output:

numbers: 12 . numbers: 12

Result: 24

Shashank-SP

(Bm22cs256)

Functions Used:-

- i) `JFrame` :- It is a top level container in Java string that represent a window with a title bar, border and optional menu.
- ii) `setSize` :- It is used to set size of the frame.
- iii) `setLayout` :- This line sets the layout manager for the frame to `FlowLayout` which arranged components from left to right in the flow manner.
- iv) `add` :- This line adds the error label to the frame.
- v) `setText` :- This line sets the text of the 'a' label to display the value of 'A'.
- vi) `setVisible` :- This line makes the frame visible.
- vii) ~~SwingWorker~~ :- To perform tasks synchronously in the current thread.

Shashank SP

1BM202CS256

20/01/2024
20

Lab program no 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;
```

```
public class QuadraticMain
```

```
{
```

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

```
    System.out.println("Shashank S P");
    System.out.println("1BM22CS256");
    Quadratic q = new Quadratic();
    q.getd();
    q.compute();
}
```

```
}
```

```
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);
}
}

Lab program no 2:
```

Develop a Java program to create a class Student with members usn, name, an array credits and

an array marks. Include methods to accept and display details and a method to calculate SGPA of

a student.

```
import java.util.Scanner;

class Subject{
    int subjectMarks;
    int credits;
    int grades;
}

class Student{
    Subject subject[];
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Student() {
        subject = new Subject[9];
        for(int i = 0;i<9;i++) {
            subject[i] = new Subject();
        }
        s= new Scanner(System.in);
    }
    void getStudentDetails(){
        System.out.println("Enter your name: ");
        this.name = s.nextLine();
        System.out.println("Enter your usn: ");
        this.usn = s.next();
    }
    void getMarks(){
        for(int i = 0;i<8;i++) {
            System.out.println("Enter the marks of the "+(i+1)+" subject");
            subject[i].subjectMarks = s.nextInt();
            System.out.println("Enter the credits of the "+(i+1)+" subject");
        }
    }
}
```

```
subject[i].credits = s.nextInt();
subject[i].grades = (subject[i].subjectMarks/10)+1;
if(subject[i].grades >10){
subject[i].grades = 10;
}
if(subject[i].grades <4) {
subject[i].grades = 0;
}
}

void computeSGPA(){
int sum=0;
int totalCredits = 0;
for(int i = 0;i<9;i++){
sum+=(subject[i].grades * subject[i].credits);
totalCredits += subject[i].credits;
}
this.SGPA = (double) sum/totalCredits;
}

}

public class Main{
public static void main(String args[]){
Student s1 = new Student();
s1.getStudentDetails();
s1.getMarks();
s1.computeSGPA();
System.out.println("Name: "+s1.name);
System.out.println("Usn: "+s1.usn);
System.out.println("SGPA: "+s1.SGPA);
System.out.println("Shashank S P");
System.out.println("1BM22CS256");
}
}
```

```
}
```

Lab Program no 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 name;
    String author;
    int price;
    int numPages;

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

    public String toString(){
        String name, author, price, numPages;

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

        return name + author + price + numPages;
    }

    void setName(String name) {
        this.name = name;
    }
}
```

```
}

void setAuthor(String author) {
    this.author = author;
}

void setPrice(int price) {
    this.price = price;
}

void setNumPages(int numPages) {
    this.numPages = numPages;
}

String getName() {
    return name;
}

String getAuthor() {
    return author;
}

int getPrice() {
    return price;
}

int getPgNum() {
    return numPages;
}

}

class Main{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int n, price, numPages;
        String name, author;
        System.out.println("Enter the number of books");
        n = sc.nextInt();
        sc.nextLine();
        Books b[] = new Books[n];
    }
}
```

```

for(int i =0;i<n;i++) {
    System.out.println("Read name of the book");
    name = sc.nextLine();
    System.out.println("Read author of the book");
    author = sc.nextLine();
    System.out.println("Read the price of the book");
    price = sc.nextInt();
    System.out.println("Read pgNumbers of the book");
    numPages = sc.nextInt();
    sc.nextLine();
    System.out.println("-----");
}
b[i] = new Books(name,author,price,numPages);
}

for(int i =0;i<n;i++) {
    String bookDetails = b[i].toString();
    System.out.println(bookDetails);
}

for(int i =0;i<n;i++) {
    System.out.println("Book name is "+b[i].getName());
    System.out.println("Book author is "+b[i].getAuthor());
    System.out.println("Book price is "+b[i].getPrice());
    System.out.println("Book has number of pages =
"+b[i].getPgNum()+"\n");
}

System.out.println("Name : Shashank S P");
System.out.println("USN : 1BM22CS256");

}
}

Lab Program 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{
```

```
    Scanner s;
```

```
    InputScanner() {
```

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

```
    }
```

```
}
```

```
abstract class Shape extends InputScanner{
```

```
    double a;
```

```
    double b;
```

```
    abstract void getInput();
```

```
    abstract void displayArea();
```

```
}
```

```
class Rectangle extends Shape{
```

```
    void getInput() {
```

```
        InputScanner is = new InputScanner();
```

```
        System.out.println("Enter the length and breadth of the rectangle  
:" );
```

```
        a = is.s.nextDouble();
```

```
        b = is.s.nextDouble();
```

```
    }
```

```
    void displayArea() {
```

```
        System.out.println("The area of the rectangle is :" +(a*b));
```

```
    }
```

```
}
```

```
class Triangle extends Shape{
```

```
void getInput() {
    InputScanner is = new InputScanner();
    System.out.println("Enter the base and height of the triangle :");
    a = is.s.nextDouble();
    b = is.s.nextDouble();
}

void displayArea() {
    System.out.println("The area of the triangle is :" + (a*b*0.5));
}

}

class Circle extends Shape{

void getInput() {
    InputScanner is = new InputScanner();
    System.out.println("Enter radius of the Cirlce :");
    a = is.s.nextDouble();
}

void displayArea() {
    System.out.println("The area of the Circle is :" + (3.14*a*a));
}

}

public class AbstractMain{
    public static void main(String args[]){
        System.out.println("HI");
        Rectangle rect = new Rectangle();
        rect.getInput();
        rect.displayArea();
        Triangle triangle = new Triangle();
        triangle.getInput();
    }
}
```

```
triangle.displayArea();

Circle circle = new Circle();

circle.getInput();

circle.displayArea();

System.out.println("Shashank S P");

System.out.println("1BM22CS257");

}

}
```

Lab program 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.

```
import java.util.*;

class Account{

    String name;

    int accno;

    String type;

    double balance;

    private int mins=2000;

    Account(String name,int accno,String type,double balance){

        this.name = name;

        this.accno = accno;

        this.type = type;

        this.balance = balance;

        if(balance<mins){

            System.out.println("Insufficient balance");

        }

    }

}
```

```
}

void deposit(double amount) {
    balance +=amount;
}

void withdraw(double amount) {
    if((balance-amount)>=0 ) {
        balance -=amount;
    }
    else{
        System.out.println("Insufficient balance");
        return;
    }
}

void display() {
    System.out.println("Name : "+name+"\n"+
"AccountNo : "+accno+"\n"+
"Type : "+type+"\n"+
"balance: "+balance+"\n");
}

class SavingAccount extends Account{
    private static int rate = 5;
    SavingAccount(String name,int accno,String type,double balance) {
        super(name,accno,type,balance);
    }
    void balanceWithInterest() {
        balance +=balance*rate/100;
        System.out.println("balance: "+balance);
    }
}

class CurrAccount extends Account{
    private static int minBalance = 2000;
```

```
private static int charge = 100;

CurrAccount(String name,int accno,String type,double balance) {
super(name,accno,type,balance);
}

void checkMin() {
if(balance<minBalance) {
System.out.println("Balance is less than min balance service
charge exposed " + charge);
balance -=charge;
return;
}

System.out.println("balanc is "+balance);
}

}

public class Main{
public static void main(String args[]){
System.out.println("Shashank S P");
System.out.println("1BM22CS257");
Scanner s = new Scanner(System.in);
System.out.println("Enter your name: ");
String name = s.nextLine();
System.out.println("Enter the account type (current or deposit)");
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();
```

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

SavingAccount sa = new SavingAccount(name,accno,type,balance);

CurrAccount ca = new CurrAccount(name,accno,type,balance);

double amount;

while(true) {

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

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

System.out.println("1. Deposit \t2.Withdraw \t

3.compute interest for SavingsAccount \t 4.Display Account Details\n

5.Exit\t");

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

int choice = s.nextInt();

switch(choice) {

case 1:System.out.println("Enter the deposit

amount");

amount = s.nextDouble();

sa.deposit(amount);

break;

case 2: System.out.println("Enter the withdrawl

amount ");

amount = s.nextDouble();

sa.withdraw(amount);

break;

case 3:sa.balanceWithInterest();

break;

case 4:System.out.println("Details: ");

sa.display();

break;

case 5: return;

default: System.out.println("Invalid choice ");

}

}
```

```

}

else{
    System.out.println("1. Deposit \t2.Withdraw \t
3.Display Account Details\n 4.Exit\t");
    System.out.println("Enter the choice");
    int choice = s.nextInt();
    switch(choice){
        case 1:System.out.println("Enter the amount : ");
        amount = s.nextInt();
        ca.deposit(amount);
        break;
        case 2:System.out.println("Enter the amount: ");
        amount = s.nextInt();
        ca.withdraw(amount);
        ca.checkMin();
        break;
        case 3 : ca.display();
        break;
        case 4: System.exit(0);
    }
}

}
}
}
}

Lab program 6-a

```

Write a Java program to create a generic class Stack which hold 5 integers and 5 double values

and

```

String method demonstraθons
import java.util.ArrayList;
import java.util.List;
class Stack<T> {

```

```
private List<T> elements = new ArrayList<>();
private int maxSize;
public Stack(int maxSize) {
    this.maxSize = maxSize;
}
public void push(T element) {
    if (elements.size() < maxSize) {
        elements.add(element);
        System.out.println("Pushed: " + element);
    } else {
        System.out.println("Stack is full. Cannot push more elements.");
    }
}
public T pop() {
    if (!elements.isEmpty()) {
        T poppedElement = elements.remove(elements.size() - 1);
        System.out.println("Popped: " + poppedElement);
        return poppedElement;
    } else {
        System.out.println("Stack is empty. Cannot pop elements.");
        return null;
    }
}
public class Main {
    public static void main(String[] args) {
        // Creating a stack for integers
        Stack<Integer> intStack = new Stack<>(5);
        intStack.push(1);
        intStack.push(2);
        intStack.push(3);
        intStack.pop();
    }
}
```

```
intStack.push(4);
intStack.push(5);
intStack.push(6);
Stack<Double> doubleStack = new Stack<>(5);
doubleStack.push(1.1);
doubleStack.push(2.2);
doubleStack.push(3.3);
doubleStack.pop();
doubleStack.push(4.4);
doubleStack.push(5.5);
doubleStack.push(6.6);
System.out.println("Shashank S P");
System.out.println("1BM22CS257");
}
}
```

Strings:

```
import java.util.*;
class StringConstructor{
    public static void main(String args[]){
        System.out.println("Question : 1");
        String s1 = new String();
        s1 = "newString";
        char ch[]={ 'a','b','c','d' };
        String s2 = new String(ch);
        String demo = "Hello";
        String s = new String(demo);
```

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

System.out.println("Question : 2");
String name ="Shashank";
String lname = " s p ";
System.out.println("The first string length is "+name.length());
name = name.concat(lname);
System.out.println(name)
System.out.println("Question : 3");
Integer num = 9807;
String snum = num.toString();
System.out.println(snum);

System.out.println("Question : 4");
String extract = "Welcome to bmscse college";
char chs[] = new char[20];
extract.getChars(10,16,chs,0);
String ans = new String(chs);
System.out.println(ans);

System.out.println("Question : 5");
String myName = "Shashank";
char charArray[] = myName.toCharArray();
for(char val: charArray){
System.out.print(val+"\t");
}
System.out.println();
byte arr[] = myName.getBytes();
for(byte val: arr){
System.out.print(val+"\t");
}
System.out.println();
```

```
System.out.println("Question : 6");

System.out.println("Bmsce".equals("Bmsce")) ;
System.out.println("Bmsce".equals("College")) ;
System.out.println("Bmsce".equals("BMSCE")) ;
System.out.println("Bmsce".equalsIgnoreCase("BMSCE")) ;

}

}

class Sorts{

public static void main(String args[]){

System.out.println("b".compareTo("a"));

String st[] = {"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<st.length();i++) {

for(int j =0;j<st.length();j++) {

if(st[i].compareTo(st[j])==1) {

String temp = st[i];

st[i] = st[j];

st[j] = temp;

}

}

}

for(String c:st){

System.out.println(c);

}

}

}
```

```

import java.util.*;
class Comparestrings{
    public static void main(String args[]){

        String subString = "Bmsce collge";
        String val = "Welcome to Bmsce College of Engineering";
        if(val.regionMatches(11,subString,0,5)){
            System.out.println("String matches");
        }
        else{
            System.out.println("String not matches");
        }
        System.out.println(subString.startsWith("B"));
        System.out.println(subString.startsWith("r"));

        String a = "Hello";
        String b = "Hello";
        String c = new String("Hello");
        System.out.println(a==b);
        System.out.println(b==c);
        System.out.println(b.equals(c));
    }
}

```

Lab program 6 -b packages

Create a package CIE which has two classes- Student and Internals.
The class Student has

members like usn, name, sem. The class Internals derived from
Student 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;
import java.util.Scanner;
public class Student {
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
    public void inputStudentDetails() {
        System.out.println("Enter the usn , name , sem :\n");
        Scanner sc = new Scanner(System.in);
        this.usn = sc.nextLine();
        name = sc.nextLine();
        sem = sc.nextInt();
    }
    public void displayStudentDetails() {
        System.out.println("Student details are \n Name : name \n Usn : usn
\n
Sem : sem \n");
    }
}
```

CIE/Internals.java

```
package CIE;
import java.util.Scanner;
public class Internals extends Student {
    protected int marks[] = new int[5];
    public void inputCIEmarks() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the marks of 5 subject");
        for(int i =0;i<5;i++){
            marks[i] = sc.nextInt();
        }
    }
}
```

```
}

}

}

SEE/Externals.java

package SEE;

import CIE.*;

import java.util.Scanner;

public class Externals extends Internals {

    protected int marks[];

    protected int finalMarks[];

    public Externals() {

        marks = new int[5]; finalMarks = new int[5];

    }

    public void inputSEEmarks() {

        Scanner 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]/2 + super.marks[i];

        }

    }

    public void displayFinalMarks() {

        displayStudentDetails();

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

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

        }

    }

}
```

```

Main.java import SEE.Externals;
import CIE.*;
class Main {
    public static void main(String args[]){
        int numOfStudents = 2;
        Externals finalMarks[] = new Externals[numOfStudents];
        for(int i=0;i<numOfStudents;i++) {
            finalMarks[i] = new Externals();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE marks");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE marks");
            finalMarks[i].inputSEEmarks();
        }
        System.out.println("Displaying data:\n");
        for(int i=0;i<numOfStudents;i++) {
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
        System.out.println("Shashank S P");
        System.out.println("1BM22CS256");
    }
}

```

Lab program 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 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.*;
```

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

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

class Father extends InputScanner{
    int fatherAge;
    public Father() throws WrongAge{
        InputScanner ss = new InputScanner();
        System.out.println("Enter the father age: ");
        fatherAge = ss.sc.nextInt();
        if(fatherAge<0){
            throw new WrongAge("Age cannot be negative");
        }
    }
    void fdisplay(){
        System.out.println("Father age is : "+fatherAge);
    }
}

class Son extends Father{
    int sonAge;
    public Son() throws WrongAge{
        InputScanner ss = new InputScanner();
        System.out.println("Enter the Son age: ");
        sonAge = ss.sc.nextInt();
        if(sonAge==fatherAge){
```

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

else 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");

}

void sdisplay(){

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

}

}

public class PMain{

public static void main(String args[]){

Son p;

try{

p = new Son();

p.fdisplay();

p.sdisplay();

}

catch(WrongAge e){

System.out.println(e);

}

System.out.println("Shashank S P");

System.out.println("1BM22CS256");

}

}
```

Lab program 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.

```
class Bms extends Thread {  
    public void run() {  
        for (int i = 1; i <= 50; i++) {  
            try {  
                Thread.sleep(10000);  
                System.out.println("BMS College of Engineering" + i);  
            } catch (InterruptedException e) {  
                System.out.println("thread error");  
            }  
        }  
    }  
}  
  
class Cse extends Thread {  
    public void run() {  
        for (int i = 1; i <= 50; i++) {  
            try {  
                Thread.sleep(2000);  
                System.out.println("Computer Science " + i);  
            } catch (InterruptedException e) {  
                System.out.println("thread error");  
            }  
        }  
    }  
}  
  
public class TreadsMain {  
    public static void main(String args[]) {  
        Bms c1 = new Bms();  
        c1.start();  
        Cse i1 = new Cse();  
    }  
}
```

```
    i1.start();  
}  
  
System.out.println("Shashank S P");  
System.out.println("1BM22CS256");  
}  
  
Lab program 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

Arithmeθc Excepθon Display the excepθon 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 divident:");  
  
        // 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 :)
// to display error bois
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(err);
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(ArithmetcException 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();
}
}
```

```

}) ;

System.out.println("Name : Shashank S P");
System.out.println("USN : 1BM22CS256");
}

}

Lab program 10

Demonstrate Inter process Communication and deadlock

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("InterruptedExceptioncaught");

            }

        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++;
}
}
```

```
}

}

}

public class Corrected {

    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
        System.out.println("Shashank SP");
        System.out.println("1BM22CS256");
    }
}
```