

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



OBJECT ORIENTED JAVA PROGRAMMING

LAB RECORD

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

SANTOSH HANAMAPPA JAMBAGI
USN-1BM22CS244

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

NAME: SANTOSH H JAMBAGI

SECTION : 3 E

USN : 1BM22CS244

Sub : OOT

Lab Record

I N D E X

//_

S.I.No	Date	Title	Page No	Teacher's Sign
1.	12/12/23	Quadratic Equation	- 10	1-3
2.	19/12/23	SGPA calculator	- 10	4-7
3.	26/12/23	Book Details	- 10	8-10
4.	2/1/24	Abstract class	- 10	11-13
5.	9/1/24	Bank Details	- 10	14-18
6.	16/1/24	packages	- 10	19-22
7.	30/1/24	Exception Handling	- 10	23-25
8.	6/2/24	Threads	- 10	26-27
10 a>	13/2/24	Inter Process Communication	28-31	
10 b>	13/2/24	Dead lock	32-34	
9.	20/2/24	Division Main	- 10	35-37

> Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read 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 ( $d > 0$ )

{

$r1 = ((-b) + (\text{Math.sqrt}(d))) / (\text{double})(2*a);$

$r2 = ((-b) - (\text{Math.sqrt}(d))) / (\text{double})(2*a);$

`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'");`

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

$r2 = \text{Math.sqrt}(-d) / (2*a);$

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

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

}

}

}

public static void main(String args[])

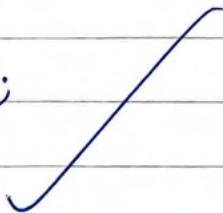
{

`Quadratic q = new Quadratic();`

`q.getd();`

`q.compute();`

}



}

Output:

Enter the coefficients of a,b,c:

1 2 1

Roots are real and equal

Root 1 = Root 2 = -1.0

Done by: Santosh Jambagi

USN: IBM22CS244

Enter the coefficients of a,b,c:

3 4 5

Roots are Imaginary

Root 1 = 0.0 + i1.10554159678

Root 2 = 0.0 - i1.10554159678

Done by: Santosh Jambagi

USN: IBM22CS244

Enter the coefficients of a,b,c:

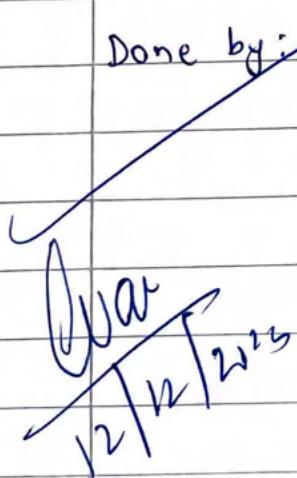
1 3 2

Roots are real and Distinct

Root 1 = -1.0 Root 2 = -2.0

Done by: Santosh Jambagi

USN: IBM22CS244



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

$$\rightarrow \text{SGPA} = \frac{\sum (\text{course credits})(\text{grade points})}{\sum (\text{course credits})}$$

```

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 the name:");
 name = s.nextLine();
 System.out.println("Enter the USN:");
 usn = s.nextLine();
}
```

```
void getMarks(){
 int i;
 for(i=0; i<9; i++)
 {
 System.out.println("Enter the marks & credits for course
 "+(i+1)+":");
 System.out.println("Marks:");
 int marks = s.nextInt();
 System.out.println("Credits:");
 int credits = s.nextInt();
 subject[i].subjectMarks = marks;
 subject[i].credits = credits;
 subject[i].grade = (subject[i].subjectMarks)/(10) + 1;
 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<9; i++)
 {
 total_grade_and_credit += subject[i].credits * subject[i].grade;
 }
 double SGPA = (float) total_grade_and_credit / 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();
 }
}

```

### Output:

Enter the name: Santosh Jambagi

Enter the USN: 1BM22CS244

Enter the marks and credits for course 1:

Marks: 88

credits: 4

Enter the marks and credits for course2:

Marks: 89

credits: 4

course 3:

Marks: 89

credits: 3

course 4:

Marks: 82

credits: 1

course 5:

Marks: 91

credits: 1

course 6:

Marks: 90

credits: 3

course 7:

Marks: 81

credits: 3

course 8:

Marks: 97

credits: 1

Your SGPA is: 9.25

3) Create a class Book which contains four members. Include a constructor. Develop a Java program to create n book objects.

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

```
class Books {
```

```
 String name, author;
```

```
 int price, numPages;
```

```
 public Books (String name, String author, int price,
 int numPages) {
```

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

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

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

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

```
}
```

```
public String toString()
```

```
{
```

```
 String name, author, price, numPages;
```

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

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

```
 price = "Price :" + this.price + "\n";
```

```
 numPages = "Number of Pages :" + this.numPages + "\n";
```

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

```
}
```

```
}
```

```
class Main{
```

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

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

```
 int price, numPages, n;
```

```
 String name, author;
```

```
 int i;
```

```
 System.out.println("Enter the no of Book:");
```

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

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

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

```
{
```

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

```
 name = s.nextLine();
```

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

```
 author = s.nextLine();
```

```
 System.out.println("Enter the price of Book:");
```

```
 price = s.nextInt();
```

```
 System.out.println("Enter the no of pages:");
```

```
 numPages = s.nextInt();
```

```
 b[i] = new Books(name, author, price, numPages);
```

```
}
```

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

```
{
```

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

```
}
```

```
}
```

Output:

Enter the number of Books: 3

Enter the name of Book1: Physics

Enter the name of author: H.C.Verma

Enter the price: 599

Enter the no of pages: 400

Enter the name of Book2: Chemistry

Enter the name of author: John

Enter the price: 499

Enter the no of pages: 399

26/12/22

Enter the name of Book3: Maths

Enter the name of author: R.D.Sharma

Enter the price: 500

Enter the no of Pages: 300

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 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 the length and breadth of the rectangle:");
 a = sc.nextDouble();
 b = sc.nextDouble();
 }

 void displayarea()
 {
 System.out.println("The area of rectangle is "+(a*b));
 }
}

```

class triangle extends shaped

void getinput()

{

System.out.println("Enter the base & height of  
a triangle:");

a = sc.nextDouble();

b = sc.nextDouble();

}

void displayarea()

{

System.out.println("The area of triangle is:" +  
(0.5 \* a \* b));

}

}

void getinput()

{

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

a = sc.nextDouble();

}

void displayarea()

{

System.out.println("The area of circle is:" +  
(3.14 \* a \* a));

}

}

class Area

{

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 the rectangle:

4 7

The area of rectangle is: 28

Enter the base and height of the triangle:

3 6

The area of triangle is: 9.0

Enter the radius of circle:

4

The area of circle is: 50.24

~~Q&A  
2-1-2014~~

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");
 }

 void display()
 {
 System.out.println("Name:" + name, "Account No:" + accno,
 "type:" + type, "Balance:" + balance);
 }
}

```

```
class SavAcc extends account {
 private static double rate = 5;
 SavAcc(String name, int accno, double balance)
 {
```

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

```
 void interest()
```

```
{
```

```
 balance += balance * (rate) / 100;
```

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

```
}
```

```
class CurAcc extends account {
```

```
 private double minBal = 500;
```

```
 CurAcc(String name, int accno, double balance)
```

```
{
```

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

```
 void checkmin()
```

```
{
```

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

```
 System.out.println("Balance is less than min Balance");
```

```
 "Service charges imposed :" + servicecharges;
```

```
 System.out.println("Balance is " + balance);
```

```
}
```

```
}
```

```
class Bank{
 public static void main(String args[])
 {
 Scanner s = new Scanner(System.in);
 System.out.println("Enter the name:");
 String name = s.next();
 System.out.println("Enter the type of account:");
 String type = s.next();
 System.out.println("Enter the account number:");
 int accno = s.nextInt();
 System.out.println("Enter the initial balance:");
 double balance = s.nextDouble();
 int ch;
 double amount1, amount2;
 account acc = new account(name, accno, type, balance);
 savAcc sa = new account(name, accno, balance);
 curAcc ca = new account(name, accno, balance);
 while(true)
 {
 if(acc.type.equals("savings"))
 {
 System.out.println("In Menu In 1. Deposit
 In 2. withdraw In 3. ComputeInterest In
 4. display");
 System.out.println("Enter the choice:");
 ch = s.nextInt();
 switch(ch)
 {
 case 1: System.out.println("Enter the amount:");
 amount1 = s.nextInt();
 sa.deposit(amount1);
 break;
 }
 }
 }
 }
}
```

```

 case 2: System.out.println("Enter the amount :");
 amount 2 = s.nextInt();
 sa.withdraw(amount2);
 break;
 case 3: sa.Interest();
 break;
 case 4: sa.display();
 break;
 case 5: exit(0);
 default: System.out.println("Invalid input");
 break;
}
else
{

```

System.out.println("In Menu In 1.Deposit In 2.Withdraw  
In 3.Display In 4.Exit);

System.out.println("Enter the choice:");
ch = s.nextInt();

switch(ch)

{

case 1: System.out.println("Enter the amount :");
amount 1 = s.nextInt();
ca.deposit(amount1);
break;

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

amount 2 = s.nextInt();

ca.withdraw(amount2);

ca.checkmin();

break;

case 3: ca.display();

break;

case 4 : System.exit(0);

}

}

}

Output:

Enter the name: Santosh

Enter the type (current/savings):

current

Enter the account number:

5425

Enter the initial balance:

1000

----- MENU -----

1. Deposit    2. Withdraw    3. display

Enter the choice: 1

Enter the amount: 1000

----- MENU -----

1. Deposit    2. Withdraw    3. Display

4.

Enter the amount: 500

----- MENU -----

1. Deposit    2. Withdraw    3. Display

2.

Enter the amount = 500

----- MENU -----

1. Deposit    2. Withdraw    3. Display

3.

Name: santosh    Accno: 5425    Type: current    Balance: 1500

QW  
16/11/2023

## Strings:

### String constructors:

```
char chars[] = {'b', 'm', 's', 'c', 'e'}
```

```
String s1 = new String(chars)
```

```
String s2 = new String(chars, 1, 3);
```

Output: bmsce

msc.

### String length:

```
char chars[] = {'p', 'y', 't', 'h', 'o', 'n'}
```

```
String s = new String(chars);
```

```
System.out.println(s.length());
```

Output: 5

### String literal & concatenation:

```
System.out.println("abc".length());
```

```
String car = "BMW";
```

```
System.out.println("He has " + car + " car");
```

Output: 2

He has a BMW car.

### getChars:

~~String clg = "Welcome to Bmsce College";~~~~getChars(11, 17, buff, 0);~~

Output: Bngce

### equals & equalsIgnoreCase:

Bmsce equals Bmsce → true

Bmsce equals College → false

Bmsce equals BMSC → false

Bmsce equalsIgnoreCase BMSC → true

## regionMatches:

Boolean isMatch = str1.regionMatches(11, str2, 0, 13);

O/p: substring is matched

## starts with and ends with:

String game = "Basketball"

System.out.println("Basket game.startsWith(\"Basket\"));

System.out.println(game.endsWith("ball"));

O/p:

true

true

## equals v/s ==

Hello equals Hello  $\rightarrow$  true

Hello == Hello  $\rightarrow$  false

## sort

apple ball cat dog end free gun hen ice jug kite lift man  
net orange parrot queen ring star tree umbrella van watch xmas  
yatch zee.

11 12

2 3 4 5 6 7 8 9

13

This is a test. This is, too

14

World

15

Commege

16

Hello friends

17

Student 1

name: Santosh

Reg no: 244

Sem: 3

(CGPA: 8.35)

Student 2

name: Sanket

Reg no: 242

Sem: 3

(CGPA: 9.2)

Write a java program to create a generic class stack which holds 5 integers & 5 double values.

```

import java.util.*;
class Stack<E>{
 E stk[];
 int top;
 int size=10;
 Stack(){
 stk = (E[]) new Object[size];
 top = -1;
 }
 void push(E item){
 if (top == size - 1)
 System.out.println("Overflow");
 else
 stk[++top] = item;
 }
 E pop(){
 if (top < 0)
 System.out.println("Underflow");
 return null;
 }
}

```

```
else {
 return stk[top--];
}

}

public class Teststack {
 public static void main(String[] args) {
 Stack<Integer> mystack1 = new Stack<Integer>();
 Stack<Integer> 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();
 }
}
```

WU 1/24  
WU

→ 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 name = new String();
 protected String usn = new String();
 protected int sem;
 public void inputStudentDetails() {
 Scanner s = new Scanner(System.in);
 System.out.println("Enter the name:");
 name = s.nextLine();
 System.out.println("Enter the usn:");
 usn = s.nextLine();
 System.out.println("Enter the sem:");
 sem = s.nextInt();
 }
 public void displayStudentDetails() {
 System.out.println("Name:" + name);
 System.out.println("USN:" + usn);
 System.out.println("Sem:" + sem);
 }
}

```

## //Internals

```
package CIE;
import java.util.*;
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();
 }
 }
}
```

## //Externals

```
package SEE;
import CIE.internals;
import java.util.*;
public class Externals extends Internals{
 protected int marks[];
 protected int finalmarks[];
 public Externals(){
 marks = new int[5];
 finalmarks = new int[5];
 }
 public void calculateFinalMarks(){
 Scanner s = new Scanner(System.in);
 for(int i=0; i<5; i++){
 System.out.println("Enter 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) + " Final Marks : " + finalmarks[i]);
 }
}
```

// Main

```
import SEE.Externals;
class Main {
 public static void main(String args[]) {
 int n = 2;
 Externals finalmarks[] = new Externals[n];
 for (int i = 0; i < 5; 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 : ");
for (int i = 0; i < 5; i++) {
 finalmarks[i].calculateFinalMarks();
 finalmarks[i].displayFinalMarks();
}
```

Output:

Enter the USN: IBM22CS244

Enter the name: Santosh H Tambagi

Enter Sem : 3

Enter CIE marks:

Subject 1 : 44

Subject 2 : 47

Subject 3 : 42

Subject 4 : 43

Subject 5:44

ENTER SEE

Subject : 88

Subject 3: 87

Subject 4 : 88

Display Data:

USN : IBM 22CS244

Name : Santosh H Tambay;

Sem : 3

Subject 1: 86

Subject 2 : 90

Subject 3:85

Subject 4 : 87

Subject 5: 87

A hand-drawn graph on lined paper. It features two intersecting curves, both drawn in red ink. The upper curve is labeled with the letter 'W' in red ink, positioned near its peak. The lower curve is also labeled with the letter 'M' in red ink, positioned near its lowest point. The graph is drawn on a background of horizontal blue lines.

→ Write a program that demonstrates handling of exceptions in inheritance tree.

```

import java.util.*;
class WrongAge extends Exception{
 public WrongAge(String message){
 super(message);
 }
}

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

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

```

```
class Son extends Father{
 int sonAge;
 public Son() throws WrongAge{
 System.out.println("Enter Son's age :");
 sonAge = sc.nextInt();
 if (sonAge > fatherAge)
 throw new WrongAge("Son's age cannot be greater");
 else if (sonAge < 0)
 throw new WrongAge("Age cannot be negative");
 }
 void sdisplay(){
 System.out.println("Son's age is :" + sonAge);
 }
}
public class ExceptionHandling{
 public static void main(String args[]){
 Son p;
 try {
 p = new Son();
 p.fdisplay();
 p.sdisplay();
 } catch (Exception e) {
 System.out.println(e);
 }
 }
}
```

11

Output:

Enter the father's age:

34

Enter the son's age:

54

Wrong Age: Son's age cannot be greater than father's age.

Enter the father's age:

34

Enter the son's age:

-24

Wrong Age: Age cannot be negative.

Enter the father's age:

34

Enter the son's age:

34

Wrong Age: Age cannot be equal.

WR  
501-2y

→ Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```

import java.util.*;
class Thread1 extends Thread
{
 public void run()
 {
 for(int i=0; i<5; i++)
 {
 try
 {
 System.out.println("BMS College of Engineering");
 Thread.sleep(10000);
 }
 catch(InterruptedException e)
 {
 System.out.println("Interrupted");
 }
 }
 }
}

class Thread2 extends Thread
{
 public void run()
 {
 for(int j=0; j<5; j++)
 {
 for(int i=0; i<5; i++)
 {
 try
 {
 System.out.println("CSE");
 Thread.sleep(2000);
 }
 catch(InterruptedException e)
 {
 System.out.println("Interrupted");
 }
 }
 }
 }
}

```

```
public class Threads
{
```

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

```
 Thread1 t1 = new Thread1();
 t1.start();
```

```
 Thread2 t2 = new Thread2();
 t2.start();
```

}

}

Output:

CSE

CSE

CSE

CSE

CSE

BMS College Of Engineering.

WU/24  
6/2

10) Demonstrate Inter process Communication and deadlock.

class A{

int n;

boolean valueSet = false;

synchronized int get(){

while(!valueSet){

try{

System.out.println("In Consumer waiting\n");

wait();

}

catch(InterruptedException e){

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

}

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

valueSet = true;

System.out.println("In Intimate Producer\n");

notify();

return n;

}

synchronized void put(int n){

while(valueSet)

try{

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

wait();

}

catch(InterruptedException e){

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

}

this.n = n;

valueSet = true;

```
System.out.println("Put:" + n);
System.out.println("In Intimate Consumer\n");
notify();
}
```

```
}
```

```
class Producer implements Runnable
```

```
{ q;
```

```
Producer(q)
```

```
this.q = q;
```

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

```
}
```

```
public void run()
```

```
int i=0;
```

```
while(i<15){
```

```
q.put(i++);
```

```
}
```

```
}
```

```
}
```

```
class Consumer implements Runnable
```

```
{ q;
```

```
Consumer(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:

Put: 0

Intimate Consumer

Producer waiting

Press Control-C to stop

Get: 0

Intimate Producer

consumed: 0

Put: 1

Intimate Producer

Producer waiting

Get: 1

Intimate Producer

consumed: 1

Put: 2

Intimate Consumer

Producer waiting

Get: 2

Intimate Producer

consumed: 2

Put: 3

WW  
11-2-24

## 10) 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("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");
```

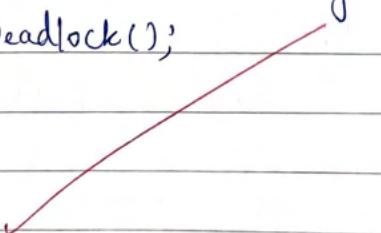
}

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

```
 new Deadlock();
```

}

}



Output:

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 A.last

Back in main thread.

WW  
13-2-24

Q) WAP that creates a UI to perform integer divisions.

```

import java.awt.*;
import java.awt.event.*;

public class DivisionMain1 extends Frame implements ActionListener {
 TextField num1, num2;
 Button dResult;
 Label outResult;
 String out = "";
 double resultNum;
 int flag = 0;

 public DivisionMain1() {
 setLayout(new FlowLayout());
 dResult = new Button("RESULT");
 Label number1 = new Label("Number 1:", Label.RIGHT);
 Label number2 = new Label("Number 2:", Label.RIGHT);
 num1 = new TextField(5);
 num2 = new TextField(5);
 outResult = new Label("Result:", Label.RIGHT);

 add(number1);
 add(num1);
 add(number2);
 add(num2);
 add(dResult);
 add(outResult);
 }

 public void actionPerformed(ActionEvent e) {
 if (flag == 0) {
 resultNum = Double.parseDouble(num1.getText());
 resultNum /= Double.parseDouble(num2.getText());
 out = String.valueOf(resultNum);
 outResult.setText(out);
 } else if (flag == 1) {
 resultNum = Double.parseDouble(num1.getText());
 resultNum *= Double.parseDouble(num2.getText());
 out = String.valueOf(resultNum);
 outResult.setText(out);
 }
 }
}

```

num1.addActionListener(this);  
num2.addActionListener(this);  
dResult.addActionListener(this);  
addWindowListener(new WindowAdapter() {  
 public void windowClosing(WindowEvent we)  
 {  
 System.exit(0);  
 }  
});

}  
public void actionPerformed(ActionEvent ae)  
{  
 int n1, n2;  
 try  
 {  
 if (ae.getSource() == dResult)  
 {  
 n1 = Integer.parseInt(num1.getText());  
 n2 = Integer.parseInt(num2.getText());  
 }  
 }  
}

/\* if (n2 == 0)  
throw new ArithmeticException();\*/  
out = n1 + "" + n2;  
resultNum = n1 / n2;  
out += String.valueOf(resultNum);  
repaint();  
}

}  
catch (NumberFormatException e1)  
{  
 out = "Number Format Exception! "+e1;  
 repaint();  
}

+ lay = 1;  
out = "Number Format Exception! "+e1;  
repaint();

```
 catch(ArithmeticException e2)
```

```
{
```

```
 flag = 1;
```

```
 out = "Divide by 0 Exception! " + e2;
```

```
 repaint();
```

```
}
```

```
 }
```

```
}
```

```
 if(flag == 0)
```

```
 g.drawString(out, outResult.getX() + outResult.
```

```
 getWidth(), outResult.getY() + outResult.getHeight() - 8);
```

```
 else
```

```
 g.drawString(out, 100, 200);
```

```
 flag = 0;
```

```
}
```

```
 }
```

```
DivisionMain1 dm = new DivisionMain1();
```

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

```
dm.setTitle("Division of Integers");
```

```
dm.setVisible(true);
```

```
}
```

Output:

Number1:

Number2:

RESULT Result 10 5 2.0

## Functions :

- 1) setLayout(..): method allows you to set the Layout of the container, often a JPanel, to say Flow Layout, BorderLayout etc
- 2) repaint(): It's an asynchronous method of applet class
- 3) drawString(): takes as parameters an instance of the String class containing the text to be drawn, and two integer values specifying the coordinates where the text should start.
- 4) addActionListener(): event handlers to implement. / to define what should be done when an user performs certain operation.
- 5) addWindowListener(): overriding only the methods of interest.

~~Very  
Very~~

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("Santosh H Jambagi USN-1BM22CS244");

 }

}

```

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("Santosh H Jambagi USN-1BM22CS244");

 }

}

```

3)Create a class Book which contains four members: name,author,price,num\_pages. Include a constructor to set 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("Santosh H Jambagi USN-1BM22CS244");
}
}

```

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("Santosh H Jambagi USN-1BM22CS244");

 }

}

```

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+"\tAccountNo:"+accNo+"\tAccountType:"+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:Withdraw3: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 withdrawingamount");
 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("Santosh H Jambagi USN-1BM22CS244");

}
}

}

```

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("Santosh H Jambagi USN-1BM22CS244");
 }
}

```

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("Santosh H Jambagi USN-1BM22CS244");
 }
}
```

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("Santosh H Jambagi USN-1BM22CS244");
 }
}

```

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 textfield");
}

};

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(ArithmeticException 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("Santosh Jambagi1BM22CS244");
}
}

```

10 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("Santosh H Jambagi USN-1BM22CS244");
}
}
```