

```

b. add(b); f.add(f);
f.setSize(width: 300, height: 300);
f.setLayout(mgr: null);
f.setVisible(b: true);
}

```

public void actionPerformed(ActionEvent event)

```

System.out.println(status);
}

```

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

Output:

Hello

click

↓

Welcome

click

Rishabh  
26.02.24

create button & add a action listener  
for mouse click.

```
import java.awt.*;
import java.awt.event.*;
public class EventHandling extends
WindowAdapter implements
ActionListener {
```

frame f;

textfield tf;

Event Handling();

f = new frame();

f.add windowListener(this);

tf = new Textfield();

tf.setBounds (x: 60, y: 50, width: 170,  
height: 20);

Button b = new Button (label: "click me");  
b.setBounds (x: 100, y: 120, width: 50,  
height: 30);

F. set Title ("Employee info");

f. set Layout (null);

f. setVisible (true);  
g.

public void windowClosing (WindowEvent)

{  
System.exit(0);}

public static void main (String [] args)  
{

AWTExample awtObj = new AWTExample()  
( );

Output

Employee Id: 1001  
Name: ABC

[ - - - ] [Submit]

26/02/24.

label, button & Textfield

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

```
public class AWTExample extends Window  
    Adapter of
```

AWTExample { }

```
f = new frame();  
f.addWindowListener(this);  
Label l = new Label("Employee id:");  
Button b = new Button("Submit");  
Text field t = new Textfield();  
l.setBounds(20, 80, 80, 30);  
t.setBounds(20, 100, 80, 30);  
b.setBounds(100, 100, 80, 30);  
f.add(b);  
f.add(l);  
f.add(t);  
f.setSize(400, 300);
```

Output :

Enter father's Age : 50

Enter son's Age : 100

father's & Son's age are valid.

Enter father's Age : 20

Enter son's Age : 22.4

wrong age: Son's age cannot be greater than or equal to father's age.

Enter father's Age : 30

Enter son's Age : -2

wrong age: Son's Age cannot be less than zero.

26.02.2014

class AgeException

public static void main (String [] args)

Scanner input = new Scanner (System.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 and son  
 age are valid");

catch (WrongAge e) {  
 System.out.println ("exception")

Son (int fatherAge, int sonAge) ;

if (fatherAge <= sonAge) throw new WrongAgeException("Son's age is greater than or equal to Father's age");

Super (fatherAge - );

this. sonAge = sonAge;

if (sonAge < 0)

throw new WrongAgeException("Son's Age cannot be less than 0");

if (sonAge >= fatherAge)

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

Output

Enter Father's Age : 50

Enter Son's Age : 10

Father's and Son's age are valid.

~~father son~~

```
import java.util.Scanner;  
class WrongAge extends String  
{  
    super(String);  
}
```

class father

```
int fatherAge;  
father(int fatherage) throws  
WrongAge
```

this.fatherage = father age;

if(fatherage < 0)  
throw new Wrongage ("father cannot be  
less than zero")

class Son extends father

int sonage;

Output:

(not printed)

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

ECG

CSG

CSE

CSE

CSE

BMS College of Engineering

```
for (int i=0; i<10; i++) {
```

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

```
    try {  
        Thread.sleep(2000);  
    } catch (InterruptedException e) {
```

```
        System.out.println("Exception");  
    }
```

Class BMUthread.

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

```
{  
    BMU display = new BMUdisplay();  
    CSE display = new CSEdisplay();
```

```
    BMU.start();
```

```
    CSE.start();
```

By

~~BMSC E~~

~~class BMSCDisplay extends~~

public void run()

{

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

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

try {

Thread.sleep(1000);

}

catch (InterruptedException e) {

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

~~class CSEDisplay extends Thread~~

public void run()

not maintained. service charge imposed.

balance =

public void withdraw (double amount)  
{  
    y (amount <= balance)  
    {  
        balance -= amount;  
        System.out.println ("withdraw successful  
                          updated " + balance);  
        checkminimumbalance ();  
    }  
    else  
        System.out.println ("insufficient funds  
                          withdraw failed");  
    }  
}

balance -= amount;

System.out.println ("withdraw successful  
                          updated " + balance);

checkminimumbalance ();

System.out.println ("insufficient funds  
                          withdraw failed");

else  
    System.out.println ("insufficient funds  
                          withdraw failed");

System.out.println ("insufficient funds  
                          withdraw failed");

Enter customer name for saving : Deva.  
" acc no. " : 20215

Enter initial balance : 10000

Deposit successful.

Enter interest for SA : 4

if amount <= balance

balance -= amount

System.out.println("Withdrew")

Successful update balance is " + balance

else

{ System.out.println("Insufficient funds"); }

class CurrentAcct extends Account

double minimumBalance;

double serviceCharge;

public CurrentAccount(String customerName,

long accountNumber, double

balance, double minimumBalance,

double serviceCharge);

private void checkMinimumBalance()

{ if (balance < minimumBalance)

balance -= serviceCharge; }

System.out.println("minimumBalance")

System.out.println ("Account No" +  
account number);

System.out.println ("Customer Name : " +  
CustomerName);

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

class SavAcct extends Account

public SavAcct (String CustomerName,  
long account, double balance),

public void computeAndDeposit  
Interest (double rate)

double interest = balance \* rate / 100;

balance += interest;

System.out.println ("Interest Computed &  
deposited updele");

public void withdraw (double amount)

## BANK ACCOUNT

import java.util.Scanner;

class Account {

{

String CustomerName; // no wapf

long AccountNumber;

String accountType;

double balance;

public Account (String CustomerName,  
String accountType, double balance)

this.CustomerName = CustomerName;

this.AccountNumber = AccountNumber;

this.accountType = accountType;

this.balance = balance;

public void deposit (double amount)

balance += amount;

System.out.println ("Deposit

Successful. Update Balance")

public void displayBalance ()

{

of  
rectangle rectangle = new Rectangle (5, 10),

triangle triangle = new Triangle (4, 5)

circle circle = new Circle (6);

rectangle.printArea();

triangle.printArea();

circle.printArea();

Output :

rectangle Area : 50

triangle Area : 10

circle Area : 113.0973

extends

class Triangle : Shape {

    public + single length, int breath;

    } Super (length, breath);

    public void printArea () {

        double area = 0.5 \* length \* breath;  
        System.out.println("triangle " + area);  
    }

}

class Circle extends Shape

    {

        public Circle (int radius) {

    }

        super (radius, 0);

}

    public void printArea ()

    {

        double area = Math.PI \* length \* length;  
    }

}

    public class Main {

        public static void main (Shape [])

3) Shapes

{ protected int length;

{ protected int breath;

{ public void printArea(); }

Abstract class Shape {

protected int length;

protected int breath;

public void printArea(); }

this.length = length;

this.breath = breath;

System.out.println("length = " + length + ", breath = " + breath);

public abstract void printArea(); }

class Rectangle extends Shape {

public Rectangle(int length,  
int breath) { }

super(length, breath); }

public void printArea() { }

The area = length \* breath;

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

g

```
author = s.nextLine();
System.out.println("Enter Price");
b[i] = new Books (name, author, price,
                  noOfPages);
for (int i=0; i<n; i++)
    System.out.print("Book " + (i+1) + ": ") b[i];

```

Output

No. of books : 2

Enter name of book : jungle

Enter author : Rud.

Price of book = 1500

Enter no. of pages = 500

Book 1

book name : jungle

Author : Rud.

Price : 1500

~~No. of pages = 500~~

returns name + author + price + num pages

Class Main

{

public static void main (String args [] )

{

Scanner s = new Scanner (System.in) ;

int n;

String name;

String author;

int price;

int numPages;

System.out.println ("No of books") ;

n = s.nextInt();

Book b [ ] ;

b = new Book [n] ;

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

System.out.println ("Book" + (i + 1) + ":") ;

System.out.println ("Name of book");

name = s.nextLine();

System.out.println ("Enter Author");

## 3) Book database

Import java.util.Scanner;

class Books

{

String name ;

String author ;

int price ;

int numpage ;

Books()

{ }

Books( String new , String author ,

int price , int numpage )

{

this.name = name ;

this.author = author ;

this.price = price ;

this.numpage = numpage ;

}

public String toString()

String name , author , price , numpage ;

name = "book name " + this.name + " \n " ;

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

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

price

s.Accept(1)

s.Display(1)

System.out.print("Enter marks")

94

Output

User NO. of students is 2

User USN : 1BM22CS084

User Name : Deva

User marks for 6 subjects

96

97

85

90

95

User USN : 1BMCE22

User name :

Student details

USN : 1BM13CS

Name : Deva

marks : 98, 95, 97, 85, 90, 95

percentage is 93.5%

credit[i] = & nextNode);

System.out.println("No of marks: " + (i+1) + ");  
Total marks[i] = & nextNode);  
g

void disp()

{

System.out.println("User i" + i);

System.out.println("Name " + name);

for (i=0; i < credit.length; i++)  
{

System.out.println("Subject " + (i+1) +  
credit[i]); g

{ double sgre = 0;

for (i=0; i < credit.length; i++)

sgre = credit[i] \* mark[i];  
g

sgre = credit.length;

return sgre;

g

public class Main {

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

{

Student s = new Student();

~~System.out.println("Roots are")~~

Student

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

```
class Student {
```

```
    String USN;
```

```
    String name;
```

```
    int credits[];
```

```
    int marks[];
```

```
    void accept()
```

```
{
```

```
    Scanner s = new Scanner(System.in);  
    System.out.println("Enter USN");
```

```
    USN = s.next();
```

```
    System.out.println("Enter the name");  
    name = s.next();
```

```
    System.out.println("Enter the marks")
```

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

```
        credits[i] = new int[n];
```

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

```
            System.out.println("No. of m + [i+1]);
```

Enter the coefficients of abc

1  
9

1

Roots are -0.1125 and 3.857

~~Roots are real & distinct.~~

```

        System.out.println ("roots are real")
        System.out.println (" roots1 = roots2 = " + r1);
        M = (-b) + (Math.sqrt(d)) / (double)(2*a);
        r2 = (-b) - (Math.sqrt(d)) / (double)(2*a);
        System.out.println ("roots are real & same distinct");
        System.out.println ("root1 = " + r1 + "root2" + r2);
    }

    System.out.println ("Roots are imaginary");
    M = (-b) / (2*a);
    System.out.println ("roots = " + r1 + " + " + r2);
}

```

Use quadratic Main

```

public static void main (String [] args)
{

```

```

    Quad retic q = new Quad retic ();

```

```

    q. input();

```

```

    q. compute();
}

```

Output the coefficients of a, b, c

Roots are not Real

## 1) Quadratic Equation code.

```

import java.util.Scanner;
class Quadratic
{
    int a,b,c;
    double r1, r2, d;
    void input()
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter ");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while(a==0)
        {
            System.out.println("No quad");
            System.out.println("Non zero value");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
        d = b*b - 4*a*c;
        if(d==0)
        {
            r1 = (-b)/(2*a);
        }
    }
}

```