

Quadratic.

```
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 a,b,c");  
        a=s.nextInt();  
        b=s.nextInt();  
        c=s.nextInt();  
    }  
  
    Void Compute()  
    {  
        while (Compute(a)==0)  
        {  
            System.out.println("Not a quadratic equation");  
            System.out.println("Enter a Non-zero value of 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)))/(2*a);  
            r2=((-b)-(Math.sqrt(d)))/(2*a);  
            System.out.println("Roots are real and distinct");  
            System.out.println("Root1=" +r1+" Root2=" +r2);  
        }  
    }  
}
```

{

else

{ System.out.println ("Roots are imaginary and real roots don't exist"); }

{

{

Class QuadraticEq

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

{ Quadratic q = new Quadratic(); }

q.getd();

q.compute();

{

{

Enter the values of a,b,c:

123

456

789

~~44-5666~~

the roots are real and exist.

41.5666

Sum

Class Overload S

Void print(int n) {

int sum = 0;

for (int i=1; i<=n; i++) {

sum = sum + i;

}

System.out.println("Sum of " + n + " natural numbers is " + sum);

}

Void print(int m, int n) {

System.out.println("prime numbers in range are");

for (int i=m; i<=n; i++) {

int flag=0;

for (int j=2; j<=i/2; j++) {

if (i%j==0) {

flag = 1;

break;

}

}

if (flag == 0)

System.out.println(i);

}

}

}

(7, 13)

Output: The sum prime

Numbers in range are

32

Class Overload Demo {

public static void main (String[] args) {

Overload o = new Overload (1);

o.print(5);

o.print(7, 13);

}

}

Date
22/12/23

12/08/21

SURYA Gold

Date _____
Page _____

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

Class Student {

```
    private String USN;
```

```
    private String name;
```

```
    private int[] marks = new int[6];
```

```
    public void acceptDetails() {
```

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

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

```
        this.USN = scanner.nextLine();
```

```
        System.out.println("Enter marks for 6 Subjects : ");
```

```
        for (int i=0; i<marks.length; i++) {
```

```
            System.out.print("Subject " + (i+1) + ": ");
```

```
            this.marks[i] = scanner.nextInt();
```

3
3.

B

```
    public double calculatePercentage() {
```

```
        int totalMarks = 0;
```

```
        for (int marks : marks) {
```

```
            totalMarks += marks;
```

3

```
        return (double) totalMarks / marks.length;
```

3

```
    public void displayDetails() {
```

```
        System.out.println("USN: " + this.USN);
```

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

```
        System.out.println("Marks: ");
```

```
        for (int i=0; i<marks.length; i++) {
```

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

3

System.out.println ("Percentage : " + calculatePercentage () + "%.");
 System.out.println ();

3

3

Public Class Main {

public static void main (String [] args) {

Scanner Scanner = new Scanner (System.in);

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

int numStudents = Scanner.nextInt ();

Student [] Student = new Student [numStudents];

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

Student [i] = new Student ();

System.out.println ("Enter details for student " + (i + 1) + ":");

Student [i]. acceptDetails ();

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

Output

Number of Students :

1

Subject 2 : 90

Subject 3 : 90

Enter details for Student 1:

Subject 4 : 90

Enter USN :

Subject 5 : 90

1

Subject 6 : 90

Enter name :

Raj

Enter marks of all subjects :

Subjects 1 : 90

import java.util.Scanner;

Class Book {

```
private String name;  
private String author;  
private double price;  
private int numPages;
```

public Book (String name, String author, double price, int numPages) {

```
this.name = name;  
this.author = author;  
this.Price = price;  
this.numPages = numPages;
```

}

public String getName () {

```
return name;
```

public String getAuthor () {

```
return author;
```

public double getPrice () {

```
return price;
```

}

public int getNumPages () {

```
return numPages;
```

,

public void SetAuthor (String author) {

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

public void setPrice (double price) {

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

public void setNumPages (int numPages) {

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

,

⑨ Override

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

```
    return "Book Details : \n Name : " + name + "\n Author : " +
           author + "\n Price : " + price + "\n Number of pages : " + numPages;
```

{

{

Public class Main {

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

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

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

```
int numBooks = scanner.nextInt();
```

```
Book [] book = new Book [numBooks];
```

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

```
System.out.println ("Enter details for book " + (i+1) + ":");
```

```
System.out.print ("Name : ");
```

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

```
System.out.print ("Price : ");
```

```
double price = scanner.nextDouble();
```

```
System.out.print ("Number of pages : ");
```

```
int numPages = scanner.nextInt();
```

```
book [i] = new Book (name, author, price, numPages);
```

{

```
System.out.println ("\n Book Details : \n");
```

Solve
1/1/24

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

```
System.out.println ("Details for book " + (i+1) + ":");
```

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

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

{

{

Output :

enter no of books : 1

enter book name : Raj

enter author name : Ruth

enter Price : 1000

enter no of pages : 2500

book name : Raj

book author : Ruth

" Price : 1000

no of pages : 2500

Lab 19/10/24 Area program.

import java.util.Scanner;

Abstract class Shape {

protected double dimension 1;

protected double dimension 2;

public Shape (double dimension1, double dimension2) {

this.dimension1 = dimension1;

this.dimension2 = dimension2;

}

public abstract void printArea();

}

Class Rectangle extends Shape {

public Rectangle (double length, double width) {

super(length, width);

}

② Override

public void printArea() {

double area = dimension1 * dimension2;

System.out.println ("Area of Rectangle: " + area);

}

Class Triangle extends Shape {

public Triangle (double base, double height) {

super(base, height);

}

③ Override

public void printArea() {

double area = 0.5 * (dimension1 * dimension2);

System.out.println ("Area of Triangle: " + area);

}

3

Class Circle extends Shape {
 public Circle (double radius) {
 Super (radius, 0);
 }

(a) Overide

```
public void printArea () {  

  double area = Math.PI * dimension1 * dimension1;  

  System.out.println ("Area of circle : " + area);  

}
```

Public class Main {

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

  Scanner scanner = new Scanner (System.in);  

  System.out.print ("Enter length of Rectangle : ");  

  double length = scanner.nextDouble();  

  System.out.print ("Enter width of rectangle : ");  

  double width = scanner.nextDouble();  

  Rectangle rectangle = new Rectangle (length, width);  

  rectangle.printArea();
```

```
System.out.print ("Enter radius of circle : ");  

double radius = scanner.nextDouble();  

Circle circle = new Circle (radius);  

circle.printArea();
```

{}

Output: Enter length of Rectangle : 10

Enter width of Rectangle : 10

Area of Rectangle : 100 unit²

Enter base of Triangle : 20

Enter height of Triangle : 50

Area of Triangle : 500 unit²

Enter radius of circle : 15

Area of circle : 706.8583

unit²

Lab 19/01/24 Bank program

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

```
Class Account {
```

```
    String CustomerName;
```

```
    long accountNumber;
```

```
    String accountType;
```

```
    double balance;
```

```
    Public 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 += amount;
```

```
    System.out.println ("Deposit successful. Updated balance: " +  
        balance);
```

```
}
```

```
Void displayBalance () {
```

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

```
}
```

```
}
```

```
Class Current extends Account {
```

```
    double minBalance;
```

```
    double serviceCharge;
```

```
    Public Current (String CustomerName, long accountNumber,  
        double balance) {
```

```
        Super (CustomerName, accountNumber, "Current Account",  
            balance);
```

this. minBalance = 1000;

this. ServiceCharge = 50;

{}

② Overwrite

Void deposit (double amount) {

Super. deposit (amount);

{}

③ Override

Void displayBalance () {

Super. displayBalance ();

{}

Void withdraw (double amount) {

if (balance - amount >= minBalance) {

balance -= amount;

System.out.println ("Withdrawal");

Successful. updated balance : \$" + balance)

{}

else {

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

Minimum balance requirement reached");

{}

{}

~~Void imposeServiceCharge () {~~

~~if (balance < minBalance) {~~

~~balance -= ServiceCharge;~~

~~System.out.println ("Service charge imposed. Updated balance : \$" +~~

~~balance);~~

{}

{}

{}

Class SavAcct extends Account {
 double interestRate;

public SavAcct (String customerName, long accountNumber,
 double balance, double interestRate)
 {

 Super (customerName, accountNumber, "Savings Account",
 balance);

 this.interestRate = interestRate;

}

② Overide

void deposit (double amount) {
 Super.deposit (amount);

}

③ Overide

void displayBalance () {

 Super.displayBalance ();

}

void computeAndDepositInterest () {

 double interest = balance * interestRate / 100;

 balance += interest;

 System.out.println ("Interest deposited");

 updated balance : \$" + balance);

}

void withdraw (double amount) {

 if (balance - amount >= 0) {

 balance -= amount;

 System.out.println ("withdraw successful");

 updated balance : \$" + balance);

}

 else {

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

 }

Public class BankApp {

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

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

```
    System.out.println ("Creating current Account...");
```

```
    CurrentAcct CurrentAccount = new CurrentAcct ("John Doe",  
        123456789, 1500);
```

```
    CurrentAccount.displayBalance();
```

```
    System.out.print ("Enter deposit amount for current Account:  
        ");
```

```
    double currentDeposit = scanner.nextDouble();
```

```
    CurrentAccount.withdraw (currentDeposit);
```

```
    System.out.println ("Enter withdraw amount for Current  
        Account: $");
```

```
    double currentWithdrawal = scanner.nextDouble();
```

```
    CurrentAccount.withdraw (currentWithdrawal);
```

```
    CurrentAccount.imposeServiceCharge();
```

```
    CurrentAccount.displayBalance();
```

```
    System.out.println ("Creating Savings account...");
```

```
    SavingsAccount savingsAccount = new SavingsAccount ();
```

```
    System.out.print ("Enter deposit amount for savings  
        Account: $");
```

```
    double savingsDeposit = scanner.nextDouble();
```

```
    SavingsAccount.deposit (savingsDeposit);
```

```
    SavingsAccount.displayBalance();
```

```
    System.out.print ("Enter withdraw amount for  
        savings account: $");
```

```
    double savingsWithdrawal = scanner.nextDouble();
```

```
    SavingsAccount.withdraw (savingsWithdrawal);
```

```
    SavingsAccount.computeAndDepositInterest();
```

```
    SavingsAccount.displayBalance();
```

}

{

Output:

Deposit of 500.0 Successful. updated balance 1500.0

Account Balance: 1500

Compound interest deposited: 75.0

Updated -balance: 1575.0

Withdrawal of 200.0 Successful. updated balance 3000.0

Account Balance: 3000.0

Withdrawal of 1500.0 Successful

Updated balance: 1500.0

~~Testing~~

Creating current Account...

Account Balance: 1500.0

Enter deposit amount for current Account: 500

Deposit Successful. update balance = 2000.0

Account Balance: 2000.0

Enter withdrawal amount for current Account: 500

Withdrawal Successful. updated balance: 1500.0

Account Balance: 1500.0

*Save
19/11/2018*

Lab-160212n

Package program.

1. CIE package

package CIE;

public class personnel {

 public String USN;

 public String name;

 public String info Sem;

 public personnel (String USN, String name, int Sem) {

 this.USN = USN;

 this.name = name;

 this.Sem = Sem;

 }

1500.0
import java.util.Arrays;
public class Internal {
 public int[] internalMarks;
 public Internal (int[] internalMarks) {
 this.internalMarks = internalMarks;
 }
}

3000.0
3
Package SEE;
import CIE.personal;
public class External extends personal {
 public int[] seeMarks;
 public External (String usn, String name, int sem, int[] seller) {
 super (usn, name, sem);
 this.seeMarks = seeMarks;
 }
}

3
Package ForMain;
import java.util.Arrays;
import CIE.Internal;
import CIE.Personal;
import SEE.External;
Public class Main {
 Public static void main (String[] args) {
 int n = 3;
 Student[] students = new Student[n];
 for (int i = 0; i < n; i++) {
 int[] internalMarks = {80, 75, 80, 85, 75};
 int[] seeMarks = {70, 80, 75, 90, 85};
 students[i] = new Student (new personal ("USN" + i,
 "Student" + i), new Internal (internalMarks));
 students[i].see = new External ("USN" + i, "Student" + i,
 seeMarks);
 }
 for (int i = 0; i < students.length; i++) {
 Student student = students[i];
 }
 }
}

Lab-16 OOPs Exception program

import java.util.Scanner;

Class WrongException extends Exception
{

 public WrongException (String message)
 {

 super (message);
 }

}

Class Father

{

 private int fatherAge ;

 public Father (int age) throws WrongException
 {

 if (age < 0) {

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

 this.fatherAge = age ;

}

Class Son extends Father

{

 private int sonAge ;

 public Son (int fatherAge, int sonAge) throws WrongException
 {

 super (fatherAge);

 if (sonAge >= fatherAge)

{

 throw new WrongException ("Son's age should be less
 than Father's age");

}

 this.sonAge = sonAge ;

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

 System.out.println ("Son's Age : " + sonAge);

}

 public class ExceptionInheritanceDemo

{

 public static void main (String [] args) {

STUDENT'S NAME

CLASS

SUBJECT

ROLL NO.

DATE

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

```
try
```

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

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

```
Father father = new Father (fatherAge);
```

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

```
int SonAge = Scanner.nextInt();
```

```
Son Son = new Son (fatherAge, SonAge);
```

```
3
```

```
Catch (WrongException e)
```

```
3
```

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

```
3
```

```
3
```

Output:

Enter Father's Age : 50

Enter Son's Age : 25

Father's Age : 50

Son's Age : 25

Enter Father's Age : 30

Enter Son's Age : 50

Exception : Son's age should be less than Father's age

Enter Father's Age : -30

Exception : Age cannot be negative.

STUDENT'S NAME

CLASS

SUBJECT

ROLL NO.

DATE

Sop ("Student:" + Student.personal_name);

Sop ("Internal marks:" + Arrays.tostring (Student.internals));

Sop ("SEE Marks:" + Arrays.tostring (Student.SEE.Gmarks));

Sop (

³
3

Static class Student {

 public personal personal;

 public Internals internals;

 public external SEE;

 public Student (personal personal, internals internals) {

 this.personal = personal;

 this.internals = internals;

³
³
3

Object :

Student : Student 0

Internal marks : [80, 75, 90, 85, 95]

SEE marks : [70, 80, 75, 90, 85]

Student : Student 1

Internal marks : [80, 75, 90, 85, 95]

SEE marks : [70, 80, 75, 90, 85]

Student : Student 2.

Internal marks : [80, 75, 90, 85, 95]

SEE marks : [70, 80, 75, 90, 85].

Lab-16(02)24 Thread Program

Class NewThread1 implements Runnable

Thread t1;

NewThread1()

{

t1 = New Thread (this, name : "Thread 1");

System.out.println("T: " + t1);

t1.start();

}

public void run()

{

try {

for (int n=5; n>0; n--) {

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

Thread.sleep(millis: 1000);

}

Catch (InterruptedException ie.)

{

System.out.println("X: " + "Thread 1 Interrupted");

}

System.out.println("X: " + "Thread 1 quitting");

}

Class NewThread2 implements Runnable

Thread t2;

NewThread2()

{

t2 = New Thread (this, name : "Thread 2");

System.out.println("T: " + t2);

t2.start();

}

public void run()

{

try {

for (int n=5; n>0; n--) {

System.out.println("X: " + "CSE ");

Thread.sleep(millis: 2000);

}

Catch (InterruptedException exception ie.)

{

M

System.out.println(x: "Thread 2 Interrupted");

}

~

System.out.println(x: "Thread 2 quitting");

}

3

Class MainThread

{

public static void main (String ssm)

{

new Thread1();

new Thread2();

try

{

Thread.sleep (mills: 60000);

System.out.println(x: "main Thread is awake \n");

}

Catch (InterruptedException e)

{

System.out.println(x: "main Thread Interrupted");

}

~

System.out.println(x: "main thread exiting");

}

3

Surekha
7/6/2024

O/p: CT: Thread [#29, Thread 1, 5, main]

BMS College of Engineering

CT: Thread [#30, Thread 2, 5, main]

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

Thread 2 quitting

BMS College of Engineering

Thread 2 quitting

BMS College of Engineering

BMS College of Engineering

Main Thread is awake

Main Thread exiting

BMS College of Engineering

Thread 1 quitting

SURYA GOLD

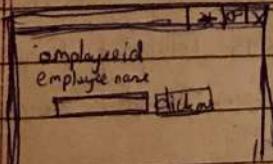
Lab 23/02/23 LABEL, BUTTON AND TEXTFIELD - AWT

```
import java.awt.*;  
import java.awt.event.*;  
public class awt extends WindowAdapter {  
    Frame f;  
    awt() {  
        f = new Frame();  
        f.addWindowListener(this);  
        Label l1 = new Label("Employee Id:");  
        Button b = new Button("Submit");  
        TextField t = new TextField();  
        l1.setBounds(71, 20, y: 20, width: 80, height: 30);  
        b.setBounds(71, 20, y: 100, width: 80, height: 30);  
        b.setBounds(71, 100, width: 80, height: 30);  
        t.setEditable(true);  
        t.setBounds(71, 100, width: 80, height: 30);  
        f.setSize(600, height: 200);  
        f.setTitle("Employee Info");  
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        f.setVisible(true);  
    }  
}
```

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

```
public static void main(String[] args) {  
    awt awtObj = new awt();  
}
```

O/P:



Lab 23/02/23 BUTTON AND ACTION LISTENER FOR MOUSE CLICK

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

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

Class EventHandling extends windowAdapter implements ActionListener {

```
Frame f;
```

```
TextField tf;
```

```
EventHandling() {
```

```
f = new Frame("N");
```

```
f.addWindowListener(this);
```

```
tf = new TextField("1");
```

```
tf.setBounds(70, 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 windowClosing(WindowEvent e) {
```

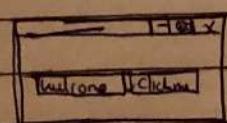
```
System.exit(0);
```

}

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

```
EventHandling app = new EventHandling();
```

}



Lab 23/08/24

I/O Programs

1)

```
import java.io.*;
public class ByteArrayInput {
    public static void main (String [] args) throws IOException {
        byte [] buf = { 35, 36, 37, 38 };
        ByteArrayInputStream bout = new ByteArrayInputStream (buf);
        int k = 0;
        while ((k = bout.read ()) != -1) {
            char ch = (char) k;
            System.out.println ("ASCII Value of character is :" + ch +
                " Special character b :" + ch);
        }
    }
}
```

O/p: ASCII value of character is 35

Special character is :

ASCII Value of character is 36

Special character is :

ASCII Value of character is 37

Special character is :

ASCII Value of character is 38

Special character is :

2)

```
import java.io.*;
public class ByteArrayEx {
    public static void main (String [] args) throws Exception {
        FileOutputStream bout1 = new FileOutputStream ("Example1.txt");
        FileOutputStream bout2 = new FileOutputStream ("Example2.txt");
        ByteArrayOutputStream bout = new ByteArrayOutputStream ();
        bout.write (5);
        bout.writeTo (fout1);
        bout.writeTo (fout2);
        fout1.close ();
        fout2.close ();
    }
}
```

System.out.println ('Success');

O/P: Success

CLASS	SUBJECT
ROLL NO.	DATE

3) `import java.io.*;`

`public class FileEx1 {`

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

`FileInputStream fin = new FileInputStream ("example.txt");`

`System.out.println ("Remaining bytes that can be read : " + fin.available());`

`int content;`

`while ((content = fin.read ()) != -1) {`

`System.out.print (char) (content + ' '));`

`}`

`System.out.println ("\nRemaining bytes that can be read : " + fin.available());`

`fin.close ();`

`+ fin.available ()`

`}`

O/p: Remaining bytes that can be read : 11

HELLO WORLD!

Remaining bytes that can be read : 0

4) `import java.io.FileInputStream;`

`import java.io.IOException;`

`public class FileEx2 {`

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

`FileInputStream fin = new FileInputStream ("example.txt");`

`byte [] bytes = new byte [20];`

`int i;`

`i = fin.read (bytes);`

`System.out.println ("Number of bytes read : " + i);`

`System.out.print ("Bytes read : ");`

`for (byte b : bytes) {`

`char c = (char) b;`

`System.out.print (c);`

`}`

`fin.close ();`

`}`

Note
23/2/24

O/p: Number of bytes read : 13

Bytes read : HelloWorld