

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



## **LAB OBSERVATION**

### **OBJECT ORIENTED JAVA PROGRAMMING 23CS3PCOOJ**

### **PROGRAMS WITH OUTPUTS**

*Submitted by:*

**CHARAN G - 1BM22CS078**

**III SEMESTER 3 'B'**

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

8/1/24

## LAB Prg - 1 QUADRATIC EQUATION

Program ::

```
import java.util.Scanner;  
class Quad  
{
```

```
    double a,b,c;
```

```
    Quad (double a, double b, double c)
```

```
    {
```

```
        this.a = a;
```

```
        this.b = b;
```

```
        this.c = c;
```

```
}
```

```
    void calc()
```

```
{
```

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

```
    double root1;
```

```
    double root2;
```

```
    if (a == 0)
```

```
{
```

```
    System.out.print("Please enter valid  
quadratic equation");
```

```
}
```

```
else
```

```
{
```

```
    if (d == 0)
```

```
{
```

```
        System.out.println("Roots are real & equal");  
        root1 = (-b)/(2*a); System.out.print(root1); }
```

else if ( $d \geq 0$ )

{

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

    root1 =  $(-b + \sqrt{d}) / (2 * a)$ ;

    root2 =  $(-b - \sqrt{d}) / (2 * a)$ ;

    System.out.println (root1 + " " + root2);

}

else

{

    root1 =  $(-b) / (2 * a)$ ;

    root2 =  $\text{Math.abs}(d) / (2 * a)$ ;

    System.out.println ("Roots are imaginary");

    System.out.println (root1 + "i" + root2);

    System.out.println (root1 + "-i" + root2);

}

}

class Quadratic

{

    public static void main (String [] args)

{

        Scanner input = new Scanner (System.in);

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

        double a = input.nextDouble();

        System.out.print ("Enter value of b: ");

        double b = input.nextDouble();

        System.out.print ("Enter value of c: ");

        double c = input.nextDouble();

        Quadratic quadratic = new Quadratic (a, b, c);

        quadratic.calculate (); input.close ();

}

}

Output ::

① Enter value of a : 2

Enter value of b : 4

Enter value of c : 2

Roots are real and equal

Roots -4.0

② Enter value of a : 1

Enter value of b : 5

Enter value of c : 6

Roots are real and distinct

-2.0 -3.0

③ Enter value of a : 3

Enter value of b : 1

Enter value of c : 4

Roots are imaginary

-0.1666666 + i 7.833333

-0.1666666 - i 7.833333

④ Enter value of a : 0

Enter value of b : 4

Enter value of c : 5

Please enter valid quadratic equation.

8/1/24

LAB PRGM - 2  
Student SGPA

Program :-

```
import java.util.Scanner;  
class Student {  
    String usn, name;  
    int n;  
    int marks[], credits[];  
  
    void inputMarks()  
    {  
        Scanner input = new Scanner(System.in);  
        System.out.print("Enter USN: ");  
        usn = input.nextLine();  
        System.out.print("Enter Name: ");  
        name = input.nextLine();  
        System.out.print("Enter no. of subjects: ");  
        n = input.nextInt();  
        credits = new int[n];  
        marks = new int[n];  
        for (int i=0; i<n; i++)  
        {  
            System.out.print("Marks subject " + (i+1) + ":");  
            marks[i] = input.nextInt();  
            System.out.print("Credits subject " + (i+1) + ":");  
            credits[i] = input.nextInt();  
        }  
    }  
}
```

g -

int gradePoints (int marks)

{

```
    if (marks >= 90) return 10;  
    else if (marks >= 80) return 9;  
    else if (marks >= 70) return 8;  
    else if (marks >= 60) return 7;  
    else if (marks >= 50) return 6;  
    else return 0;
```

}

double gpa()

{

```
    double total credits = 0, sum = 0;
```

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

{

```
    total credits = total credits + credits[i];
```

```
    sum = sum + gradePoints(marks[i]) * credits[i];
```

}

```
    return (sum / total credits);
```

}

void display()

{

```
    system.out.println("USN : " + usn);
```

```
    system.out.println("Name : " + name);
```

}

class Main

{

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

{

```
        Student student = new Student();
```

student. input Marks();

student. display();

System.out.println("SGPA : " + student. sgpa);

3  
3

Enter USN: 1BM22CS078

Ente Name: Charan G

Ente Ente number of subjects: 8

Marks subject 1: 98

Credits subject 1: 4

Marks subject 2: 97

Credits subject 2: 4

Marks subject 3: 92

Credits subject 3: 3

Marks subject 4: 93

Credits subject 4: 3

Marks subject 5: 91

Credits subject 5: 3

Marks subject 6: 85

Credits subject 6: 1

Marks subject 7: 95

Credits subject 7: 1

Marks subject 8: 92

Credits subject 8: 1

USN: 1BM22CS078

Name: Charan G

SGPA: 9.95

X 100/124

8/1/24

## LAB PROGRAM-3

### BOOK OBJECTS

Create a class Book that contains four members: name, author, price, and num-pages.

Include a constructor to set the values for the members. Include methods to set and get the details of objects. Include a toString() method that could display the complete details of the Book.

Develop a main program to create n book objects.

#### A. Program:

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

```
class Books
```

```
{
```

```
    String name, author;
```

```
    int price, num-pages;
```

```
Books (String name, String author, int price, int num-pages)
```

```
{
```

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

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

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

```
    this.num-pages = num-pages;
```

```
}
```

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

```
Books () // Default constructor
```

void accept()

{

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

name = input.nextLine();

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

author = input.nextLine();

System.out.print("Enter price of book : ");

price = input.nextInt();

System.out.print("Enter no. of pages in the book : ");

num-pages = input.nextInt();

System.out.print("\n");

}

public String toString()

{

String name, author; int price, num-pages;

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

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

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

num-pages = "Number of pages : " + this.num-pages + " pages\n";

return name + author + price + num-pages;

}}

class BookRun

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

System.out.print("Enter number of books : ");

int n = input.nextInt();

System.out.print("\n");

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

```
for (int i=0; i<n; i++)  
{ System.out.println ("Book " + (i+1) + ":");  
book[i] = new Books();  
book[i].accept();  
}
```

```
for (int i=0; i<n; i++)  
{  
System.out.println ("Book " + (i+1) + ":" + "\n"  
+ book[i] + "\n");  
}
```

```
input.close();
```

```
}
```

```
y + input.read() + " " + "Name that's already
```

```
in the list." + " Name what's written" + " Enter
```

```
"Book " + input.read() + " Enter" + " Enter"
```

OUTPUT

Enter number of books : 2

Book 1 :

Enter name of book : Harry Potter and The Chamber  
of Secrets

Enter name of author : J. K. Rowling

Enter price of the book : 418

Enter no. of pages in the book : 341

Book 2 :

Enter name of book : Adventures of Ulubin Noothan

Enter name of author : Sharad

Enter price of the book : 300

Enter no. of pages in the book : 250

Book 1 :

Book name : Harry Potter and The Chamber of Secrets

Author name : J. K. Rowling

Price : 418 Rs

Number of pages : 341 pages

Book 2 :

Book name : Adventures of Vibin Nethan

Author name : Charan G

Price : 300 Rs

Number of pages : 250 pages

2/1/24

## LAB PROGRAM-4

### Abstract Shape Class

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.

#### A. PROGRAM:

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

```
abstract class Shape
```

```
{
```

```
    double x, y;
```

```
    Shape(double x, double y)
```

```
{
```

```
    this.x = x;
```

```
    this.y = y;
```

```
}
```

```
    abstract void printArea();
```

```
}
```

~~```
class Rectangle extends Shape
```~~~~```
{
```~~~~```
    Rectangle(double x, double y)
```~~~~```
{ super(x, y); }
```~~~~```
    void printArea()
```~~~~```
{
```~~~~```
        System.out.println("Area of Rectangle : " +  
                           (x * y) + " square units \n");
```~~~~```
}
```~~~~```
}
```~~

```
class Triangle extends Shape
{
```

```
    Triangle(double x, double y)
```

```
    { super(x, y); }
```

```
    void printArea()
```

```
    System.out.println("Area of Triangle : " +
```

```
(0.5 * x * y) + " square units \n");
```

```
}
```

```
class Circle extends Shape
```

```
{
```

```
    Circle(double x)
```

```
    { super(x, 0); }
```

```
    void printArea()
```

```
{
```

```
System.out.println("Area of Circle : " + (3.14 * x * x))
```

```
+ " square units \n");
```

```
}
```

```
class ShapeRun
```

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

```
{
```

```
    double x, y;
```

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

length and width

```
System.out.println("Enter dimensions of Rectangle : ");
```

```
x = input.nextDouble();
```

```
y = input.nextDouble();
```

```
Rectangle rectangle = new Rectangle(x, y);
```

```
rectangle.printArea();
```

height & base

```
System.out.println("Enter dimensions of Triangle:");
x = input.nextDouble();
y = input.nextDouble();
Triangle triangle = new Triangle(x,y);
triangle.printArea();
```

radius

```
System.out.println("Enter dimension of Circle:");
x = input.nextDouble();
Circle circle = new Circle(x);
circle.printArea();
```

input.close();

}

}

OUTPUT →

Enter length and width of Rectangle:

10 ~~20~~

20

Area of Rectangle: 200.0 square units

Enter height and base of Triangle:

10

20

Area of Triangle: 100.0 square units

Enter radius of Circle:

10

~~Area of Circle: 314.0 square units~~

~~10/2<sup>2</sup>~~

22/11/24

LAB PROGRAM - 5BANK

class account with savings &amp; current account

Interest % Deposit → savings

Service charge % minimum Balance → current

## A. PROGRAM :

import java.util.Scanner;

class Account

{

String customerName;

long accountNumber;

String accountType;

double balance;

Account (String customerName, long accountNumber,  
String accountType, double balance)

{

this.customerName = customerName;

this.accountNumber = accountNumber;

this.accountType = accountType;

this.balance = balance;

}

void deposit(double amount)

{

balance = balance + amount;

System.out.println("Deposit of " + amount + " was  
successful. Balance : "  
+ balance);

}

```
void displayBalance()
```

```
{
```

```
    System.out.println ("Account Number: " + accountNumber +  
                        "Customer Name: " + customerName +  
                        "Account Type: " + accountType +  
                        "Balance: " + balance);
```

```
}
```

```
}
```

```
class SavingsAccount extends Account
```

```
{
```

```
    SavingsAccount (String customerName, long accountNumber,  
                    double balance)
```

```
{
```

```
    super (customerName, accountNumber, "Savings", balance);
```

```
}
```

```
void interest (double rate)
```

```
{
```

```
    double interest = balance * rate / 100;
```

```
    balance = balance + interest;
```

```
    System.out.println ("Interest computed and deposited.  
                        Updated balance: " + balance);
```

```
}
```

```
void withdraw (double amount)
```

```
{
```

```
    if (amount <= balance)
```

```
{
```

```
        balance = balance - amount;
```

```
        System.out.println ("Withdrawal of " + amount +  
                            "successful. Updated balance:  
                            " + balance);
```

```
}
```

else  
{

    System.out.println("Insufficient funds.

    Withdraw failed. (" );

}  
}  
}

class CurrentAccount extends Account  
{

    double minimumBalance;

    double serviceCharge;

    CurrentAccount (String customerName, long accountNumber,  
        double balance, double minimumBalance,  
        double serviceCharge)

}

    super (customerName, accountNumber, "Current", balance);

    this.minimumBalance = minimumBalance;

    this.serviceCharge = serviceCharge;

}

void checkMinimumBalance()

{

    if (balance < minimumBalance)

{

        balance = balance - serviceCharge;

        System.out.println("In Minimum balance not maintained.

        Service charge imposed. Balance: " + balance);

}

    else

{

        System.out.println("In Minimum balance maintained.

        Service charge not imposed.  
        Balance: " + balance);

} }

~~void change(double amount)~~

{

balance = balance - amount;

System.out.println("Withdrawal of " + amount +  
" successful. Updated balance: " + balance);

}

}

Bank

~~class Bank~~

{

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

{

~~Scanner input = new Scanner(System.in);~~

~~System.out.println("Savings Account: ");~~

~~System.out.println("Enter customer name: ");~~

~~String name = input.nextLine();~~

~~System.out.println("Enter account number: ");~~

~~long no = input.nextLong();~~

~~System.out.println("Enter balance: ");~~

~~double balance = input.nextDouble();~~

~~SavingsAccount SA = new SavingsAccount(name,  
no, balance);~~

~~System.out.println("\nCurrent Account: ");~~

~~System.out.println("Enter customer name: ");~~

~~~~String~~ name = input.nextLine();~~

System.out.println("Enter account number : ");  
 no = input.nextInt();

System.out.println("Enter balance : ");

balance = input.nextInt();

System.out.println("Enter minimum Balance : ");

double minBalance = input.nextInt();

System.out.println("Enter min charge : ");

double charge = input.nextInt();

Current Account CA = new Current Account

(name, no, balance, minBalance,  
 charge);

System.out.println("Enter deposit for savings : ");

double DS = input.nextInt();

SA.deposit(DS);

System.out.println("Enter interest rate of savings : ");

double IS = input.nextInt();

SA.interest(IS);

System.out.println("Enter withdraw amount of savings : ");

double WS = input.nextInt();

SA.withdraw(WS);

System.out.println("Enter deposit of current : ");

double DC = input.nextInt();

CA.deposit(DC);

System.out.println("Enter withdraw of current : ");

double WC = input.nextInt();

CA.cheque(WC);

```
System.out.println("In Fixed Balances : ");
System.out.println("Saving Account : ");
SA.displayBalance();
```

```
System.out.println("In Current Account : ");
CA.displayBalance();
```

{  
}

## Output

Saving Account :

Enter customer name : shawn G

Enter account number : 100

Enter balance : 50000

Current Account :

Enter customer name : sangethi

Enter account number : 200

Enter balance : 150000

Enter minm balance : 20000

Enter service charge : 10

Enter deposit amount for saving account : 10000

Deposit of 10000.0 was successful. Balance : 60000.0

Enter interest rate for Saving Account : 5

Interest computed and deposited. Update balance : 63000.0

Enter withdraw amount for saving account : 2000

Withdrawal of 2000.0 was successful.

Updated balance : 61000.0

Enter deposit amount for current account : 20000

Deposit of 20000.0 successful. Updated balance:

170000.0

Enter withdrawal amount for current Account : 50000

Withdrawal of 50000.0 successful. Updated balance

: 120000.0

Final Balance:

Savings account:

Account Number : 100

Customer Name : Sharmi

Account Type : Savings

Balance : 61000.0

Current Account:

Account Number : 200

Customer name : Langethra

Account Type : Current

Balance : 120000.0

28/10/2024

29/1/24

## Lab Prgm - 6 Packages - CIE & SEE Marks

CIE Package = On CIE folder :-

Student.java :-

```
package CIE;
public class Student {
    public String usn, name;
    public int sem;
    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
```

Internals.java :-

```
package CIE;
import CIE.Student;
public class Internals extends Student {
    public double [] internalMarks = new double [5];
    public Internals(String usn, String name, int sem, double []
        internalMarks) {
        super (usn, name, sem);
        this.internalMarks = internalMarks;
    }
}
```

SEE Package is on SEE folder.

External.java :-

```
package SEE;  
import CIE.*;
```

```
public class External extends Student  
{
```

```
    public double [] seeMarks = new double [5];  
    public External (String usn, String name, int com,  
                    double [] seeMarks)
```

}

```
    super (usn, name, com);  
    this.seeMarks = seeMarks;
```

}

Final Marks.java :-

```
import CIE.Internals;  
import SEE.External;  
import java.util.Scanner;
```

```
public class Final Marks
```

{

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

{

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

```
    System.out.print ("Enter number of students : ");
```

```
    int n = input.nextInt(); input.nextLine();
```

```
Intervals [] intervals = new Intervals [n];  
Intervals [] externals = new Intervals [n];
```

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

```
    System.out.println("Enter details for student marks of student " + (i+1));
```

```
    System.out.print("Enter USN : ");
```

```
    String usn = input.next();
```

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

```
    String name = input.next();
```

```
    System.out.print("Enter semester : ");
```

```
    int sem = input.nextInt(); input.nextLine();
```

```
    double [] internalMarks = new double [5];
```

```
    System.out.print("Enter Internal Marks for 5 courses : ");
```

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

```
{  
    internalMarks [j] = input.nextDouble();  
}
```

```
    intervals [i] = new Intervals (usn, name, sem, internalMarks);
```

```
{
```

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

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

```
    String usn = intervals [i].usn;
```

```
    String name = intervals [i].name;
```

```
    int sem = intervals [i].sem;
```

```
    double [] eeMarks = new double [5];
```

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

```
{ secMarks[j] = input.nextDouble(); }
```

```
internals[i] = new Internal(usn, name, idno, secMarks)
```

```
}
```

```
System.out.println("Input Marks of Student :");
```

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

```
{ System.out.println("Student " + (i+1) + "\n" +  
"USN : " + internals[i].usn + "\nName : "  
+ internals[i].name + "\nIDNO : " +  
internals[i].idno); }
```

~~Topper~~

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

```
{ System.out.print("Subject " + (j+1) + ":" +  
(internals[i].internalMarks[j]/2) +  
(internals[i].secMarks[j]/2)) + "\n"); }
```

```
}
```

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

~~Topper~~~~Topper~~

## OUTPUT

Enter the number of students : 2

Enter details of Student 1

Enter USN : 078

Enter Name : Charan G

Enter Semester : 3

Enter Internal Marks for 5 courses : 80  
90

80

85

100

Enter details of Student 2

Enter USN : 086

Enter name : Dhanush C

Enter Semester : 3

Enter Internal Marks for 5 courses : 80

90

90

80

80

Enter SEE Marks for 5 courses of Student 1

80

80

90

95

100

Enter SEE Marks for 5 courses of Student 2

90

90

95

100

75

## Final Marks of Students :

Student 1 :

USN : 078

Name : Charan G

Semester : 3

Subject 1 : 80.0

Subject 2 : 85.0

Subject 3 : 85.0

Subject 4 : 90.0

Subject 5 : 100.0

Student 2 :

USN : 086

Name : Dhamesh C

Semester : 3

Subject 1 : 85.0

Subject 2 : 90.0

Subject 3 : 92.5

Subject 4 : 90.0

Subject 5 : 77.5

29/11/24

# LAB - PRGM - FATHER AND SON - AGE EXCEPTION

Program: Throw exceptions if father age < 0 & son age  $\geq$  father age  
with son's class inheriting Father class's exception

PRGM: import java.util.Scanner;

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

```
class Father  
{  
    int fatherAge;  
    Father (int fatherAge) throws WrongAge  
    {  
        this.fatherAge = fatherAge;  
    }  
}
```

this.fatherAge = fatherAge;

if (fatherAge < 0)  
{

throw new WrongAge ("Father's Age cannot be  
less than zero");

}

}

class Son extends Father

{

int sonAge;

Son(int fatherAge, int sonAge) throws WrongAge

{

super(fatherAge);

this.sonAge = sonAge;

if (sonAge < 0)

{

throw new WrongAge("Son's Age cannot be  
less than zero");

}

if (sonAge >= fatherAge)

{

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

}

}

}

```
class AgeException Main
```

```
{
```

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

```
{
```

```
Scanner input = new Scanner (Scanner.in);
```

```
System.out.print ("Enter Father's age : ");
```

```
int fatherAge = input.nextInt();
```

```
System.out.print ("Enter Son's Age : ");
```

```
int sonAge = input.nextInt();
```

```
try {
```

```
Son son = new Son (fatherAge, sonAge);
```

```
System.out.println ("Father's and Son's age  
are valid");
```

```
} catch (NegativeAge e)
```

```
{
```

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

```
}
```

```
}
```

OUTPUT : ~~for valid input~~

① Enter father's age : 50

Enter son's age : 20

Father's and Son's age are valid

② Enter father's age : 50

Enter son's age : 60

Exception : ~~Wrong Age~~ : Son's age cannot be greater than or equal to Father's age

③ Enter father's age : -1

Enter son's age : 20

Exception : ~~Wrong Age~~ : Father's age cannot be less than zero

## LAB PRGM-8

### MULTITHREADING

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.

PRGM :- class BMS display extends Thread

{

public void run()

{

for (int i=0; i<3; i++)

{

System.out.println("BMS College of Engineering");

try {

Thread.sleep(10000);

} catch (InterruptedException e)

{

System.out.println("Exception! "+e);

}

}

}

Date: \_\_\_\_\_  
Page: \_\_\_\_\_

```
class CSE display extends Thread  
{  
    public void run()  
    {  
        for (int i=0; i<10; i++)  
        {  
            System.out.println("CSE");  
        }  
        try {  
            Thread.sleep(2000);  
        } catch (InterruptedException e)  
        {  
            System.out.println("Exception :" + e);  
        }  
    }  
}
```

```
class BMSThread  
{  
    public static void main(String [] args)  
    {  
        BMSThread bms = new BMSThread();  
        CSE display cse = new CSE display();  
    }  
}
```

```
bms.start();  
cse.start();
```

?

OUTPUT :

BMS College of Engineering

CSE

CS15

CS16

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS College of Engineering

19.02.24

## AWT PROGRAMS

1. Creating label, button and textfield in a frame using awt :-

A:-

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

```
public class AWTExample extends WindowAdapter
```

```
{
```

```
Frame f;
AWTExample()
{
```

```
f = new Frame();
f.addWindowListener(this);
```

```
Label l = new Label("Employee id :");
```

```
Button b = new Button("Submit");
```

```
TextField t = new TextField();
```

```
l.setBounds(20, 80, 80, 30);
```

```
f.setBounds(20, 100, 80, 30);
```

```
b.setBounds(100, 100, 80, 30);
```

```
f.add(b); f.add(l); f.add(t);
```

```
f.setSize(400, 300);
```

```
f.setTitle("Employee info");
```

~~```
f.setLayout(null);
```~~~~```
f.setVisible(true);
```~~

g

public void windowClosing(WindowEvent e)

{  
System.exit(0);

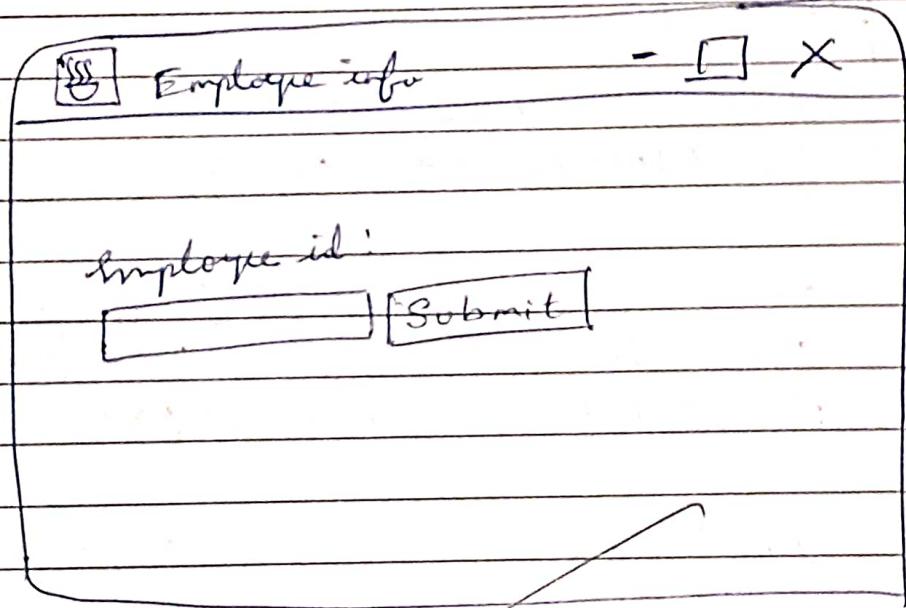
public static void main(String[] args)

{  
AWTExample awtObj = new AWTExample();

}

}

OUTPUT:



2. Create button & add a action listener for Mouse click

A. import java.awt.\*;  
import java.awt.event.\*;

public class EventHandling extends WindowAdapter  
implements ActionListener

{

Frame f;  
TextField tf;

EventHandling()

{

f = new Frame();  
f.addWindowListener(this);

tf = new TextField();  
tf.setBounds(60, 50, 170, 20);

Button b = new Button("click me");  
b.setBounds(100, 120, 80, 30);

b.addActionListener(this);

f.add(b); f.add(tf);

f.setSize(300, 300);

f.setLayout(null);

f.setVisible(true);

}

public void actionPerformed(ActionEvent e)

{

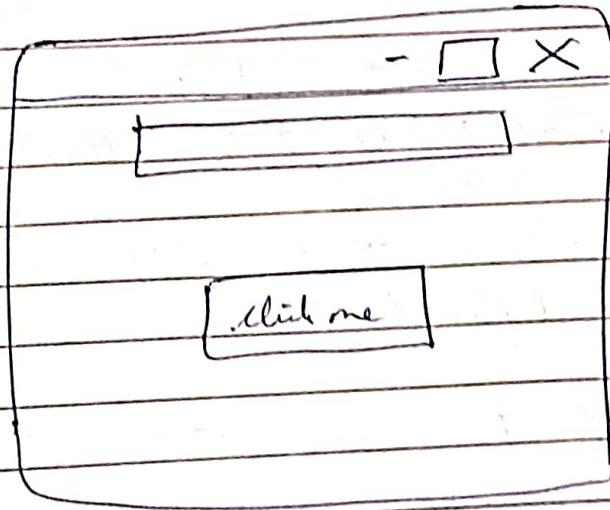
tf.setText("Welcome");

}

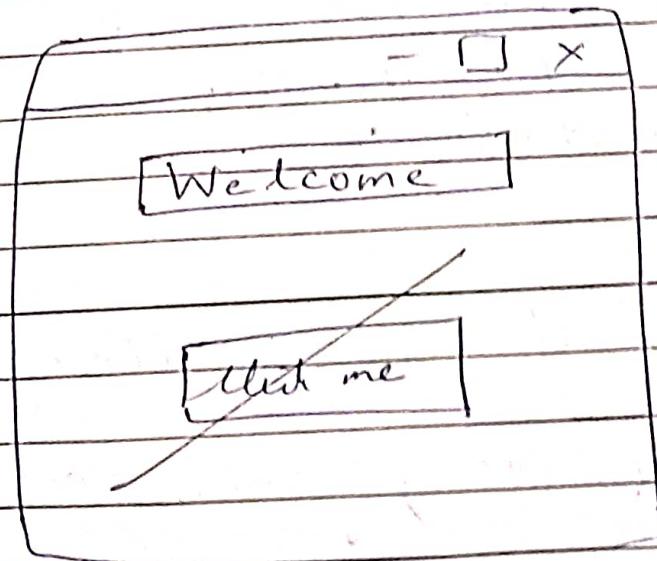
```
public void windowClose(WindowEvent e)  
{  
    System.exit(0);  
}
```

```
public static void main(String [] args)  
{  
    new EventHandling();  
}
```

OUTPUT :



after clicking on "click me"



# I O PROGRAMS

## 1. Byte Array Input :

A. import java.io.\*;

public class ByteArrayInput

{ public static void main (String [] args) throws IOException

byte [] buf = { 35, 36, 37, 38 };

ByteArrayInputStream byt = new ByteArrayInputStream  
(buf);

int k = 0;

while ( (k = byt.read ()) != -1 )

char ch = (char) k;

System.out.println ("ASCII: " + k + " special  
character: " + ch);

}

}

## OUTPUT :

ASCII: 35 special character: FF

ASCII: 36 special character: \$

ASCII: 37 special character: %

ASCII: 38 special character: &

*Ans. 02. g4  
26.02.2024*