

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



OBJECT ORIENTED JAVA PROGRAMMING

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

**SANJEEET PRAJWAL PANDIT
1BM22CS241**

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

Date	Java LAB	Sign	Date.	
(1) 5/12/23	Sample Programs	Han		
(2) 12/12/23	Program 1	12/12/23		
(3) 19/12/23	Program 2	12/12/23		
(4) 26/12/23	Program 3	12/12/23		
(5) 2/1/24	Program 4	1/1/24		
(6) 9/1/24	Program 5	9/1/24		
(7) 16/1/24	Program 5	16/1/24		
(8) 23/1/24	Program 6	23/1/24		
(9) 30/1/24	Program 7	30/1/24		
(10) 6/2/24	Program 8	6/2/24		
(11) 13/2/24	Program 10 (i) & (ii)	13/2/24	6-2-24	CSC
(12) 20/2/24	Program 9	20/2/24		

Car
15-2-24
Car
21-2-24

- LAB 2
- (1) Demonstrate `parseInt()` → returns int egmt of the string
 - (2) Area of Rectangle
 - (3) Scanner class
 - (4) 1D, 2D Arrays
 - (5) Fact of a no.
 - (6) Palindrome
 - (7) Sum of digits

~~(8) Develop a Java prog. that prints all real solns~~

~~(1) Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read a,b,c & use the quadratic formula. If discriminant b^2-4ac is negative. Display a message stating that there are no real solutions.~~

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

```
class quadratic
```

```
{
```

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

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

```
    void getd()
```

```
{
```

```
    Scanner 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();
```

}

$$d = b^2 - 4 \times a \times c;$$

if ($d == 0$)

{

$$r_1 = (-b) / (2 \times a);$$

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

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

}

else if ($d > 0$)

{

$$r_1 = ((-b) + (\text{Math.sqrt}(d))) / (\text{double})(2 \times a);$$

$$r_2 = ((-b) - (\text{Math.sqrt}(d))) / (\text{double})(2 \times a);$$

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");

$$r_1 = (-b) / (2 \times 0);$$

$$r_2 = (\text{Math.sqrt}(-d)) / (2 \times a);$$

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

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

}}

class QuadraticMain

{

public static void main (String args [])

Quadratic q = new Quadratic();

q.getd();

q.compute();

Output :-

(i) Enter the coefficients of a, b, c :

1 2 3

Roots are imaginary

$$\text{Root}_1 = -1.0 + i1.414213562373095i$$

$$\text{Root}_2 = -1.0 - i1.414213562373095i$$

(ii) Enter the coefficients of a, b, c :

1 -2 1

Roots are real and equal

$$\text{Root}_1 = \text{Root}_2 = 1.0$$

(iii) Enter the coefficients of a, b, c :

1 -3 2

Roots are real and distinct

$$\text{Root}_1 = 2.0 \quad \text{Root}_2 = 1.0$$

USN: 1BM22CS243

Name: Sayjet P. Pandit

1
2
3
4
5
6
7
8
9
0
.

LNB 3

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

Ans

$$\text{SGPA} = \frac{\sum (\text{course credit}) (\text{grade pkr})}{\sum (\text{course credits})}$$

(considering all courses in that sem including those with F grade)

$$\text{CGPA} = \frac{\sum (\text{course credits}) (\text{grade pkr})}{\sum (\text{course credits})}$$

(considering all courses until that sem excluding F grade)

Q Area of rectangle

```
import java.util.*;
public class Rectangle{
    public static void main (String args[])
    {
        int len, breadth;
        len = Integer.parseInt(args[0]);
        breadth = Integer.parseInt(args[1]);
        int area = len * breadth;
        System.out.println ("Length = " + len);
        System.out.println ("Breadth = " + breadth);
        System.out.println ("Area of Rectangle = " + area);
    }
}
```

O/P:

Java args 23 20

length = 23

breadth = 20

Area of Rectangle = 460

(2) Scanner:

```

import java.util.Scanner;
class Hello {
    public static void main (String args[]) {
        int a; float b; String s;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter a string");
        s = sc.nextLine();
        System.out.println ("String is " + s);
        System.out.println ("Enter an integer:");
        a = sc.nextInt();
        System.out.println ("Your entered integer is " + a);
        System.out.println ("Enter a float:");
        b = sc.nextFloat();
        System.out.println ("Your entered float is " + b);
    }
}

```

⇒ Enter string

Sanjeet

String is Sanjeet

Enter an integer

5
your entered integer is 5

Enter a float

1.23

Your entered float is 1.23

(3) Array (a) 1D:

```

import java.util.*;
class Array {
    public static void main (String args[]) {
        int days[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 21};
        System.out.println ("March has " + days[2] + " days");
    }
}

```

⇒ March has 31 days.

(9) Factorial :

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

```
class Factorial {
```

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

```
{    int fact=1; System.out.println ("Enter a num :");  
    Scanner sc = new Scanner (System.in); int n = sc.nextInt();  
    for (int i=0; i<n; i++) { fact=fact*i; }
```

```
    System.out.println ("The factorial is " + fact);
```

```
}
```

→ Enter a number: 5 → The factorial is 120.

LAB 3B

(Subject P. Pandit - IBM22(S241))

```
import java.util.Scanner;  
class Subject {  
    int SubjectMarks;  
    int credits;  
    int grade;  
}
```

```
class Student {  
    String name;  
    String USN;  
    double SGPA;  
    Scanner S;  
    Subject Subject[];  
    Student() {  
        int i;  
        Subject = new Subject[9];  
        for (i = 0; i < 9; i++) {  
            Subject[i] = new Subject();  
        }  
        S = new Scanner(System.in);  
    }
```

```
void getStudentDetails() {  
    System.out.println("Enter your name:");  
    name = S.nextLine();  
    System.out.println("Enter your USN:");  
    USN = S.nextLine();  
}
```

```
void getMarks() {  
    int i;  
    for (i = 0; i < 8; i++) {  
        System.out.print("Enter marks and credit for  
course " + (i + 1) + ":");  
        System.out.print("Marks:");  
    }
```

```
int marks = s.nextInt();
System.out.println("Qualif:");
int credits = s.nextInt();
subject[i].subjectMarks = marks;
subject[i].credit = credits;
subject[i].grade = (subject[i].subjectMarks) / (10);
if (subject[i].grade == 11) {
    subject[i].grade = 10;
}
if (subject[i].grade <= 4) {
    subject[i].grade = 0;
}
}
void computeSGPA() {
    int total_credits = 0;
    int i;
    for (i = 0; i < 8; i++) {
        total_credits += subject[i].credit;
    }
    int totalGradeAndCredit = 0;
    for (i = 0; i < 8; i++) {
        totalGradeAndCredit += subject[i].credit * subject[i].grade;
    }
    double SGPA = (float) totalGradeAndCredit / total_credits;
    System.out.println("Your SGPA is " + SGPA);
}
```

(Class Main {

```
public static void main (String args[]) {  
    Student s1 = new Student();  
    s1.getStudentDetails();  
    s1.getMarks();  
    s1.computeSGPA();
```

}

}

O/P:

Enter your name:

Sanjeet

Enter your USN:

IBM22CS241

Enter the marks and credits for course 1:

Marks:

98

Credits:

4

Enter the marks and credits for course 2:

Marks:

98

Credits:

4

Enter the marks and credits for course 3:

Marks:

93

Credits:

3

Enter the marks and credits for course 4:

Marks:

99

Credits:

3

Enter the marks and credits for course 5:

marks:

94

credit:

3

Enter the marks and credits for course 6:

marks:

91

credit:

1

Enter the marks and credits for course 7:

marks:

93

credit:

1

Enter the marks and credits for course 8:

marks:

99

credit:

1

your SGPA is 10.0

19-11-2022

6/12/23
(LAB 3) Create a class Book which contains 4 members: name, number, price, numPages. Include a constructor to set values for members. Include methods to get & get details of the objects. Include a toString() method that could display the complete details of 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 = "No. of pages: " + this.numPages + "\n";
```

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

```
}
```

```
String getName() {
    this.name = name;
}

String getAuthor() {
    this.author = author;
}

String getPrice() {
    this.price = price;
}

String getNumPages() {
    this.numPages = numPages;
}
```

class main {

```
public static void main (String args[]) {
    Scanner s = new Scanner (System.in);
    int n, price, numPages;
    String name, author;
```

```
System.out.println ("Enter no. of books : ");
n = s.nextInt();
s.nextLine();
```

Books b[];

```
b = new Books [n];
```

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

```
System.out.print ("Enter name of book " + (i+1) + ": ");
name = s.nextLine();
```

```
System.out.print ("Enter author of book " + (i+1) + ": ");
author = s.nextLine();
```

```
System.out.println ("Enter price of Book " + (i+1) + ":");
price = scanner.nextInt();
System.out.println ("Enter no. of pages of book " + (i+1) + ":");
numPages = scanner.nextInt();
scanner.nextLine();
b[i] = new Books (name, author, price, numPages);
}
for (int i=0; i<n; i++) {
    String booksDetails = b[i].toString();
    System.out.println (booksDetails);
}
for (int i=0; i<n; i++) {
    System.out.println ("Book " + (i+1) + ":" );
    System.out.println ("Name : " + b[i].getname());
    System.out.println ("Author : " + b[i].getAuthor());
    System.out.println ("Price : " + b[i].getPrice());
    System.out.println ("No. of Pages : " + b[i].getNumPages());
    System.out.println ("*****");
}
System.out.println ("Sanyeeet P. Pandit \n IBM22(1242);
```

O/P:

Enter no. of books

2

Enter the name of book 1:

Da Vinci Code

Enter Author of book 1:

Dan Brown

Enter price of book 1:

499

Enter no. of pages of book 1:

350

Enter the name of book 2:

Harry Potter

Enter the price of book 2:

JK Rowling

Enter price of book 2:

499

Enter no. of pages of book 2:

500

Book 1:

Name : Da Vinci Code

Author : Dan Brown

Price : 499

No. of Pages : 350

WAB
12/13
75

Book 2:

Name : Harry Potter

Author : JK Rowling

Price : 499

No. of Pages : 500

Sanjeev P. Pandit

ISBN 97893862341

~~QUESTION~~

Quadratic Eqn

```
import java.util.*;  
class  
(2/1/2024)
```

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named `getArea()`.

Provide three classes Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method `getArea()` that prints the area of the given shape.

Steps

(1)

```
import java.util.*;  
class InputScanner {  
    Scanner sc;  
    public InputScanner() {  
        sc = new Scanner(System.in);  
    }  
}
```

abstract class Shape extends InputScanner

```
    double a, b;  
    abstract void getInput();  
    abstract void displayArea();  
}
```

class Rectangle extends Shape {

```
    void getInput() {  
        System.out.println("Enter length and breadth of Rectangle");  
        a = sc.nextDouble();  
        b = sc.nextDouble();  
    }
```

```
void displayarea() {
```

```
    System.out.println("Area of Rectangle is:");
```

```
    System.out.format("%..2f\n", a * b);
```

```
}
```

```
class Triangle extends Shape {
```

```
    void getinput() {
```

```
        System.out.println("Enter height and base of triangle");
```

```
        a = sc.nextDouble();
```

```
        b = sc.nextDouble();
```

```
}
```

```
    void displayarea() {
```

```
        System.out.println("Area of triangle is:");
```

```
        System.out.format("%..2f\n", 0.5 * a * b);
```

```
}
```

```
class Circle extends Shape {
```

```
    void getinput() {
```

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

```
        a = sc.nextDouble();
```

```
}
```

```
    void displayarea() {
```

```
        System.out.println("Area of circle is:");
```

```
        System.out.format("%..2f\n", 3.14 * a * a);
```

```
}
```

public class Area {

 public static void main (String args[]) {

 Shape ref;

 ref = new Rectangle();

 ref.getinput();

 ref.displayarea();

 ref = new Triangle();

 ref.getinput();

 ref.displayarea();

 ref = new Circle();

 ref.getinput();

 ref.displayarea();

}

}

Output:

Enter the length and breadth of Rectangle :

4

3

Area of Rectangle is :

12.00

Enter the height and base of Triangle :

4

6

Area of Triangle is :

12.00

Enter the radius of Circle :

3

Area of Circle is :

28.26

9/1/24

Page No.

Date. / /

- Q Develop a Java program to develop a class Bank with current account & savings account.

```
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, cannot withdraw");  
    }  
    void display ()  
    {  
        System.out.println ("Name: " + name + " Account No.: " + accno +  
                           " Type: " + type + " Balance: " + balance);  
    }  
}
```

class Savings extends Account

```
{  
    private static double rate = 5;  
    Savings (String name, int accno, double balance)  
    {  
        super (name, accno, "Savings", balance);  
    }  
}
```

void interest()

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

class Current extends Account

```
{  
    private double minbal = 500;  
    private double servicecharges = 50;  
    Current (String name, int accno, double balance)  
    {  
        super (name, accno, "Current", balance);  
    }  
}
```

void checkmin()

```
{  
    if (balance < minbal)  
    {  
        System.out.println ("Balance is less than the minimum  
        balance, thus service charges imposed: " + servicecharges);  
        balance = servicecharges;  
        System.out.println ("Balance is: " + balance);  
    }  
}
```

class bank {

public static void main (String args [])

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

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

String name = sc.nextLine();

System.out.println ("Enter account No.:");

int accno = sc.nextInt();

System.out.print ("Enter Initial Balance:");

double balance = sc.nextDouble();

int ch;

double amt1, amt2;

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

savacc sav = new savacc (name, accno, type, balance);

curracc cur = new curracc (name, accno, balance);

while (true)

{

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

{

System.out.print ("Menu\n 1. Deposit\n 2. Withdraw")

3. Compute the interest 4. Display");

System.out.print ("Enter the choice:");

ch = sc.nextInt();

switch (ch)

{

case 1: System.out.println ("Enter the amount:");

amt1 = sc.nextInt();

sav.deposit (amt1);

break;

case 2 : System.out.println (" Enter the amount: ");

amt2 = sc.nextInt();

sav.withdraw (amt2);

break;

case 3 : sav.interest ();

break;

case 4 : sav.display ();

break;

case 5 : System.exit(0);

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

break;

}

}

else

System.out.println (" Menu)\n 1. Deposit 2. withdraw

3. Display");

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

ch = sc.nextInt();

switch (ch)

case 1 : System.out.println (" Enter amount: ");

amt1 = sc.nextInt();

cur.deposit (amt1);

break;

case 2 : System.out.println (" Enter amount: ");

amt2 = sc.nextInt();

cur.withdraw (amt2);

cur.checkMin();

break;

case 3 : cur.display ();

break;

case 4 : System.exit(0);

}

Output: (Sanjeet P. Pandit / 1BM22CS241)

Enter the name: ~~Sanjeet~~ Sanjeet

Enter the type: (Current/Savings):

Current

Enter the account number:

1009

Enter the initial balance:

2000

Menu:

1. Deposit 2. Withdraw 3. Display

1

Enter the amount: 1000

Menu:

1. Deposit 2. Withdraw 3. Display

2.

Enter the amount: 500

Menu:

1. Deposit 2. Withdraw 3. Display

3.

Name: Sanjeet Name: ~~Sanjeet~~ Account no: 1009 type: Current Balance : 2500.0

Open
File
Edit
19

1/24 Q write a Java program using generics, show stack class for 5 integers and 5 double values. (Create generic class Stack)

(Sandeep P. Pandit, IBM22CS24I)

Q Strings

// 1

O/P: (*) abcd

cd // 2

(*) char c[] = {'A', 'B', 'C'};

(*) String c_obj = new String(c);

System.out.println(c_obj.length()); // 3

String batch = "22-26";

String mys = "Sandeep belongs to "+batch+" batch";

System.out.println(mys); // Subject belongs to 22-26 batch

Q (Generics) (Additional)

(*) output: // 1 2 3 4 5 in integer stack & 1.1 2.2 3.3 4.4 5.5 in double stack.

Popping integers from the stack:

5

4

3

2

1

Popping doubles from the stack:

5.5

4.4

3.3

2.2

1.1

113

→ (*) 9807

114

→ (*) BMSC E

115

→ (*) abc

97 98 99

116

→ (*) true

false

false

true

117

→ (*) Substring is Matched

118

→ (*) true

false

119

110

→ (*) true

true

false

false

111

→ (*) opple	orange
ball	pant
cat	queen
dog	sing
ent	stay
free	tree
gun	umbrella
hen	van
ice	watch
jug	Xwas
kite	yatch
lift	zee
non	
net	

//13 This is a test. This is, too

//14 hello world

//15 BMSCe Commge

//16 "Hello friends"

generics :

//17 write a Java Program to create generic class Stack
which holds s integers & s double values

O/P:

Popping integers from the Stack:

5

4

3

2

1

Popping doubles from the Stack:

5.5

4.4

3.3

2.2

1.1

// 12

1 2 3 4 5 6 7 8 9 10

// 17

Student 1

name: A
RegNo: 1
Sem: 3
CGPA: 9.8

Student 2

name: B
RegNo: 2
Sem: 3
CGPA: 9.8

11 | 18

Chay at 3 is '0'

dhoni

awake; in bed

1119

~~eagle~~ is flying

edge makes sound

11

20

Calcarea : 28.26

calc peri: 18 84

Total eq : 40

Totperi : 83

(Additional code for generics):

→ import java.util.*;

```
class Stack <E> {
```

E SHK();

int top;

int size = 10;

Stack() {

`strk = (E[J] new) object [size];`

$$\phi_D = -1;$$

1

void push(E item) {

if (top == size - 1) {

`System.out.println("No elements can be added further,
overflow");`

3

$$Stk[+top] \approx 9km;$$

```
E.pop() {  
    if (top < 0) {  
        System.out.print("No element can be deleted further, underflow");  
        return null;  
    }  
    else {  
        return STK[top--];  
    }  
}
```

```
public class Main {  
    public static void main (String args[]) {  
        Stack<Integer> S1 = new Stack<Integer>();  
        Stack<Double> S2 = new Stack<Double>();  
        Scanner sc = new Scanner (System.in);  
        System.out.println("Enter elements in integer stack");  
        for (int i=0; i<5; i++) {  
            int n1 = sc.nextInt();  
            S1.push(n1);  
        }  
        System.out.println("Enter elements in Double stack");  
        for (int i=0; i<5; i++) {  
            double n2 = sc.nextDouble();  
            S2.push(n2);  
        }  
        System.out.println("Elements of S1"); System.out.println("Popping  
        integer from stack");  
        for (int i=0; i<5; i++) {  
            System.out.println(S1.pop());  
        }  
        System.out.println("Popping doubles from the stack");  
        for (int i=0; i<5; i++) {  
            System.out.println(S2.pop());  
        }  
        sc.close();  
    }  
}
```

23/1/24

No. / /
Date. / /

- Q Create a package CIE which has two classes - Student and Intervals. The class Student has members like usn, name, sem. The class Intervals derived from Student has an array that stores the internal marks scored in 5 courses of current semester of student. Create another package SEE which has class External which is a derived class of Student. This class has an array that stores the SEE marks scored in 5 courses of current Semester of the student. Import 2 packages in a file that calculates the final marks of n students in all 5 courses.

A // Student.java file

```
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() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = s.next();
        System.out.println("Enter name: ");
        name = s.next();
        System.out.println("Enter Sem: ");
        sem = s.nextInt();
    }
}
```

```
public void inputdisplayStudentDetails()
```

```
{  
    System.out.println("USN:" + USN);  
    System.out.println("Name:" + name);  
    System.out.println("Semester:" + sem);  
}
```

```
} // Internals.java file
```

```
package CIE;  
import java.util.Scanner;
```

```
public class Internals extends Student
```

```
{  
    protected int marks[] = new int [5];
```

```
    public Internals()  
    {
```

```
        public void input(CIEMarks())
```

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

```
    System.out.println("Enter internal marks for " + name);  
    for(int i=0; i<5; i++)
```

```
{  
    System.out.print("Subject " + (i+1) + " Marks: ");
```

```
    marks[i] = s.nextInt();  
}
```

```
}
```

```
}
```

// External's java file

```
package SEE;
import CIE.Tutorials;
import java.util.Scanner;
public class External's extends Tutorials
{
    protected int marks[];
    protected int finalmarks[];
    public External()
    {
        marks = new int[5];
        finalmarks = new int[5];
    }
}
```

public void inputSEEmarks()

```
{ Scanner s = new Scanner(System.in);
System.out.println("Enter SEE marks for " + name);
for (int i=0; i<5; i++)
{
    System.out.println("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 file
import SEE.External;
public class Main {
    public static void main (String args[])
    {
        int numofStudents = 2;
        External finalMarks[] = new External [numofStudents];
        for (int i=0; i<numofStudents; i++)
        {
            finalMarks[i] = new External();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE Marks:");
            finalMarks[i].inputCIEMarks();
            System.out.println("Enter IEF Marks:");
            finalMarks[i].inputIEFMarks();
            System.out.println("Displaying data:");
            for (int i=0; i< numofStudents; i++)
            {
                finalMarks[i].calculateFinalMarks();
                finalMarks[i].displayFinalMarks();
            }
        }
    }
}
```

Output:

Enter USN: 1

Enter Name: A

Enter Semester: 3

Enter CIE Marks:

Enter internal Marks for A

Subject 1 Marks: 50

Subject 2 Marks: 50

Subject 3 Marks: 40

Subject 4 Marks: 50

Subject 5 Marks: 40

Enter SEE marks for A

Subject 1 Marks: 90

Subject 2 Marks: 92

Subject 3 Marks: 94

Subject 4 Marks: 96

Subject 5 Marks: 98

Enter USN: 2

Enter Name: B

Enter Semester: 3

Enter CIE Marks:

Enter internal Marks for B

Subject 1 Marks: 33

Subject 2 Marks: 36

Subject 3 Marks: 23

Subject 4 Marks: 27

Subject 5 Marks: 40

Subject SEE marks for B

Subject 1 Marks : 89

Subject 2 Marks : 90

Subject 3 Marks : 98

Subject 4 Marks : 92

Subject 5 Marks : 90

Displaying data :

USN: 1

Name: A

Semester: 3

Subject 1: 95

Subject 2: 96

Subject 3: 87

Subject 4: 98

Subject 5: 89

USN: 2

Name: B

Semester: 3

Subject 1: 77

Subject 2: 81

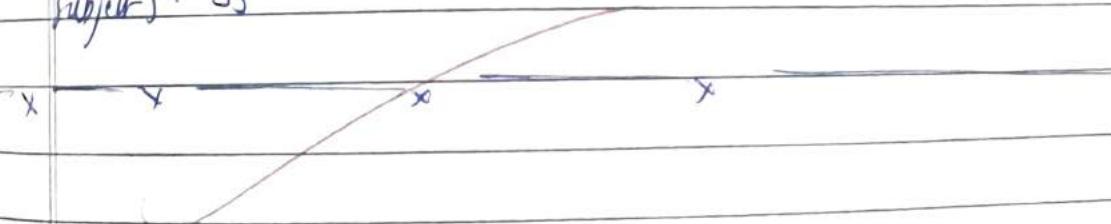
(Sanyog P. Pandit)

Subject 3: 72

(IBM22(S241))

Subject 4: 73

Subject 5: 85



~~11/11/2024~~

30/1/24

Q WAP that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class "Son" which extends 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 calls both father and son's age and throws an exception if son's age is \geq father's age.

```

→ import java.util.Scanner;
class Wrongage extends Exception {
    public Wrongage (String message) {
        super (message);
    }
}
class InputScanner {
    protected Scanner sc;
    public InputScanner () {
        sc = new Scanner (System.in);
    }
}

```

```

class Father extends InputScanner {
    protected int fatherage;
    public Father () throws Wrongage {
        System.out.println ("Enter father's age: ");
        fatherage = sc.nextInt ();
        if (fatherage < 0) {
            throw new Wrongage ("Age can't be negative");
        }
    }
}

```

```
public void display() {
    System.out.println("Father's age is: " + fatherage);
}

class Son extends Father {
    private int sonage;
    public Son() throws Wrongage {
        super();
        System.out.println("Initial son's age : ");
        sonage = sc.nextInt();
        if (sonage >= fatherage) {
            throw new Wrongage ("Son's age can't be greater or
equal to his father's age");
        }
    }

    else if (sonage < 0) {
        throw new Wrongage ("Age can't be negative");
    }
}

public void display() {
    System.out.println("Son's age is: " + sonage);
}

public class FatherOnAge {
    public static void main(String args[]) {
        try {
            Son son = new Son(),
            son.display();
        } catch(Wrongage e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Output:

① Enter father's age:

50

Enter son's age:

20

Father's age is: 50

Son's age is: 20

② Enter father's age

-50

Error: Age can't be negative

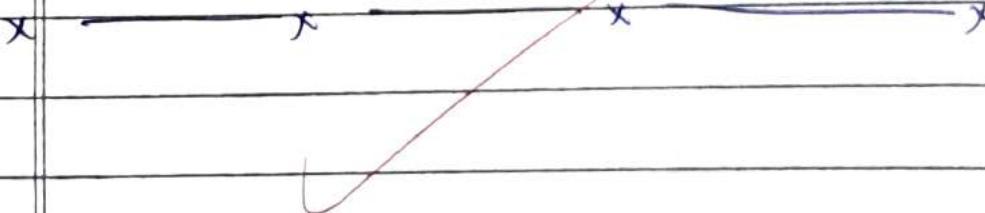
③ Enter father's age:

20

Enter son's age:

50

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



W.W.I.Y
3^o

6/2/21

Q Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every 10 sec and another displaying "CRE" once every 2 sec.

class DisplayMessageThread extends Thread {
 private final String message;
 private final long interval; // in ms

DisplayMessageThread (String message, long interval) {

this.message = message;

this.interval = interval;

}

public void run() {

try {

while(true) {

System.out.println(message);

Thread.sleep(interval);

}

}

catch (InterruptedException e) {

System.out.println(message); Thread.currentThread().getStackTrace();

"interrupted");

}

}

```

public class TwoThreadDemo {
    public static void main (String [] args) {
        DisplayMessageThread thread1 = new DisplayMessageThread ("BMS
college of engineering", 10000); //10s
        DisplayMessageThread thread2 = new DisplayMessageThread ("CSE", 2000); //2s
        thread1.setName ("Thread 1");
        thread2.setName ("Thread 2");
        thread1.start();
        thread2.start();
    }
    try {
        // let threads run for a while
        Thread.sleep(30000); // let the program run for 30s
    } catch (InterruptedException e) {
        System.out.println ("Main thread interrupted");
    }
    // interrupt both threads to stop them
    thread1.interrupt();
    thread2.interrupt();
    System.out.println ("Main thread exiting");
}

```

Output:

GMS College of Engineering

CSE

CSE

CSE

CTE

CSE

BMS College of Engineering

CSE

CTE

CSE

CSE

CTE

BMS College of Engineering

CSE

CTE

CSE

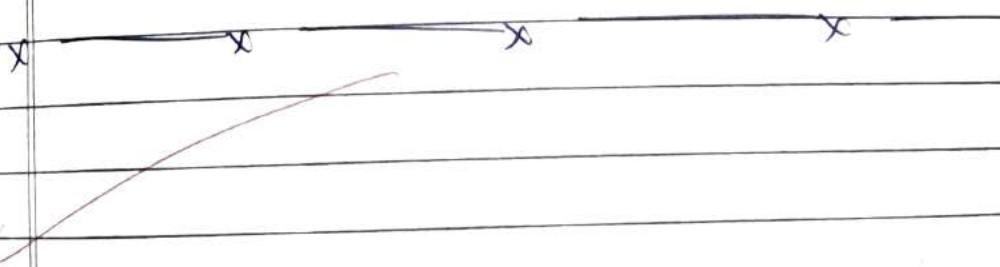
CSE

CSE

Main thread exiting

Thread interrupted

Thread interrupted



W²
b² z² z⁴

13/2/29

Ques (i) (IPC)

(ii) class Q {

int n;

boolean valueSet = false;

synchronized int get() {

{

while (!valueSet)

try {

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

wait();

}

catch (InterruptedException e) {

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

}

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

valueSet = false;

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

notify();

return n;

}

synchronized void put(int n) {

while (valueSet)

try {

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

wait();

}

catch (InterruptedException e) {

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

}

twi.n = n;

valueSet = true;

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

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

notify();

class Producer implements Runnable {

 Q q;

 Producers (Q q) {

 this.q = q;

 new Thread (this, "Producers").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 PCFixed {

 public static void main(String args[]) {

 Q q = new Q();

 new Producer(q);

 new Consumer(q);

 System.out.println ("Press control-c to stop");

}

}

Output:

Press control-c to stop

consumed: 3

Put: 0

put: 4

Intimate consumer

Intimate consumer

Producer waiting

got: 4

Got: 0

Intimate Producer

Intimate Producer

consumed: 4

Put: 1

Intimate consumer

Producer waiting

consumed: 0

got: 1

Intimate producer

consumed: 1

put: 2

Intimate consumer

Producer waiting

got: 2

Intimate producer

consumed: 2

put: 3

Intimate consumer

Producer waiting

got: 3

Intimate Producer

2022-2-24
X3-2

(iii) (Deadlock)

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.bar()");

b.out();

}

void out() {

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

}

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");

}

void out() {

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

}

class Deadlock implements Runnable {

A a = new A();

B b = new B();

Deadlock() {

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

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

t.start();

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

System.out.println("Back in Main Thread");

}

public void run() {

b.bar(a); // get lock on b in other thread.

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

}

public static void main (String args[]) {

new Deadlock();

}

Output:

MainThread entered A.foo

RacingThread entered B.bar

MainThread trying to call B.bar()

Inside A.foo

Back in Main Thread

RacingThread trying to call A.foo()

Inside A.foo

Back in other Thread

Nov 2-24
15

→ Write a program that creates a user interface to perform integer divisions. The user enters 2 nos in text fields, Num1 and Num2. The division of Num1 and 2 is displayed in the result field when the divide button is clicked. If Num1 or Num2 were not an integer. The program would show NumberFormatException if Num2 were zero, the program would throw an ArithmeticException display exception to message display box.

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

```
class SwingDemo
```

```
SwingDemo()
```

```
JFrame frm = new JFrame ("Divide Type");
```

```
frm.setSize(275,150);
```

```
frm.setLayout(new GridLayout());
```

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

```
JLabel lab = new JLabel ("Enter divisor and dividend");
```

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

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

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

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

```
alab = new JLabel();
```

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

```
onslab = new JLabel();
```

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

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

```
jfrm.add(ajtf);
```

```
jfrm.add(bjtf);
```

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

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

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

```
jfrm.add(onslab);
```

```

button.addActionListener(new ActionListener()) {
    public void actionPerformed(ActionEvent e) {
        calculateDivision();
    }
};

} // form.setVisible(true);

// ActionListener e = new ActionListener() {
private void calculateDivision() { // actionPerformed(ActionEvent e)
    try {
        int a = Integer.parseInt(tf1.getText());
        int b = Integer.parseInt(tf2.getText());
        if (b == 0) {
            throw new ArithmeticException("Division by zero.");
        }
        int ans = a/b;
        tf1.setText("A = " + a);
        tf2.setText("B = " + b);
        ansLab.setText("Ans = " + ans); // catch(NumberFormatException)
        err.setText(" ");
        displayErrorMessage("Enter values", "Warning");
    } catch (NumberFormatException e) {
        tf1.setText(" ");
        tf2.setText(" ");
        ansLab.setText(" ");
        err.setText("B should be non-zero!");
        displayErrorMessage("B should be non-zero!", "Warning");
    }
}

public static void main(String args[]) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDemo();
        }
    });
}
}

```

Output:

Enter the divisor and dividend

4 1

[calculate] A=4 B=1 Ans=1

Functions:

- (1) JFrame: It is a top level container in Java swing that represents a window with a title bar, border and optional menu bar.
- (2) setSize: It is used to set size of the frame.
- (3) setLayout: This line sets the layout manager for the frame to flowlayout which arranges components from left to right in a flow like manner.
- (4) add: This line adds the user label to the frame.
- (5) invokeLater: to perform task asynchronously in awt event dispatch.
- (6) setVisible: This line makes the frame visible.
- (7) setText: This line sets the text of 'n' label to display value of n.

WAP
2/2/14
2/1

Lab 1:

(1) Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions

```
import java.util.Scanner;
class Quadratic
{
    int a,b,c;
    double r1,r2,d;
    void getd()
    {
        Scanner 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();
        System.out.println("Sanjeet P. Pandit 1BM22CS241");
    }
}

```

Lab 2:

(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 MainSGPA{
    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("Sanjeet P. Pandit 1BM22CS241");
    }
}

```

Lab 3:

(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="No of Pages : "+this.numPages+"\n";
        return name + author + price + numPages;
    }

    String getName() {
        this.name=name;
    }

    String getAuthor() {
        this.author=author;
    }

    int getPrice(){
        this.price=price;
    }

    int getNumPages(){
        this.numPages=numPages;
    }
}
```

```

class MainB{
    public static void main(String args[]){
        Scanner s = new Scanner(System.in);
        int n,price,numPages;
        String name,author;

        System.out.println("Enter no of books :");
        n= s.nextInt();
        s.nextLine();

        Books b[];
        b= new Books[n];

        for(int i=0;i<n;i++){
            System.out.println("Enter the name of book"+(i+1)+":");
            name=s.nextLine();
            System.out.println("Enter Author of book"+(i+1)+":");
            author=s.nextLine();
            System.out.println("Enter price of book"+(i+1)+":");
            price=s.nextInt();
            System.out.println("Enter no of pages of
book"+(i+1)+":");
            numPages=s.nextInt();
            s.nextLine();
            b[i]=new Books(name,author,price,numPages);
        }

        for(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 "+(i+1)+":");
            System.out.println("Name :" + b[i].getName());
            System.out.println("Author :" + b[i].getAuthor());
            System.out.println("Price :" + b[i].getPrice());
            System.out.println("No of pages :" + b[i].getNumPages());
        }

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

    System.out.println("Sanjeet P. Pandit 1BM22CS241");
}

```

Lab 4:

(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 sc=new InputScanner();
        System.out.println("Enter the length and breadth of rectangle");
        a=sc.s.nextInt();
        b=sc.s.nextInt();
    }
    void displayArea(){
        double area_rect=a*b;
        System.out.println("Area of retangle is : "+area_rect);
    }
}

class Triangle extends Shape{
    void getInput(){
        InputScanner sc=new InputScanner();
        System.out.println("Enter the base and height of triangle:");
        a=sc.s.nextInt();
        b=sc.s.nextInt();
    }
    void displayArea(){
```

```

        double area_tri=a*b/2;
        System.out.println("Area of tritangle is : "+area_tri);
    }
}

class Circle extends Shape{
    void getInput(){
        InputScanner sc=new InputScanner();
        System.out.println("Enter the radius of circle:");
        a=sc.s.nextInt();

    }
    void displayArea(){
        double area_circle=3.14*a*a;
        System.out.println("Area of circle is : "+area_circle);
    }
}

public class AbstractMain{
    public static void main(String args[]){
        Rectangle a = new Rectangle();
        a.getInput();
        a.displayArea();

        Triangle b = new Triangle();
        b.getInput();
        b.displayArea();

        Circle c = new Circle();
        c.getInput();
        c.displayArea();

        System.out.println("Sanjeet P. Pandit 1BM22CS241");
    }
}

```

Lab 5:

(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.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 amt) {
        balance=balance+amt;
    }

    void withdraw(double amt) {
        if(balance<amt) {
            System.out.println("Insufficient Balance");
        }
        else{
            balance=balance-amt;
        }
    }

    void display(){
        System.out.println("Name:"+name+"\tAccount
No:"+accNo+"\tAccount Type:"+type+"\tBalance"+balance);
    }
}

class Savings_acc extends Account{
    private static double rate= 3.5;

    Savings_acc(String name,int accNo,double balance) {
        super(name,accNo,"savings",balance);
    }

    void calInt(){
        double interest=(balance*rate)/100;
        System.out.println("Interest is "+interest);
    }
}

class Current_acc extends Account{
    private double minBal=500;
    double s_charges=50;

    Current_acc(String name,int accNo,double balance) {

```

```

        super(name,accNo,"current",balance);
    }

void check_bal(){
    if(balance<minBal){
        System.out.println("Insufficient Balance");
        balance=balance-s_charges;

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

public class Bank{
    public static void main(String args[]){
        String name;
        int AccNo;
        String Type;
        double init_bal;

        Scanner s=new Scanner(System.in);
        System.out.println("Enter Customer Name:");
        name=s.nextLine();
        System.out.println("Enter Account No:");
        AccNo=s.nextInt();
        System.out.println("Enter Account Type:");
        Type=s.next();
        System.out.println("Enter Initial Balance:");
        init_bal=s.nextDouble();
        double amt;
        Account a=new Account(name,AccNo,Type,init_bal);
        Savings_acc sv=new Savings_acc(name,AccNo,init_bal);
        Current_acc ca=new Current_acc(name,AccNo,init_bal);

        while(true){
            if(Type.equalsIgnoreCase("savings")){
                System.out.println("-----MENU-----");
                System.out.println("Enter 1:Deposit  2:Withdraw
3:Interest 4:Display Details 5:Exit");
                int ch=s.nextInt();
                switch(ch){
                    case 1:
                        System.out.println("Enter The
Amount:");
                        amt=s.nextDouble();
                        a.deposit(amt);
                        break;
                    case 2:
                        System.out.println("Enter the
withdrawing amount");
                        amt=s.nextDouble();
                        a.withdraw(amt);
                }
            }
        }
    }
}
```

```

        break;
    case 3:
        sv.calInt();
        break;
    case 4:
        a.display();
        break;
    case 5:
        System.exit(0);
    default:
        System.out.println("Invalid Choice");
    }

}

else{
    System.out.println("----MENU----");
    System.out.println("Enter 1:Deposit  2:Withdraw
3:Display Details 4:Exit");
    int ch=s.nextInt();

    switch(ch){
        case 1:
            System.out.println("Enter The
Amount:");
            amt=s.nextDouble();
            a.deposit(amt);
            break;
        case 2:
            System.out.println("Enter the
withdrawing amount");
            amt=s.nextDouble();
            a.withdraw(amt);
            ca.check_bal();
            break;
        case 3:
            a.display();
            break;
        case 4:
            System.exit(0);
        default:
            System.out.println("Invalid Choice");
    }
}
System.out.println("Sanjeet P. Pandit 1BM22CS241");
}
}

```

Lab 6:

(6) 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.

```
package CIE;
import java.util.*;
public class Student{
    protected String usn=new String();
    protected String name =new String();
    protected int sem;

    public void inputStudentDetails(){
        Scanner s=new Scanner(System.in);
        this.usn=s.nextLine();
        this.name=s.nextLine();
        this.sem=s.nextInt();
    }

    public void displayStudentDetails(){
        System.out.println(this.usn+" "+this.name+" "+this.sem);
    }
}

package CIE;
import java.util.Scanner;
public class Internals extends Student{
    protected int marks[]={new int[5]};
    public void inputCIEmarks(){
        Scanner s=new Scanner(System.in);
        for(int i=0;i<5;i++){
            marks[i]=s.nextInt();
        }
    }
}

package SEE;
import CIE.Internals;
```

```

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]);
    }
}

import SEE.*;

class Main1{
    public static void main(String args[]){
        int num=2;
        Externals finalMarks[] = new Externals[num];
        for(int i=0;i<num;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<num;i++){
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}

```

```

        System.out.println("Sanjeet P. Pandit 1BM22CS241");

    }
}

```

Lab 7:

(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.Scanner;

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

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

class Father extends InputScanner{
    int fatherAge;
    public Father() throws WrongAge{
        InputScanner sf=new InputScanner();
        fatherAge=sf.sc.nextInt();
        if(fatherAge<0){
            throw new WrongAge("Age cannot be negative");
        }
    }
    void Fdisplay(){
        System.out.println("Father's Age: "+fatherAge);
    }
}

class Son extends Father{
    int sonAge;
    public Son() throws WrongAge{
        InputScanner ss=new InputScanner();
        sonAge=ss.sc.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");
        }
    }

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

public class AgeCheck{
    public static void main(String args[]){
        Son a;
        try{
            a=new Son();
            a.Fdisplay();
            a.Sdisplay();
        }
        catch(WrongAge e){
            System.out.println(e);
        }
        System.out.println("Sanjeet P. Pandit 1BM22CS241");
    }
}

```

Lab 8:

(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 Thread1 implements Runnable{
    Thread t;
    public Thread1()
    {
        t=new Thread(this, "NThread");

        System.out.println("CT:"+t);

        t.start();
    }
}

```

```

        }
    public void run()

    {

        try
        {

            for(int n=5;n>0;n--)
            {

                System.out.println("CSE "+n);
                Thread.sleep(2000);

            }
        }
        catch(InterruptedException ie) {
            System.out.println("CSE Interrupted"); }
        System.out.println("CSE quitting");

    }

}

public class PrintColleg {

    public static void main(String ss[])
    {
        new Thread1();
        System.out.println("Back in main");
        try
        {
            for(int n=2;n>0;n--)
            {

                System.out.println("BMSCE "+n);

                Thread.sleep(10000);
            }
        }
        catch(InterruptedException ie) {

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

        }
        System.out.println("BMSCE quitting");
        System.out.println("Sanjeet P. Pandit 1BM22CS241");
    }
}

```

```
}
```

Lab 9:

(10) Demonstrate Inter process Communication and deadlock

1) Inter process Communication

```
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 {
    Queue q;
    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }
    public void run() {
        int i = 0;
        while(i<4) {
            q.put(i++);
        }
    }
}
class Consumer implements Runnable {
    Queue q;
    Consumer(Q q) {
        this.q = q;
    }
}
```

```

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

}

public void run() {
    int i=0;
    while(i<4) {
        int r=q.get();
        System.out.println("consumed:"+r);
        i++;
    }
}
}

class PCFixed {
    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("Sanjeet P. Pandit 1BM22CS241");
    }
}

```

2) Deadlock

```

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();
    System.out.println("Sanjeet P. Pandit 1BM22CS241");
}

```

Lab 10:

(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 Arithmetic Exception Display the exception in a message dialog box.

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

class SwingDemo1 {
    SwingDemo1(){
        // 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 :)
        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 SwingDemol();
        }
    });
    System.out.println("Sanjeet P. Pandit 1BM22CS241");
}
}

```