

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:

SANKETH. M. HANASI
1BM22CS242

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

Name : Sanketh M. Hanaci

Section : 3F

USN : IBM22CS242

<u>Sl.no</u>	<u>Date</u>	<u>Title</u>	<u>Marks</u>	<u>Sign</u>
1	12/12/23	Quadratic	10	✓
2	26/12/23	Student SGPA	10	✓
3	26/12/23	Books	10	✓
4	02/01/24	Abstract	10	✓
5	09/01/24	Bank	10	✓
6	23/01/24	Package	10	16/1/24 ✓
7	30/01/24	Exception Handling	10	6/2/24 ✓
8	06/02/24	Multi Threading	10	5/2/24 ✓
9	13/02/24	Lab 10 → A) ICP B) Deadlock	10	02/24 ✓
10	20/02/24	Lab 9 → AWT	10	13-2-24 ✓

① Develop a Java program that prints all real solⁿ to the quadratic eqⁿ $ax^2+bx+c=0$. Read in a,b,c and use the quadratic formula.

```
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 ($d > 0$)

{

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

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

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

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

g

else if ($d < 0$)

{

$$r_1 = \frac{(-b)}{(2^a)};$$

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

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

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

g

3

class QuadraticMain

{

public static void main (String[] args)

{

Quadratic q = new Quadratic();

q.getd();

q.compute();

System.out.println ("Sanketh M Hanasi
1Bm2cs262");

g

g

Output:

Enter the coefficients of a,b,c:

1 2 1

Roots are real and equal

$$\text{Root1} = \text{Root2} = -1.0$$

Sanketh M Hanasi IBM22CS242

Enter

3 4 2

Roots are imaginary

$$\text{Root1} = 0.0 + i 1.0554159678$$

$$\text{Root2} = 0.0 - i 1.0554159678$$

Sanketh M Hanasi IBM22CS242

Enter

1 3 2

Roots are real and distinct

$$\text{Root1} = -1.0 \quad \text{Root2} = -2.0$$

Sanketh M Hanasi IBM22CS242



QW

$\sqrt{2}$ $\sqrt{2}$

12

Lab 2:

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

$$\text{SGPA} = \frac{\sum [\text{course credits} \times \text{course points}]}{\sum [\text{course credits}]}$$

```
import java.util.*;  
class Subject {  
    int sub_marks;  
    int credits;  
    int grade;  
}  
  
class student {  
    Subject subject[7];  
    String name;  
    String usn;  
    double SGPA;  
    Scanner s;  
  
    student() {  
        int i;  
        subject = new Subject[9];  
        for (i=0; i<9; i++) {  
            subject[i] = new Subject[7];  
        }  
        s = new Scanner(System.in);  
    }
```

void getstudentdetails()

{
System.out.println("Enter name and USN");
this.name = s.nextLine();
this.USN = s.nextLine();

3

void getmarks()

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

{
System.out.println("Enter the marks of " + i + "
subject");

subject[i].sub_marks = s.nextInt();

System.out.println("Enter the credits of " + i + "
subject");

subject[i].credits = s.nextInt();

subject[i].grade = (subject[i].sub_marks / 10);

if (subject[i].grade > 10) {

subject[i].grade = 10;

3

if (subject[i].grade < 4) {

subject[i].grade = 0;

3

3
void computeschpa()

int totalcredits = 0;

int sum = 0;

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

sum = sum + subject[i].grade * subject[i].credits;

totalcredits = totalcredits + subject[i].credits;

3

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);  
    }  
}
```

Output: Enter name and usn

Sanketh M. Hanand

1BM22CS242

Enter marks of 1 subject : 85

Enter credits of 1 subject : 4

Enter marks of 2 subject : 86

Enter credits of 2 subject : 4

Enter marks of 3 subject : 90

Enter credits of 3 subject : 3

Enter marks of 4 subject : 88

Enter credits of 4 subject : 3

Enter marks of 5 subject : 79

Enter credits of 5 subject : 3

Enter marks of 6 subject : 84

Enter credits of 6 subject : 1

Enter marks of 7 subject : 99

Enter credits of 7 subject : 1

Enter marks of 8 subject : 90

Enter credits of 8 subject : 1

Name: Sanketh M Hanasi

USN: 1BM22CS2W2

SchPA : 9.1

SS
26/10/2028

26/12/23

Lab 3 i

Create a class Book which contains 4 members: name, author, price, num pages. Include a constructor to set the values for the members. Include methods to set & 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 books 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;  
}
```

```
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] = newBooks(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("Sanketh M Hanur | IBM22CS262");
}
```

Output: 10 + 5 = 15

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 Author of book 2:

J K Rowling

Enter price of book 2:

599

Enter no of pages of book 2:

600

Book 1:

Name : Da Vinci code

Author : Dan Brown

Price : 499

No of pages : 350

***** * * * * *

Book 2:

Name : Harry Potter

Author : J K Rowling

Price : 599

No of Pages : 600

***** * * * * *

~~Sanketh M Hanasi~~

~~1BM22CS21,2~~

~~SB
26/12/23~~

Lab 1:

Date: 02/01/2021

Develop a Java program to create an abstract class named Shape that contains 2 integers & an empty method named printArea(). Provide three classes named Rectangle, Triangle & Circle such that each one of them 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 rectangle 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 triangle 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 ("Sanketh.M.Hanasi" + "IBM22CS242")
```

Output:

Enter the length and Breadth of rectangle:

10

2

Area of rectangle is 20.0

Enter the height and base of triangle:

3

4

Area of triangle is 10.0

Enter the radius of circle:

10

Area of circle is 314.0

Sanketh.M.Hanasi IBM22CS242

W/24

LAB 5 :

Develop a java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other called 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 & if the balance falls below this level, 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;
```

3

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

```
void display() {  
    System.out.print("Name: " + name + " It Account  
    " + accNo + " It Account Type: " + type + " It Balan  
    + balance);  
}
```

```
class savingsAcc extends Account {
```

```
    private static double rate = 3.5;
```

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

```
void collect() {
```

```
    double interest = (balance * 1 * rate) / 100;  
    System.out.println("Interest is " + interest);  
}
```

```
class CurrentAcc extends Account {
    private double minBal = 500;
    private double s_charge = 50;
    currentAcc(string int name, accno, "current", double balance) {
        super(name, accno, "current", balance);
    }
}
```

```
void checkBal()
```

```
{
```

```
if (balance < minBal)
```

```
{
```

```
System.out.println("Insufficient Balance");
```

```
balance = balance - s_charge;
```

```
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);  
SavingsAcc sv = SavingsAcc(name, AccNo, init.bal);  
CurrentAcc ca = CurrentAcc(name, AccNo, init.bal);  
  
while(true){  
    if (type.equalsIgnoreCase("savings")){  
        System.out.println("---- MENU ----");  
        System.out.println("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");

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 amount : ");

amt = s.nextDouble();

a.withdraw(amt);

c.checkBal();

break;

case 3:

a.display();

break;

case 4:

System.exit(0);

}

}

System.out.println("Sanket M Hanasi IGM22CS207")

g

Output:

Enter the name :

Sanketh

Enter the Account Number:

123

Enter the Account type:

savings

Enter the Balance

1000

--MENU--

Enter 1: Deposit 2: Withdraw 3: Interest

4: Display 5: Exit

Enter amount

1000

--MENU--

Enter 1: Deposit 2: Withdraw 3: Interest

4: Display 5: EXIT

2

Enter amount

1000

Insufficient Balance

Sanketh M - Hanock

IRM77CS747

16/11/24

create a package CIE which has two classes - Student & 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 sem 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 course.

```
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);
```

3.

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

```
package SEF;
import CIF.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 inputSEFmarks(){
        Scanner s = new Scanner(System.in);
        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]);
    }
}
```

```
import SEF.Externals;
```

```
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 IIE marks");
            finalMarks[i].inputIIEmarks();
            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 ("Samketh M Hanesi 1BM20SCX23");
```

Output:

Enter student details:

1BM22CS242

Sanketh

3

Enter CIE marks:

Subject 1 : 40

Subject 2 : 45

Subject 3 : 41

Subject 4 : 48

Subject 5 : 44

Enter SEE marks:

Subject 1 : 80

Subject 2 : 90

Subject 3 : 86

Subject 4 : 80

Subject 5 : 84

Enter student details:

1BM22CS243

Sentash

3

Enter CIE marks:

Subject 1 : 50

Subject 2 : 45

Subject 3 : 40

Subject 4 : 48

Subject 5 : 46

Enter SEE marks:

Subject 1 : 90

Subject 2 : 96

Subject 3 : 100

Subject 4 : 98

Subject 5 : 92

Displaying Details:

USN: IBM22CS242 Name: Sanketh sem:3
108 MY. 1000000000

subject 1: 80

subject 2: 90

subject 3: 84

subject 4: 88

subject 5: 86

USN: IBM22CS243 Name: Santosh Sem:3

subject 1: 95

subject 2: 93

subject 3: 90

subject 4: 96

subject 5: 92

Sanketh M. Hanasi IBM22CS242

16-01-24

Strings

1) string constructor : BMSCE

abcd

string constructor using object : abcd

string constructor with index of chars : bcd

string constructor using ascii values of characters : ABCDEF

string constructor using ascii : BCD

2) string length of chars : 3

string length of literals : 8

string concatenation : BMS COLLEGE

3) `toString()` method : 242 sanketh

4) string using `getBytes()`:

87 101 108 99 111 109 101

string using `toCharArray()`:

W e l c o m e

5) string using `getChars()`:

B M S C E

6) BmSce equals BmSce : true

BmSce equals College : false

BmSce equals BMSCE : false

BmSce equals BMSCE : true

7) substring is matched

8) String using `startsWith()` : true

String using `startsWith()` : false

9) strings using endswith(): True
Strings using endswith(): False

10) Hello equals Hello : true
Hello == Hello : false

11) apple ball cat dog ent free gun hen ice
bug kite lift man net orange parrot queen
ring star tree umbrella van watch xmas
yacht zee.

12) 1 2 3 4 5 6 7 8 9

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

14) Hello world

15) Cammege

16) Hello friends

17) Student 1

name: Sanketh

reg no: 242

sem : 3

GPA : 9.1

Student 2

name: vayu

reg no : 249

sem : 3

GPA : 9.8

12) strings
string
string

13) Eagle i
Eagle m

Hawk
Hawk

14) circle
circle

Triangle
Triangle

Sanketh

1) strings using `charAt()` : char at 3. is 'm'
string using `getChars()` : BMSCB
string reverse() : BCSMB

Eagle is flying
Eagle makes a sound

Hawk is flying
Hawk makes a sound

$$\text{circle-area} = 3\pi$$

$$\text{circle-peri} = 62.8$$

$$\text{Tri-area} = 2.5$$

$$\text{Tri-peri} = 11$$

Niketh M Maneesha 1BM22CS262

16/01/24

Generics

1. Write a java program to create a generic class stack which hold 5 integers & 5 double value.

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

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

```
    E stck[];
```

```
    int top;
```

```
    int size=10;
```

```
Stack() {
```

```
    stck = (E[]) new Object[size];
```

```
    top=-1;
```

3

```
void push(E item) {
```

```
    if (top == size - 1)
```

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

```
    else
```

```
        stck[++top] = item;
```

g

```
E pop() {
```

```
    if (top < 0) {
```

~~System.out.println("Underflow");~~~~return null;~~

g

```
else {
```

```
    return stck[top--];
```

g

```
public class TestStacks {  
    public static void main (String [] args) {  
        Stack < Integer > myStack1 = new Stack < Integer > ();  
        Stack < Double > myStack2 = new Stack < Double > ();  
  
        Scanner s = new Scanner (System.in);  
        System.out.println ("Enter Elements into the integer stack");  
        for (int i=0; i<5; i++) {  
            int n = s.nextInt();  
            myStack1.push (n);  
        }  
  
        System.out.println ("Enter Elements into the Double stack");  
        for (int i=0; i<5; i++) {  
            double m = s.nextDouble();  
            myStack2.push (m);  
        }  
  
        System.out.println ("Elements of myStack1 ");  
        for (int i=0; i<5; i++) {  
            System.out.println (myStack1.pop());  
        }  
  
        System.out.println ("Elements of myStack2 ");  
        for (int i=0; i<5; i++) {  
            System.out.println (myStack2.pop());  
        }  
  
        s.close();  
    }  
}
```

Output 1

Enter Elements into the integer stack

1 2 3 4 5

Enter Elements into the Doublet stack

6 7 8 9 10

Elements of mystack1

1 2 3 4 5

Elements of mystack2

6.0 7.0 8.0 9.0 10.0

Sanketh M Hanasi

1BM22CS262

60 24

Exception Handling:

WAP that demonstrates handling of exception in inheritance tree.

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

```
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) {
```

~~throws new WrongAge ("Age can't be negative")~~

```
}
```

```
void Fdisplay () {
```

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

class Son extends Father {

int sonAge;

public Son() throws WrongAge {

InputScanner sc = new InputScanner();

sonAge = sc.sc.nextInt();

if (sonAge >= fatherAge) {

throw new WrongAge("Son's age cannot be
greater or equal to 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.Cdisplay();

catch (WrongAge e) {

System.out.println(e);

System.out.println("Son's age must be positive");

Output :

30

8

Father's Age : 30

Son's Age : 8

10

12

Wrong Age : Son's age can't be greater than or equal to
Father's age

Q-1

wrong age : Age cannot be negative

Sanketh M. Hanasi

IBN22CS11,2

60/1-24

30

Lab Program 8:

06/02/24

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.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 PrintCollege {
    public static void main(String args[]) {
        new Thread();
        System.out.println("Back in main");
        try {
            for (int n=2; n>0; n--) {
                System.out.println("BMSCE" + n);
                Thread.sleep(1000);
            }
        } catch (InterruptedException ie) {
            System.out.println("BMSCE interrupted");
        }
        System.out.println("BMSCE quitting");
    }
}

```

Output:

Back in main

CSE 5

BMSCE 2

CSE 4

CSE 3

CSE 2

CSE 1

BMSCE 1

CSE. quitting

BMSCE quitting

Lab 9 - 20-02-2024

Write a program that creates a user interface to perform integer divisions.

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

```
class SwingDemo
```

```
SwingDemo {
```

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

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

```
    JTextField qtf = new JTextField(8);  
    JTextField htf = new JTextField(8);
```

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

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

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

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

```
    JLabel anslabel = new JLabel();
```

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

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

```
jfm.add(ajtf);  
jfm.add(bjtf);  
jfm.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("A/B = " + a);  
            blab.setText("B = " + b);  
            anslab.setText("ANS = " + ans);  
            err.setText("A should be NON zero!");  
  
        }  
        catch(NumberFormatException e){  
            alab.setText("");  
            blab.setText("");  
            anslab.setText("");  
            err.setText("B should be NON zero!");  
        }  
    }  
});
```

```
        catch(ArithmeticException e) {  
            alab.setText("");  
            blab.setText("");  
            anslab.setText("");  
            err.setText("B should be non zero");  
        }  
    }  
}
```

```
3) j  
jfrm.setVisible(true);
```

3

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

```
System.out.println("Sanketh M Hanasi IITM22CS242")
```

4

Output:

Enter the divider and dividend:

Calculate

A=10 B=2 Ans=5

Sanketh M Hanasi IITM22CS242

`setSize(w,h)` : Method that resizes the component so that it has width w, height h.

`setLayout()` : Method allows you to set the layout of the container of ten a JPanel to say FlowLayout, BorderLayout, etc.

`addActionListener()`: Event handlers to implement to define what should be done when an user performs certain operation.

`addWindowListener()`: Overriding only the methods of interest.

`setDefaultCloseOperation`: sets the operation that will happen by default when the user initiates a "close" on this frame.

`JFrame` : JFrame in Java is a class that allows you to create and manage a top level window in a Java application. It serves as the main window for GUI-based Java applications and provides a platform independent way to create graphical user interfaces.

`JLabel` : JLabel is a class of java swing. It is used to display a short string or even image icons. It can display text, image, or both. It is insensitive to input events such as mouse focus or keyboard focus.

`invokeLater()` : It can be used to perform a task asynchronously in AWT Event dispatcher thread.

* `setVisible()` : It is a method that has return type boolean. It sets the currently made attributes to be visible on the screen GUI application mainly Java Swing.

* `swingUtilities` : The swingUtilities class has 2 important static methods, `invoke` and `wait()` and `invokeLater()` to add put references to blocks of code into event queue.

UNO
20-2-19

Lab 10 : 1
12/02/14

A) Demonstrate inter process communication.

class A {

int n;

boolean valueSet = false;

synchronized int get() {

while (!valueSet)

try {

System.out.println("In consumer waiting");

wait();

}

catch (InterruptedException e) {

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

}

System.out.println("Not :" + n);

valueSet = false;

System.out.println("In Interlock Producer");

notify();

return n;

;

synchronized void put(int n) {

while (!valueSet)

try {

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

wait();

}

catch (InterruptedException e) {

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

}

```
this.n = n;  
valueSet = true;  
System.out.print("Put: " + n);  
System.out.println("In Intimate consumer\n");  
notify();
```

4

3
class Producer implements Runnable {

Q2:

Producer(Q2){

this.g = g;

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

}

public void run(){

int i = 0;

while (i < 3) {

g.put(i++);

}

3

3
class Consumer implements Runnable {

Q2:

Consumer(Q2){

this.g = g;

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

4
public void run(){

int i = 0;

while (i < 3) {

int r = g.get();

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

i++;

3

```
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.'");
    }
}
```

Input:

Put: 0

Intimate. Consumer

Producer waiting

Press control-c to stop.

Got: 0

Intimate. Producer

consumed: 0

Put: 1

Intimate. Consumer

Producer waiting

Got: 1

Intimate. Producer

consumed: 1

Put: 2

Intimate. Consumer

Producer waiting

Got: 2

Intimate. Producer

consumed: 2

B) Demonstrate 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.blast();");

b.last();

}

void last() {

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

}

3
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();

3

```
void last() {
```

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

class Deadlock implements Runnable {

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

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

```
Deadlock() {
```

```
    Thread.currentThread().setName("Main Thread");
```

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

```
    t.start();
```

```
    a.foo(b);
```

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

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

```
    b.bar(a);
```

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

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

```
    new Deadlock();
```

MainThread entered A.foo

RacingThread entered B.bar

RacingThread trying to call A.last()

Inside A.last

Back in other thread

MainThread trying to call B.last()

Inside B.last

Back in main thread

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

class QuadraticMain
{
    public static void main(String[] args)
    {
        Quadratic q=new Quadratic();
        q.getd();
        q.compute();
        System.out.println("Sanketh M Hanasi 1BM22CS242");
    }
}

```

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("Sanketh M Hanasi 1BM22CS242");
}
}

```

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("Sanketh M Hanasi \n 1BM22CS242");

    }
}

```

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 ofthe 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("Sanketh.M.Hanasi 1BM22CS242");

    }
}

```

5)Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

- Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:
- a)Accept deposit from customer and update the balance.
- b)Display the balance.
- c)Compute and deposit interest
- d)Permit withdrawal and update the balance
- Check for the minimum balance, impose penalty if necessary and update the balance.

```

import java.util.Scanner;

class Account{
    String name;
    int 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 callInt(){
        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.callInt();
            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("Sanketh M Hanasi      1BM22CS242");
}

}
}

```

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("Sanketh.M.Hanasi 1BM22CS242");
    }
}

```

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("Sanketh.M.Hanasi 1BM22CS242");
    }
}

```

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("Sanketh.M.Hanasi 1BM22CS242");  
    }  
}
```

10) Demonstrate Inter process Communication and deadlock

a) 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 {

    Q 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 {

    Q 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("Sanketh.M.Hanasi 1BM22CS242");
    }
}

```

b)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 thisthread.

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

    }

    public void run() {

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

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

    }

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

```

        new Deadlock();
        System.out.println("Sanketh.M.Hanasi 1BM22CS242");
    }
}

```

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 SwingDemo1();
        }
    });
    System.out.println("Sanketh M Hanasi      1BM22CS242");
}
}
```