

B.M.S. COLLEGE OF ENGINEERING, BENGALURU



LAB REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for Lab

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

**PANNAGA R BHAT
(1BM22CS189)**

Department of Computer Science and Engineering

**B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
2023-2024.**

Lab Program 1

classmate

Date _____

Page _____

* Lab Program I. Quadratic Equation

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;
import java.*;

class Quadratic
{
    public static void main(String ar[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Pannaga R. Bhat");
        System.out.println("IBM22CS189");
        System.out.println("Enter values of a,b,c");
        double a, b, c;
        a = sc.nextInt();
        b = sc.nextDouble();
        c = sc.nextDouble();
        int r1, r2;
        if (a == 0) {
            System.out.println("Roots cannot be formed");
            System.exit(0);
        }
    }
}

```

```

else
{
    int 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))) / (2*a);
        r2 = (-b - (Math.sqrt(d))) / (2*a);
        System.out.println("Roots are real and
                           distinct");
        System.out.println("Root1 = " + r1);
        System.out.println("Root2 = " + r2);
    }
    else
        System.out.println("Roots are imaginary");
}

```

Output:-

a) Pannaga R. Bhat

1BM22CS189

Enter values of a,b,c

1

2

1

Roots are real and equal

Root1 = Root2 = -1.0

b) Pannaga R. Bhat

1BM22CS189

Enter values of a,b,c

1

-3

2

Roots are real and distinct

Root 1 = 2.0

Root 2 = 1.0

c) Pannaga R. Bhat

1BM22CS189

Enter values of a,b,c

1

1

2

Roots are imaginary.

d) Pannaga R. Bhat

IBM22CS189

Enter values of a,b,c

0

4

2

Roots cannot be formed

8/11/2022

Source Code

```
import java.util.Scanner;

class Quadratic

{
    public static void main(String ar[])
    {
        Scanner sc = new Scanner(System.in);

        System.out.println("Pannaga R.Bhat");
        System.out.println("1BM22CS189");
        System.out.println("Enter values of a,b,c");

        double a,b,c;

        a = sc.nextDouble();
        b = sc.nextDouble();
        c = sc.nextDouble();

        double r1, r2;

        if (a==0)
        {
            System.out.println("Roots cannot be formed");
            System.exit(0);
        }
        else
        {
            double 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))) / (2*a);
    r2 = (-b - (Math.sqrt(d))) / (2*a);

    System.out.println("Roots are real and distinct");
    System.out.println("Root1="+r1);
    System.out.println("Root2="+r2);
}

else
    System.out.println("Roots are imaginary");
}
}
}
```

Lab Program 2

19/12/2023

* Lab 2. SGPA calculation.

```
import java.util.Scanner;  
class SUBJECT {  
    int subjectmarks;  
    int credits;  
    int grade;  
}  
  
class SGPA {  
    // Array of objects  
    SUBJECT subject[];  
    String usn, name;  
    double sgpa;  
    Scanner s = new Scanner(System.in);
```

```
SGPA() {  
    subject = new SUBJECT[8];  
    for(int i=0; i<8; i++) {  
        subject[i] = new SUBJECT();  
    }  
}
```

```
void getstudentdetails() {  
    system.out.println("Enter the name  
    of the student");  
    name = s.nextLine();  
    system.out.println("Enter the usn of  
    the student");  
    usn = s.nextLine();  
}
```

```
void getmarks() {  
    for(int i=0; i<8; i++) {  
        system.out.println("Enter Subject " + (i+1)  
                           + " marks");  
        subject[i].subjectmarks = s.nextInt();  
        system.out.println("Enter credits of subject  
                           " + (i+1));  
        subject[i].credits = s.nextInt();  
    }  
}
```

```
for(int i=0; i<8; i++) {  
    if(subject[i].subjectmarks >= 90 && subject[i].  
        subjectmarks <= 100)
```

```
subject[i].grade = 10;  
else if (subject[i].subjectmarks >= 80 &&  
        subject[i].subjectmarks < 90)  
    subject[i].grade = 9;  
else if (subject[i].subjectmarks >= 70 &&  
        subject[i].subjectmarks < 80)  
    subject[i].grade = 8;  
else if (subject[i].subjectmarks >= 60 &&  
        subject[i].subjectmarks < 70)  
    subject[i].grade = 7;  
  
else if (subject[i].subjectmarks >= 50 &&  
        subject[i].subjectmarks < 60)  
    subject[i].grade = 6  
  
else if (subject[i].subjectmarks >= 40 &&  
        subject[i].subjectmarks < 50)  
    subject[i].grade = 5;  
  
else if (subject[i].subjectmarks >= 0 &&  
        subject[i].subjectmarks < 40)  
    subject[i].grade = 0;
```

```
else {
```

```
    System.out.println("Enter the valid  
    number.");
```

```
    break;
```

```
}
```

```
double computeSGPA()
{
    double sum = 0.0, num;
    for(int i = 0; i < 2; i++) {
        num = subject[i].grade * subject[i].credits;
        sum += num;
    }
    sgpa = sum / 20;
    return sgpa;
}
```

```
void display(double sgpa) {
    System.out.println("Name of the
candidate: " + name);
    System.out.println("USN of the
candidate: " + usn);
    System.out.println("SGPA = " + sgpa);
}
```

```
class A {
```

```
public static void main(String ar[])
{
    SGPA ob = new SGPA();
}
```

```
    ob.getStudentDetails();
    ob.getMarks();
    double c = ob.computeSGPA();
    ob.display(c);
}
```

output is

Enter the name of the student
PANNAGA R BHAT

Enter the usn of the student

IBM22CS189

Enter subject 1 marks

100

Enter credits of subject 1

4

~~Enter~~

Enter subject 2 marks

100

Enter credits of subject 2

4

Enter subject 3 marks

100

Enter credits of subject 3

3

Enter subject 4 marks

89

Enter credits of subject 4

3

Enter subject 5 marks

99

Enter credits of subject 5

3

Enter subject 6 marks

100

Enter credits of subject 6

Enter subject 7 marks

88

Enter credits of subject 7

1

Enter subject 8 marks

100

Enter credits of subject 8

1

Name of the candidate : PANNAGA RBHAT

USN of the candidate : 1BM22CS189

SGPA = 9.8

* 19/12/23

Source Code

```
import java.util.Scanner;  
class SUBJECT{  
    int subjectmarks;  
    int credits;  
    int grade;  
}  
class SGPA{  
    // Array of objects  
    SUBJECT subject[];
```

```

String name, usn;
double sgpa;
Scanner s = new Scanner(System.in);
SGPA(){
    subject = new SUBJECT[8];
    for(int i=0;i<8;i++)
        subject[i] = new SUBJECT();
}
void getStudentDetails(){
    System.out.println("Enter the name of the student");
    name = s.nextLine();
    System.out.println("Enter the usn of the student");
    usn = s.next();
}
void getMarks(){
    for(int i=0;i<8;i++){
        System.out.println("Enter subject "+(i+1)+" marks");
        subject[i].subjectmarks = s.nextInt();

        System.out.print("Enter credits of subject "+(i+1));
        subject[i].credits = s.nextInt();
    }
    for(int i=0;i<8;i++){
        if(subject[i].subjectmarks>=90 && subject[i].subjectmarks<=100)
            subject[i].grade = 10;
        else if(subject[i].subjectmarks>=80 && subject[i].subjectmarks<90)
            subject[i].grade = 9;
        else if(subject[i].subjectmarks>=70 && subject[i].subjectmarks<80)

```

```

        subject[i].grade = 8;

    else if(subject[i].subjectmarks>=60 && subject[i].subjectmarks<70)

        subject[i].grade = 7;

    else if(subject[i].subjectmarks>=50 && subject[i].subjectmarks<60)

        subject[i].grade = 6;

    else if(subject[i].subjectmarks>=40 && subject[i].subjectmarks<50)

        subject[i].grade = 5;

    else if(subject[i].subjectmarks>=0 && subject[i].subjectmarks<40)

        subject[i].grade = 0;

    else{

        System.out.println("Enter the valid marks");

        break;

    }

}

double computeSGPA(){

    double sum = 0.0, num;

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

        num = subject[i].grade * subject[i].credits;

        sum += num;

    }

    sgpa = sum/20;

    return sgpa;

}

void display(double sgpa){

    System.out.println("Student's name is "+name);

    System.out.println("Student's usn is "+usn);

```

```
        System.out.println("Student's SGPA is "+sgpa);  
    }  
}  
  
class A{  
  
    public static void main(String ar[]){  
  
        SGPA ob = new SGPA();  
  
        ob.getStudentDetails();  
  
        ob.getMarks();  
  
        double c = ob.computeSGPA();  
  
        ob.display(c);  
    }  
}
```

Lab Program 3

26

11/12/2023.

* 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 declare array of object

Classname obj = new Classname^[size];

```
import java.util.Scanner;  
class Book {  
    String name, author;  
    int price, num_of_pages;
```

```
    Book(String name, String author,  
        int price, int num_of_pages)
```

{

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

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

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

```
    this.num_of_pages = num_of_pages;
```

}

```
    public String toString() {
```

```
        String s = "Book name:" + name
```

```
        + "\n" + "Author name:" + author +
```

```
        "\n" + "Price:" + price + "\n" +
```

```
        "Number of pages:" + num_of_pages;
```

*

```
        return s;
```

}

}

```
class Main {
```

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

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

```
        String name, author;
```

```
        int price, num_of_pages;
```

```
        System.out.println("IBM22CS189");
```

```
System.out.println("Pannaga R Bhat");  
int i;
```

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

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

```
sc.nextLine(); // consume newline
```

```
Book ob[i] = new Book[i];
```

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

```
System.out.println("Enter the name of  
the book " + (j+1));
```

~~name = sc.nextLine();~~~~System.out.println("Enter the author name
of the book " + (j+1));~~~~author = sc.nextLine();~~~~System.out.println("Enter the price of the
book " + (j+1));~~~~price = sc.nextInt();~~~~System.out.println("Enter the number of
pages in the book " + (j+1));~~~~num_of_pages = sc.nextInt();~~~~sc.nextLine(); // consumes newline~~~~ob[i] = new Book(name, author, price,
num_of_pages);~~

Date _____
Page _____

```
for(int j=0; j<i; j++) {  
    String c = ob[i].toString();  
    System.out.print(c);  
    System.out.println("Details of the  
Book "+(j+1)+" :");  
    System.out.print(c);  
}
```

* Output :-

IBM>>>CS189

Pannaga R Bhat

Enter the number of books

1

Enter the name of the Book 1

wings of fire

Enter the author name of the Book 1

Arun Tiwari

Enter the price of the book 1

200

Enter the number of pages in the book 1

Details of the Book 1:

Book name: wings of fire

Author name: Arun Tiwari

Price: 200

Number of pages: 342

APR
X/12/20
2

Source Code

```
import java.util.Scanner;

class Book {

    String name, author;

    int price, num_of_pages;

    Book(String name, String author, int price, int num_of_pages) {

        this.name = name;
        this.author = author;
        this.price = price;
        this.num_of_pages = num_of_pages;
    }

    public String toString() {
        String s = "Book name:" + name + "\n" + "Author name:" + author + "\n" + "Price:" + price +
        "\n" + "Number of pages:" + num_of_pages;
        return s;
    }
}

class Main {

    public static void main(String a[]) {
        Scanner sc = new Scanner(System.in);
        String name, author;
        int price, num_of_pages;
        int i;
        System.out.println("1BM22CS189");
        System.out.println("Pannaga R Bhat");
        System.out.println("Enter the number of books");
    }
}
```

```
i = sc.nextInt();

sc.nextLine(); // consume the newline character left by nextInt()

Book ob[] = new Book[i];

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

    System.out.println("Enter the name of the book " + (j+1));
    name = sc.nextLine();

    System.out.println("Enter the author name of the book " + (j+1));
    author = sc.nextLine();

    System.out.println("Enter the price of the book " + (j+1));
    price = sc.nextInt();

    System.out.println("Enter the number of pages in the book " + (j+1));
    num_of_pages = sc.nextInt();

    sc.nextLine(); // consume the newline character left by nextInt()

    ob[j] = new Book(name, author, price, num_of_pages);

}

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

    String c = ob[j].toString();
    System.out.println();
    System.out.println("Details of the Book " + (j + 1)+ ":");

    System.out.print(c);

}

}
```

Lab Program 4

CLASSMATE
Date _____
Page _____

Lab Program 4.

Develop a Java Program to create an abstract class named shape that contains two integers and an empty method named `printArea()`. Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends class shape. Each one of the class contain only the method `printArea()` that prints the area.

```
import java.util.Scanner;
abstract class shape {
    int a, b;
    abstract void printArea();
}

class Rectangle extends shape {
    void printArea() {
        double area = a * b;
        System.out.println("Area of rectangle = " + area);
    }
}

class Triangle extends shape {
    void printArea() {
        System.out.println("Area of triangle = " + area);
    }
}
```

```
        }  
    }  
class circle extends shape {  
    void printArea() {  
        double area = 3.14 * a * a;  
        System.out.println("Area of circle = " + area);  
    }  
}
```

```
class Area {  
    public static void main(String args[]){  
        Scanner sc = new Scanner(System.in);  
        Shape obj = new Rectangle();  
        System.out.println("Enter the dimensions  
of the rectangle length and breadth");  
        ob1.a = sc.nextInt();  
        ob1.b = sc.nextInt();  
    }  
}
```

```
Shape obj2 = new Triangle();  
System.out.println("Enter the dimensions of  
the triangle (base and height)");  
ob2.a = sc.nextInt();  
ob2.b = sc.nextInt();
```

```
Shape obj3 = new Circle();  
System.out.println("Enter the dimensions of  
circle (radius)");  
ob3.a = sc.nextInt();
```

```
ob1.printArea();  
ob2.printArea();  
ob3.printArea();  
}  
}
```

Output:

Enter the dimensions of the rectangle
10 20

Enter the dimensions of the triangle
base and height
10 5

Enter the dimension of the circle (radius)
4

~~Area of rectangle = 200.0~~

~~Area of triangle = 25.0~~

~~Area of circle = 50.24~~

✓
02/01/24

Source Code

```
import java.util.Scanner;

abstract class Shape {
    int a, b;
    abstract void printArea();
}

class Rectangle extends Shape {
    void printArea() {
        double area = a * b;
        System.out.println("Area of rectangle = " + area);
    }
}

class Triangle extends Shape {
    void printArea() {
        double area = (a * b) / 2.0;
        System.out.println("Area of triangle = " + area);
    }
}

class Circle extends Shape {
    void printArea() {
        double area = 3.14 * a * a;
        System.out.println("Area of circle = " + area);
    }
}
```

```
}
```

```
class Area {  
    public static void main(String a[]) {  
  
        Scanner sc = new Scanner(System.in);  
        Shape ob1 = new Rectangle();  
        System.out.println("Enter the dimensions of the rectangle (length and breadth)");  
        ob1.a = sc.nextInt();  
        ob1.b = sc.nextInt();  
        Shape ob2 = new Triangle();  
        System.out.println("Enter the dimensions of the triangle (base and height)");  
        ob2.a = sc.nextInt();  
        ob2.b = sc.nextInt();  
  
        Shape ob3 = new Circle();  
        System.out.println("Enter the dimensions of the circle (radius)");  
        ob3.a = sc.nextInt();  
        ob1.printArea();  
        ob2.printArea();  
        ob3.printArea();  
    }  
}
```

Lab Program 5

CLASSMATE
Date _____
Page _____

09/01/2024

Lab Program 5

Develop a Java Program to create a class Bank that maintains two kinds of account for its customers one called savings account and the other current account.

The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

- 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 necessary methods.
 - 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 min balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
class Bank {
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        SavAcct ob = new SavAcct();
        CurAcct obI = new CurAcct();
        ob.accept();
        System.out.println("Customer 1 enter your
                           type of account");
        String s = sc.nextLine();
        sc.nextLine();
        System.out.println("Customer 2 enter
                           your type of account");
        String c = sc.nextLine();
        sc.nextLine();
        ob.saving(s);
        obI.current(c);
        ob.display(s);
        ob.display_c(c, obI.bal(c));
    }
}
```

```
class Account
{
```

```
Scanner sc = new Scanner(System.in);
int n=2, damt, wamt, damtc, wamtc;
```

```
float bal = 0.0f;
float balc = 0.0f;
String cn[] = new String[n];
int ac[] = new int[n];

void accept() {
    for(int i=0; i<n; i++) {
        System.out.println("Enter name of
                           customer " + (i+1));
        cn[i] = sc.nextLine();
        System.out.println("Enter account
                           number of customer " + (i+1));
        ac[i] = sc.nextInt();
        sc.nextLine();
    }
}

int deposit()
{
    System.out.println("Enter the deposit amount");
    damt = sc.nextInt();
    System.out.println("Amount Deposited");
    bal += damt;
    return damt;
}

int withdraw()
{
```

```
float min = 200.0;
System.out.println("Enter the withdrawal amount:");
wamt = sc.nextInt();
System.out.println("Rs " + wamt + " withdrawn successfully");
bal -= wamt;

if(bal < min)
{
    bal -= (0.01 * bal);
    System.out.println("Interest deducted as the balance is less than minimum balance");
}
return wamt;
}
```

```
void com_in()
{
    bal = bal + (float)(0.04 * bal);
    System.out.println("Interest added to balance successfully");
    return;
}
```

```
void display(String s)
{
    if(s.equalsIgnoreCase("Savings"))
        System.out.println("Name of customer I"
                           + " : " + cn[0]);
```

Date _____
Page _____

```
System.out.println("Account number  
of customer : " + ac[0]);  
System.out.println("Type of Account : " + t);  
System.out.println("Balance : " + bal);  
}  
}
```

int deposit_c()

```
{  
    System.out.print("Enter the deposit  
amount : ");
```

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

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

```
    balc += danc;
```

```
    return danc;
```

```
}
```

int withdraw_c()

```
{  
    float min = 500.0f;
```

```
    System.out.print("Enter the deposit amount");
```

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

```
    System.out.println("Rs " + wanct + " withdrawn  
successfully");
```

```
    balc -= wanct;
```

If (bal < min)

```
{
```

```
    balc -= (0.01 * balc);
```

```
    System.out.println("Interest deducted as  
the balance is less than minimum  
balance");
```

```
{  
    return -wamtC;  
}  
  
void display_c (String c, float balc) {  
    for (int i = 1; i < n; i++)  
        System.out.println("Name of customer 2"  
                           + ":" + cn[i]);  
    System.out.println("Account number of  
                      customer 2" + ":" + ac[i]);  
    System.out.println("Type of Account: " + t[i]);  
    System.out.println("Balance: " + balc);  
}  
}  
}
```

```
class cur_acct extends Account {  
    void current(String c) {  
        boolean con = true;  
        //System.out.println(c);  
        if ("current".equalsIgnoreCase(c))  
            {  
                while (con) {  
                    float d = 0.0f;  
                    float w = 0.0f;  
                    System.out.println("Menu for current  
account(Customer 2) ");  
                    System.out.println("1. Deposit");  
                    System.out.println("2. Withdraw");  
                    System.out.println("3. Exit");  
                }  
            }  
    }  
}
```

```
System.out.println("Enter your choice");
int choice = sc.nextInt();
switch(choice) {
    case 1:
        d = deposit_();
        break;
    case 2:
        w = withdraw_();
        break;
    case 3:
        con = false;
        break;
    default:
        System.out.println("Enter a valid
choice");
        continue;
}
```

```
class sav acct extends Account {
    void savings(String s) {
        boolean con = true;
        if ("savings".equalsIgnoreCase(s)) {
            while (con) {
                float d, w;
                System.out.println("Menu for savings
account(Custom 1%)");
            }
        }
    }
}
```

```
System.out.println("1. Deposit");
System.out.println("2. Withdraw");
System.out.println("3. Compute interest
    for savings account");
```

```
System.out.println("4. Exit");
```

```
System.out.println("Enter your choice");
int choice = sc.nextInt();
```

~~switch (choice) {~~

~~case 1:~~

```
d = deposit();
break;
```

~~case 2:~~

```
w = withdraw();
break;
```

~~case 3:~~

```
com_in();
break;
```

~~case 4:~~

```
con = false;
break;
```

~~default:~~

```
System.out.println("Enter a valid choice");
```

~~continue;~~

~~}~~

~~}~~

Output:-

Enter name of customer 1-

raju

Enter account number of customer 1

19

Enter name of customer 2

ram

Enter account number of customer 2

20

Customer 1 enter your type of account

savings

Customer 2 enter your type of account

current

Menu for savings account (customer 1)

1. Deposit
2. Withdraw

3. Compute interest for savings account

4. Exit

Enter your choice

1

Enter the deposit amount:

1000.

Amount deposited

Menu for savings account (customer 1)

1. Deposit

2. Withdraw

3. Compute interest for savings account

4. Exit

Enter your choice

2

Enter the withdrawal amount:

170

Rs 170 withdrawn successfully

Menu for savings account (Customer 1)

1. Deposit

2. Withdraw

3. Compute interest for savings account

4. Exit

Enter your choice

3

Interest added to balance successfully

Menu for savings account (Customer 1)

1. Deposit

2. Withdraw

3. Compute interest for savings account

4. Exit

Enter your choice

4.

Menu for current account (Customer 2)

1. Deposit

2. Withdraw

3. Exit

Enter your choice

1

Enter the deposit amount:

1000

Amount Deposited

Menu for current account (Customer 2)

1. Deposit

2. Withdraw

3. Exit

Enter your choice

2.

Enter the withdrawal amount:

600

Rs 600 withdrawn successfully

Interest deducted as the balance is less
than minimum balance

Menu for current account (Customer 2)

1. Deposit

2. Withdraw

3. Exit

Enter your choice

3

Name of customer 1: Raju

Account number of customer 1: ~~123456789~~¹⁹

Type of Account: savings

Balance: 863.2

Name of customer 2: Ram

Account number of customer 2: 20

Type of Account: current

Balance: 396.0

12/01/24

Source Code

```
import java.util.Scanner;

class Bank{

    public static void main(String arg[]){
        Scanner sc = new Scanner(System.in);
        sav_acct ob = new sav_acct();
        cur_acct ob1 = new cur_acct();
        ob.accept();
        System.out.println("Customer 1 enter your type of account");
        String s = sc.next();
        sc.nextLine(); // Consume the newline character
        System.out.println("Customer 2 enter your type of account");
        String c = sc.next();
        sc.nextLine(); // Consume the newline character
        ob.saving(s);
        ob1.current(c);
        ob.display(s);
        ob.display_c(c,ob1.balc);
    }
}

class Account
{
    Scanner sc = new Scanner(System.in);
    int n = 2, damt, wamt, damtc, wamtc;
    float bal=0.0f;
    float balc=0.0f;
    String cn[] = new String[n];
```

```
int ac[] = new int[n];

void accept() {
    for (int i = 0; i < n; i++) {
        System.out.println("Enter name of customer " + (i + 1));
        cn[i] = sc.nextLine();
        System.out.println("Enter account number of customer " + (i + 1));
        ac[i] = sc.nextInt();
        sc.nextLine();
    }
}

int deposit()
{
    System.out.println("Enter the deposit amount: ");
    damt = sc.nextInt();
    System.out.println("Amount Deposited");
    bal+=damt;
    return damt;
}

int withdraw()
{
    float min = 200.0f;
    System.out.println("Enter the withdrawal amount: ");
    wamt = sc.nextInt();
    System.out.println("Rs "+wamt+" withdrawn successfully");
    bal-=wamt;
```

```

if(bal<min)
{
    bal-=(0.01*bal);
    System.out.println("Interest deducted as the balance is less than minimum balance");
}

return wamt;
}

void com_in(){
    bal = bal+(float)(0.04*bal);
    System.out.println("Interest added to balance successfully");
    return;
}

void display(String s){
    if(s.equalsIgnoreCase("savings"))
    {
        System.out.println("Name of customer 1"+": " + cn[0]);
        System.out.println("Account number of customer 1"+": " + ac[0]);
        System.out.println("Type of Account: " + s);
        System.out.println("Balance: " + bal);
    }
}

//Current account calculation.

int deposit_c()
{

```

```

System.out.println("Enter the deposit amount: ");
damtc = sc.nextInt();
System.out.println("Amount Deposited");
balc+=damtc;
return damtc;
}

int withdraw_c()
{
float min = 500.0f;
System.out.println("Enter the withdrawal amount: ");
wamtc = sc.nextInt();
System.out.println("Rs "+wamtc+" withdrawn successfully");
balc-=wamtc;

if(balc<min)
{
balc-=(0.01*balc);
System.out.println("Interest deducted as the balance is less than minimum balance");
}
return wamtc;
}

void display_c(String c,float balc) {
for (int i = 1; i < n; i++) {
System.out.println("Name of customer 2 ":" " + cn[i]);
System.out.println("Account number of customer 2" ":" " + ac[i]);
System.out.println("Type of Account: " + c);
System.out.println("Balance:" +balc);
}
}

```

```
}

}

}

class cur_acct extends Account

{

void current(String c){

    boolean con = true;

    if("current".equalsIgnoreCase(c))

    {

        while (con) {

            float d=0.0f;

            float w=0.0f;

            System.out.println("  Menu for current account(Customer 2)  ");

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

            System.out.println("2. Withdraw");

            System.out.println("3. Exit");

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

            int choice = sc.nextInt();




switch (choice) {

    case 1:

        d = deposit_c();

        break;

    case 2:

        w = withdraw_c();

        break;
    }
}
```

```

        case 3:
            con = false;
            break;
        default:
            System.out.println("Enter a valid choice");
            continue;
    }
}

}
}

```

```

class sav_acct extends Account
{
void saving(String s {

    boolean con = true;
    if ("savings".equalsIgnoreCase(s)) {
        while (con) {
            float d, w;
            System.out.println(" Menu for savings account(Customer 1) ");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Compute interest for savings account");
            System.out.println("4. Exit");
            System.out.println("Enter your choice");
            int choice = sc.nextInt();

```

```
switch (choice) {  
    case 1:  
        d = deposit();  
        break;  
    case 2:  
        w = withdraw();  
        break;  
    case 3:  
        com_in();  
        break;  
    case 4:  
        con = false; // Set con to false  
        break; // Exit the switch and loop  
    default:  
        System.out.println("Enter a valid choice");  
        continue;  
    }  
}  
}  
}  
}
```

Lab Program 6

23/11/24 Lab 6 Week 7

. 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 derived from student, stores SEE marks of 5 courses. Import the two packages in a file that declares final marks of n students in all 5 five courses.

Student class

```
package FTEs;
import java.util.Scanner;
public class Student
{
    protected String usn, name;
    protected int sem;
    public void input()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your USN:");
        usn = sc.nextLine();
        System.out.println("Enter your name:");
        name = sc.nextLine();
        System.out.println("Enter your semester:");
        sem = sc.nextInt();
    }

    public void display()
    {
        System.out.println("USN: " + usn);
        System.out.println("NAME: " + name);
        System.out.println("SEMESTER: " + sem);
    }
}
```

Internals class

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

External class

```
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class External extends Internals
{
    protected int sm[] = new int[5];
    protected int fm[] = new int[5];
    public void accept_smarks()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Semester marks
                           of 5 subjects");
    }
}
```

```
Scanner sc = new Scanner(System.in);
System.out.println("Enter Semester marks
                   of 5 subjects");
```

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

```
    sm[i] = sc.nextInt();
```

{

}

```
public void calc()
```

{

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

```
    fm[i] = (sm[i]/2) + super.m[i];
```

}

```
public void displayfm()
```

{

```
display();
```

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

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

}

}

{

* Main class

```
import CIE.Student;  
import CIE.Internals;  
import SEE.External;
```

```
public class Main
```

```
{
```

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

```
        int n=1;
```

```
        External ob[] = new External[n];
```

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

```
            ob[i] = new External();
```

```
            ob[i].input();
```

```
            System.out.println("Enter CIE marks");
```

```
            ob[i].accept_imarks();
```

```
            System.out.println("Enter SEE marks");
```

```
            ob[i].accept_smarks();
```

```
}
```

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

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

~~```
 ob[i].outbm();
```~~~~```
            ob[i].displaybm();
```~~~~```
}
```~~~~```
}
```~~

Output :-

Enter your USN:

IBM22CS189

Enter your name:

Pannaga R Bhat

Enter your semester:

3

Enter CIE Marks

Enter internal marks of 5 subjects

50

49

48

47

50

Enter SEE Marks

Enter semester marks of 5 subjects

100

99

98

97

95

Displaying Data:

USN: IBM22CS189

NAME: Pannaga R. Bhat

SEMESTER: 3

Subject 1 marks: 100

Subject 2 marks: 98

Subject 3 marks: 97

Subject 4 marks: 95

Subject 5 marks: 97

24.01.2022

Source Code

```
package SEE;

import CIE.Internals;

import java.util.Scanner;

public class External extends Internals

{

    protected int sm[] = new int[5];

    protected int fm[] = new int[5];

    public void accept_smarks()

    {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Semester marks of 5 subjects");

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

        {

            sm[i] = sc.nextInt();

        }

    }

    public void calfm()

    {

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

        {

            fm[i] = (sm[i]/2) + super.m[i];

        }

    }

    public void displayfm()

    {

        display();

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

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

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

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

```
public class Student
{
    protected String usn,name;
    protected int sem;
    public void input()
    {
        Scanner sc =new Scanner(System.in);
        System.out.println("Enter your usn: ");
        usn = sc.nextLine();
        System.out.println("Enter your name: ");
        name = sc.nextLine();
        System.out.println("Enter your semester: ");
        sem = sc.nextInt();
    }

    public void display()
    {
        System.out.println("USN: " +usn);
        System.out.println("NAME: " +name);
        System.out.println("SEMESTER: " +sem);
    }
}

import CIE.Student;
import CIE.Internals;
import SEE.External;

public class Main
```

```
{  
    public static void main(String ar[])  
    {  
        int n = 1;  
        External ob[] = new External[n];  
  
        for(int i = 0;i<n;i++)  
        {  
            ob[i] = new External();  
            ob[i].input();  
            System.out.println("Enter CIE Marks");  
            ob[i].accept_imarks();  
            System.out.println("Enter SEE Marks");  
            ob[i].accept_smarks();  
        }  
        System.out.println("Displaying Data:");  
        for(int i = 0;i<n;i++)  
        {  
            ob[i].calfm();  
            ob[i].displayfm();  
        }  
    }  
}
```

Lab Program 7

classmate
Date _____
Page _____

10/1/2024 Lab Program 7

write a program that demonstrates handling of exceptions in inheritance tree. create a base class "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
{
    public WrongAge()
    {
        System.out.println("Wrong Age");
    }
    public WrongAge(String s)
    {
        super(s);
    }
}

class Father
{
```

```
Scanner sc;  
int fage;  
Father() throws WrongAge  
{  
    sc = new Scanner(System.in);  
    System.out.println("Enter father's age");  
    fage = sc.nextInt();  
    if(fage < 0)  
        throw new WrongAge("Age cannot  
        be negative");  
}
```

```
public void display()  
{  
    System.out.println("Father's Age= " +  
        fage);  
}
```

```
class Son extends Father  
{
```

```
    int sage;  
    Son() throws WrongAge  
{  
        super();  
        System.out.println("Enter Son's Age");  
        sage = sc.nextInt();  
    }
```

```
    if(sage > fage)  
        throw new WrongAge("Son's age
```

cannot be greater than father's
age");

else if (sage > 0)

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

}
public void display()

{
super.display();

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

class Main

{
public static void main(String args)
throws Exception

{

try {

Son s = new Son();

s.display();

}

catch (WrongAge e)

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

}

{

}

① Enter father's age

52

Enter Son's Age

19

Father's Age = 52

Son's Age = 19

② Enter father's age

38

Enter Son's Age

40

Exception: Son's age cannot be
greater than father's age

~~20/01/24~~

Source Code

```
import java.util.Scanner;

class WrongAge extends Exception
{
    public WrongAge()
    {
        System.out.println("Wrong Age");
    }

    public WrongAge(String s)
    {
        super(s);
    }
}

class Father
{
    Scanner sc;
    int fage;
    Father() throws WrongAge
    {
        sc = new Scanner(System.in);
        System.out.println("Enter father's age");
        fage = sc.nextInt();

        if(fage < 0)
            throw new WrongAge("Age cannot be negative");
    }
}
```

```
}

public void fdisplay()
{
    System.out.println("Father's Age= "+fage);
}

}

class Son extends Father
{
    int sage;
    Son() throws WrongAge
    {
        super();
        System.out.println("Enter Son's Age");
        sage = sc.nextInt();

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

    public void display()
    {
        super.fdisplay();
        System.out.println("Son's Age= "+sage);
    }
}
```

```
class Main
{
    public static void main(String a[]) throws Exception
    {
        try{
            Son s = new Son();
            s.display();
        }
        catch(WrongAge e)
        {
            System.out.println("Exception:"+e.getMessage());
        }
    }
}
```

Lab Program 8

30.
8/21/2024

* WAP which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

class A1 extends Thread

{

 public void run()

{

 try

 { int i=0;

```
while(i<20)
{
    System.out.println("BMS college of
        Engineering");
    Thread.sleep(10000);
    i++;
}
catch(Exception e)
{
    System.out.println("Exception occurred");
}
```

- class A2 extends Thread

```
{public void run()
{
    try
    { int i=0;
        while(i<20)
        {
            System.out.println("CSE");
            Thread.sleep(2000);
            i++;
        }
    }
    catch(Exception e)
```

```
{  
    System.out.println("Exception caught");  
}  
}  
}  
  
class Lab 8  
{  
    public static void main(String args[]){  
        A1 ob1 = new A1();  
        A2 ob2 = new A2();  
        ob1.start();  
        ob2.start();  
    }  
}
```

Output:-

BMS College of Engineering

CSE

CSF

BMS College of Engineering

BMS College of Engineering

CSE

CSE

CSE

.

.

20th BMS College of Engin

CSE

Source Code

```
class A1 extends Thread
{
    public void run()
    {
        try
        {
            int i= 0;
            while(i<20)
            {
                System.out.println("BMS College of Engineering");
                Thread.sleep(2000);
                i++;
            }
        }
        catch(Exception e)
        {
            System.out.println("Exception occured");
        }
    }
}
```

```
class A2 implements Runnable
```

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

```
int i= 0;
while(i<20)
{
    System.out.println("CSE");
    Thread.sleep(1000);
    i++;
}
}

catch(Exception e)
{
    System.out.println("Exception caught");
}
}

}

class Lab8
{
public static void main(String arg[])
{
    A1 ob1 = new A1();
    Runnable r = new A2();
    Thread ob2 = new Thread(r);

    ob1.start();
    ob2.start();
}
}
```

Lab Program 9

13.

Lab Program 9

- * write a program that creates a user interface to perform Integer Division. The user enters two numbers in text field and the result is displayed by clicking on division button. The exception is generated when second number is ~~genera~~^{zero} and both text field is empty.

Report:-

Operations

JFrame, setSize, setLayout, setDefaultCloseOperation
JLabel, JTextField, addFrame
addActionListener, setText.

```

int ans = a/b;
alab.setText("In A = " + a);
blab.setText("In B = " + b);
anslab.setText("In Ans = " + ans);
}

catch(NumberFormatException e) {
    alab.setText(" ");
    blab.setText(" ");
    anslab.setText(" ");
    if Input is String → err.setText("Enter Only Integers");
    if err displays → err.setText("Enter Only Integers");
    if a is 0 → err.setText("B should be NON zero");
}

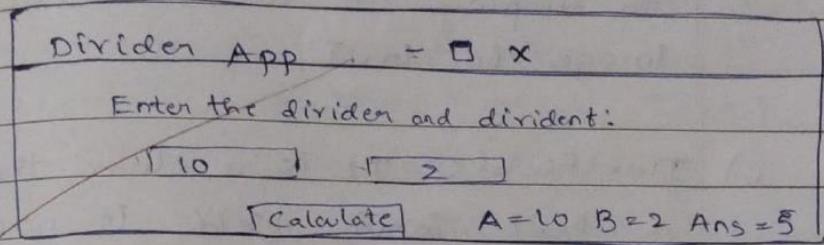
catch(ArithmeticException e) {
    alab.setText(" ");
    blab.setText(" ");
    anslab.setText(" ");
    if deno is 0 → err.setText("B should be NON zero");
}

jfrm.setVisible(true); // display on screen

} // end of string demand
public static void main(String ar[])
{
    swingUtilities.invokeLater(new
        Runnable());
}

```

```
public void run() {  
    new Swing Demo();  
}  
}  
}
```



Report AWT Functions

- 1) **JFrame**: It is a class in Java that is part of the swing library, which is used for creating GUI in java application.
- 2) **setSize()**: is a method which is used with components such as JFrame to set their size.
- 3) **setLayout()**: It is a method which is used to set the layout manager for a container, such as JFrame.

- 11) `setDefaultCloseOperation()`: It is a method which is used to specify default close operation for a JFrame.
- 5) JLabel :- It is a class which is used to display non-editable text as image in GUI.
- 6) JTextField : It is a class that provides a text input field in a GUI.
- 7) `addActionListener()` is a method that is used to register an ActionListener for a component, typically for a button or any other interactive component that generates action events.
- 8) ~~setText()~~ : used to set text content of text based components.

For
30.02.2k

Source Code

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

class SwingDemo{
    SwingDemo(){
        // create jframe container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // text label
        JLabel jlab = new JLabel("Enter the divider and divident:");

        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        // calc button
        JButton button = new JButton("Calculate");

        // labels
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
```

```
JLabel anslab = new JLabel();
// add in order :)
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("Ans = "+ 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 SwingDemo();
        }
    });
}
}
```

Lab Program 10

Page _____

* Lab Program 10 :- Interprocess communication

```
class B
{
    int n;
    boolean val = false;
    synchronized int get()
    {
        while(!val)
        {
            try
            {
                System.out.println("Consumer Waiting");
                wait();
            }
            catch(InterruptedException e)
            {
                System.out.println("Exception Caught");
            }
        }
        System.out.println("Inimate producer");
        notify();
        return n;
    }
}

synchronized void put(int n)
{
    while(val)
    {
        try
        {
            System.out.println("Producer Waiting");
            wait();
        }
    }
}
```

```
catch(InterruptedException e) {  
    System.out.println("Interrupt caught");  
}
```

```
this.n = n;
```

```
val = true;
```

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

```
System.out.println("Inimate consumer");  
notify();
```

```
}
```

class Producer implements Runnable

```
{
```

```
Q q;
```

```
Producer(Q q) {
```

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

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

```
}
```

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

```
int i = 0;
```

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

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

```
}
```

```
}
```

class Consumer implements Runnable

```
{
```

```
Q q;
```

Date _____
Page _____

```

consumer(Q q) {
    this.q = q;
    new Thread(this, "Consumer").start();
}

public void run() {
    int i = 0;
    while(i < 15) {
        System.out.println("Consumed: " +
            q.get());
        i++;
    }
}

```

```

class Lab10 {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("ctrl + C to stop");
    }
}

Output:-
ctrl + C to stop
Put: 0
Get: 0
Put: 1
Consumed: 0
Get: 1

```

Consumed: 1

Put: 2

Count: 2

Consumed: 2

Put: 3

Count: 3

Consumed: 3

Put: 3

Count: 3

;

Put: 14

Count: 14

Consumed: 14

* 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)

{

Date _____
Page _____

```
        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()");

Date _____
Page _____

```
a.rast();  
}  
void last()  
{  
    System.out.println("Inside B.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 aa[])  
    {  
        new deadlock();  
    }  
}
```

* Output

Racing Thread entered B.box

Main Thread entered A.box

Racing Thread trying to call A.last()
Inside A.last

Back in other thread

Main Thread trying to call B.last()
Inside B.last

Back in main Thread

Fix
13.02.24

Source Code

Inter-process Communication

```
class Q {  
    int n;  
    boolean val = false;  
    synchronized int get() {  
        while (!val) {  
            try {  
                System.out.println("Consumer Waiting");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("Exception Caught");  
            }  
        }  
        System.out.println("Got: " + n);  
        val = false;  
        //System.out.println("Inimate Producer");  
        notify();  
        return n;  
    }  
  
    synchronized void put(int n) {  
        while (val) {  
            try {  
                //System.out.println("Producer Waiting");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("Exception Caught");  
            }  
        }  
        val = true;  
        n;  
    }  
}
```

```
        System.out.println("InterruptedException caught");
    }
}
this.n = n;
val = true;
System.out.println("Put: " + n);
//System.out.println("Inimate Consumer");
notify();
}
}
```

```
class Producer implements Runnable {
    Q q;
    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }
    public void run() {
        int i = 0;
        while (i < 15)
            q.put(i++);
    }
}
```

```
class Consumer implements Runnable {
    Q q;
    Consumer(Q q) {
```

```
this.q = q;
new Thread(this, "Consumer").start();
}

public void run() {
    int i = 0;
    while (i < 15) {
        System.out.println("Consumed: " + q.get());
        i++;
    }
}

class Lab10 {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Ctrl-C to stop");
    }
}
```

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 B.last");
}

class deadlock implements Runnable
{
    A a = new A();
```

```
B b = new B();
deadlock()
{
    Thread.currentThread().setName("Main Thread");

    Thread t = new Thread(this, "RacingThread");
    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 aa[])
{
    new deadlock();
}
}
```