

Quadratic equation Program-1

```
import java.util.Scanner;  
class QuadraticEquation {  
    double a;  
    double b;  
    double c;  
    public QuadraticEquation(double a, double b, double c){  
        this.a=a;  
        this.b=b;  
        this.c=c;  
    }  
    public double calculateDiscriminant(){  
        return b*b - 4*a*c;  
    }  
    public void calculateRoots(){  
        double discriminant = calculateDiscriminant();  
        if(discriminant > 0){  
            double root1 = (-b + Math.sqrt(discriminant))/(2*a);  
            double root2 = (-b - Math.sqrt(discriminant))/(2*a);  
            System.out.println("Root 1: " + root1);  
            System.out.println("Root 2: " + root2);  
        } else if(discriminant == 0){  
            double root = -b/(2*a);  
            System.out.println("Root: " + root);  
        } else {  
            System.out.println("No real roots. Complex roots exist.");  
        }  
    }  
}
```

```
class QuadraticEquationSolver {  
    public static void main(String[] args){  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the coefficients of the quadratic  
equation (a,b,c): ");  
        double a = scanner.nextDouble();  
        double b = scanner.nextDouble();  
        double c = scanner.nextDouble();  
    }  
}
```

```
double c = scanner.nextDouble();
QuadraticEquation quadraticEquation = new QuadraticEquation
    (a, b, c);
quadraticEquation.calculateRoots();
scanner.close();
```

83

dp Enter the coefficients of the quadratic equation (a, b, c):

2.0 5.0 2.0

Root1:-0.5

Root2:-2.0

Program-2

Create a class Book that contain name, Author, price and num pages. Include a constructor to set values. Include a testing method that could display complete details of Book.

BookBank

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

```
class Book
```

```
String Name;
```

```
String Author;
```

```
float price;
```

```
int numPages;
```

```
public Book()
```

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

```
System.out.print("Enter Name of Book:");
```

```
this.Name = s.nextLine();
```

```
System.out.print("Enter Name of Book:");
```

```
this.Name = s.nextLine();
```

```
System.out.print("Enter Author of Book:");
```

```
this.Author = s.nextLine();
```

```
System.out.print("Enter Number of Pages in Book:");
```

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

8

```
void Setdetails()
Scanner S= new Scanner (System.in);
System.out.print ("Enter new name of Book");
this.Name = S.nextLine();
System.out.print ("Enter new Author of Book");
this.Author = S.nextLine();
System.out.print ("Enter new number of Pages in Book");
this.numPages = S.nextInt(); }
```

```
void get details()
```

```
Scanner S= new Scanner (System.in);
int choice = S.nextInt();
System.out.print ("Enter for Name of Book In 1
for Author of Book In 2 for Price of Book In 3
Number of Pages");
switch (choice){
```

```
Case 1: System.out.println ("Name of Book is: "+this.Name);
```

```
Case 2: System.out.println ("Name of Author is: "+this.Author);
break;
```

```
Case 3: System.out.println ("Price of Book "+this.Price);
```

```
Case 4: System.out.println ("Number of Pages "+this.numPages);
```

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

```
return "Name : "+this.Name + " Author : "+this.Author
+ "Price : "+ this.Price + " No of Pages : "+this.numPages; }
```

```
class BookBank {
```

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

```
System.out.println ("Enter Number of Books");
```

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

```
int n = S.nextInt();
```

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

```
for (int i=0; i<n; i++) {  
    book[i] = new Book();  
    System.out.print(book[i].toString());  
}
```

33

Q18

Enter number of books:

2

Enter Name of Book: Hizelrthon

Enter Author of Book: Dinesh Kumar G

Enter Number of Pages in Book:

226

Name: Hizelrthon Author: Dinesh Kumar G

Price 200 No of Pages 226

Enter Name of Book: Zeadu

Enter Author of Book: Kakashi

Enter Number of Pages in Book:

221

Name: Zeadu Author: Kakashi Price 200

No of Pages 221

Program-3

Write a Java program to create class Student Contain USN, Name, marks (6 subjects). Include method to accept detail ~~also~~ include method to calculate percentage.

import java.util.Scanner;

class Main{

String Name;

String USN;

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

public Main(){

Scanner s = new Scanner(System.in);

System.out.print("Enter Name of Student: ");

this.Name = s.nextLine();

```

System.out.print("Enter USN of Student");
this.USN = s.nextLine();
System.out.println("Enter Marks of Student");
for (int i=0; i<6; i++) {
    System.out.print("Enter marks for Subject" +
        (i+1) + ": ");
    this.marks[i] = s.nextInt();
}

```

void percentage()

```

double percentage = 0;
for (int i=0; i<6; i++) {
    percentage += this.marks[i];
}

```

System.out.println("Total percentage is : " + (100 * percentage))

```

public static void main(String args[]) {
    System.out.println("Enter number of Students:");
    Scanner s = new Scanner(System.in);
    int n = s.nextInt();
    Main std[] = new Main[n];
    for (int i=0; i<n; i++) {
        std[i] = new Main();
        std[i].percentage();
    }
}

```

333

O/P

Enter number of Students: 1

Enter name of Student: Disha Kumar

Enter USN of Student: 1Bm22CS091

Enter Mark of Student: 40 out of 50

Enter mark for Subject 1: 40

Enter mark for Subject 2: 40

Enter mark for Subject 3: 40

Enter mark for Subject 4: 30

Enter mark for Subject 5: 20

Enter mark for Subject 6: 10

Total Percentage is: 60

Program - h

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 the method printArea() that prints the area of the given shape.

abstract class Shape {

```
int l=1;  
int b=3;
```

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

}

class Rectangle extends Shape {

```
void printArea(){
```

```
System.out.println("Area of rectangle is :" +(l*b));
```

33

class Triangle extends Shape {

```
void printArea(){
```

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

$+ (l * b) / 2;$

33

class Circle extends Shape {

```
void printArea(){
```

```
System.out.println("Area of Circle is : " +
```

$(3.14 * l * l);$

33

public class Run

public static void main (String [] args) {
Shape s1 = new Rectangle();

s1.printArea();

Shape t = new Triangle();

t.printArea();

Shape c = new Circle();

c.printArea();

33

Output area of rectangle is: 6

area of Triangle is: 3

Area of Circle is: 12.56

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

Program - 5

```
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; }
```

```
    public void deposit (double amount) {
```

```
        balance += amount;
```

```
        System.out.println ("Deposit of $" + amount + " Successful."); }
```

```
    public void displayBalance () {
```

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

```
    public void withdraw (double amount) {
```

```
        if (balance >= amount) {
```

```
            balance -= amount;
```

```
            System.out.println ("Withdrawal of $" + amount +  
                " Successful."); }
```

```
        else {
```

```
            System.out.println ("Insufficient funds. Withdrawal  
                failed."); }
```

```
    class SavAcct extends Account {
```

```
        double interestRate;
```

```
        public SavAcct (String customerName, long accountNumber,
```

```
            double balance, double interestRate) {
```

Super (CustomerName, accountNumber, "savings", balance);
this. interestRate = interestRate; }

public void ComputeInterest() {

double interest = balance * interestRate / 100;

deposit (interest);

System.out.println ("Interest of \$" + interest + " Computed
and deposited. "); }

public void withdraw (double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println ("Withdrawal of \$" + amount +
" successful."); }

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

withdraw failed.); } }

class Current extends Account {

double minBalance;

double serviceCharge;

public Current (String customerName, long accountNumber,
double balance, double minBalance, double serviceCharge) {

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

this. minBalance = minBalance;

this. serviceCharge = serviceCharge; }

public void withdraw (double amount) {

if (balance - amount >= minBalance) {

super.withdraw (amount); }

else {

System.out.println ("Withdrawal failed.. Minimum

balance not maintained. Service charge of \$" +
serviceCharge + " imposed.");

balance -= serviceCharge; }

public class Bank{

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

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

SavAcct Savings Account now SavAcct "John" 123L56789,
(200.0, 50);

Savings Account . display Balance () ;

Saving Account. deposit (500.0)

Savings Account. Compute Interest ();

Savings Account (Withdrew (200.0))

SavingsAccount displayBalance();

(ConAcct Current Account = New ConAcct ("Jinn", 876543)

1500.0, 1000.0, 20.0);

(current Account. display Balance());

CURRENT ACCOUNT, DEPOSIT (300.0);

(Current Account. Withdraw C700.0)

(Current Account. display Balance());

```
scanner.Close(); } }
```

S/P Account Balance: \$1000.0

Deposit of \$500. Q successful.

Deposit of \$75. O successful.

Interest of \$75.0 computed and deposited.

Withdrawal of \$200.0 successful.

Account Balance : \$1375.0

~~Account Balance~~: \$1500.0

~~Deposit of \$300.00 successful.~~

Withdrawal of \$700.0 successful.

Account Balance: \$1100.0

~~860~~
221.124

CIE

Internals.java

package CIE;

public class Internals extends Student {

 public int[] internalMarks = new int[5];

}

Program-6

Student.java

package CIE;

public class Student {

 public String usn, name;

 public int sem;

}

Internals.java

package CIE;

public class Student {

 public String usn,

 public class Internals extends Student {

 public int[] internalMarks = new int[5];

}

SEE

package SEE;

import CIE.Student;

public class External extends Student {

 public int[] seeMarks = new int[5];

}

```

import CIE.Internal;
import SEL.External;
import java.util.Scanner;

public class Main {
    public static void main (String [] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.print ("Enter the number of students: ");
        int n = scanner.nextInt();

        Internal [] cieStudents = new Internal [n];
        External [] selStudents = new External [n];

        for (int i=0; i<n; i++) {
            CIE.Student [i] = new Internal ();
            System.out.println ("Enter details for CIE of Student " +
                +(i+1)+ ":");

            System.out.print ("USN: ");
            cieStudents [i].usn = scanner.nextInt();
            System.out.print ("Name: ");
            cieStudents [i].name = scanner.next();

            System.out.print ("Semester: ");
            cieStudents [i].sem = scanner.nextInt();

            System.out.println ("Enter Internal marks for 5 Courses: ");
            for (int j=0; j<5; j++) {
                System.out.print ("Course " + (j+1) + ": ");
                CIE.Student [i].internalMarks [j] = scanner.nextInt();
            }
        }
    }
}

```

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

See Students[i].new External();

System.out.println ("Enter details for SEE of Student");
(i+1) + ":");

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

See Students[i].usn = scanner.nextInt();

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

See Students[i].name = scanner.nextLine();

System.out.print ("Semester: ");

See Students[i].sem = scanner.nextInt();

System.out.println ("Enter External marks for 5 Courses");

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

System.out.print("Course " + (j+1) + ":");

See Students[i].SeeMarks[j] = scanner.nextInt();

??

System.out.println ("Final Marks of Students: ");

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

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

System.out.println ("USN: " + cieStudents[i].usn);

System.out.println ("Name: " + cieStudents[i].name);

System.out.println ("Semester: " + cieStudents[i].sem);

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

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

System.out.println ("Course " + (j+1) + ":" +

cieStudents[i].internalmarks[j]); }

```
System.out.println("SEE Marks:");  
for (int j=0; j<5; j++) {
```

```
    System.out.println ("Grade " + (j+1) + ":" +  
    SeeStudents[i].seeMarks[j]);
```

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

```
}
```

```
Scanner.close();
```

```
}}
```

dp Enter the number of Students :)

Enter details for Cls and SEE of Student 1:-

USN: 2023 BMS 02519

Name: ~~Hemant~~ Dinesh

Semester: 3

Enter Internal Marks for Cls 5 Courses:

Course 1: 99

Course 2: 98

Course 3: 95

Course 4: 86

Course 5: 100

Enter External Marks for SEE 5 Courses:

Course 1: 99

Course 2: 90

Course 3: 95

Course 4: 100

Course 5: 100

Final marks of Students:

Student 1:

USN: 2023 BMS 02519

Name: Dinesh

Semester : 3

CIE marks:

Course 1 : 99

Course 2 : 98

Course 3 : 95

Course 4 : 86

Course 5 : 100

SEE Marks:

Course 1 : 99

Course 2 : 96

Course 3 : 95

Course 4 : 100

Courses : 100

✓

BB
29/11/2022

Exception handling Program-7

```
import java.util.Scanner;  
  
class Father {  
    int age;  
    Father(int age) {  
        this.age = age;  
    }  
    void check() throws ArithmeticException {  
        if (age < 0) {  
            throw new ArithmeticException ("Invalid age! age is less than  
0");  
        }  
        else {  
            System.out.println ("Father's age accepted");  
        }  
    }  
}  
  
class Child extends Father {  
    int childage;  
    Child (int fatherAge, int childage) {  
        super (fatherAge);  
        this.childage = childage;  
    }  
    void validate() throws ArithmeticException {  
        if (childage >= age) {  
            throw new ArithmeticException ("Invalid age! Child's age is  
greater than or equal to Father's age");  
        }  
        else {  
            System.out.println ("Child's age accepted");  
        }  
    }  
}
```

33

class Run {

public static void main (String args)

Scanner scanner = new Scanner (System.in);

System.out.println ("Enter father's age: ");

int fatherAge = scanner.nextInt();

Father father = new Father (fatherAge);

try {

father.check();

catch (ArithmaticException e) {

System.out.println (e);

return;

}

System.out.println ("Enter child's age: ");

int childAge = scanner.nextInt();

Child child = new Child (fatherAge, childAge);

try {

child.validate();

catch (ArithmaticException e) {

System.out.println (e);

333

df Enter father's age:

40

Father's age accepted

Enter child's age:

21

Child's age accepted

Multi-Threading Program 8

```
class Thread 1 extends Thread
{
    public void run()
    {
        for (int i=0; i<5; i++)
        {
            System.out.println("BMS College of Engineering");
            try
            {
                Thread.sleep(1000);
            }
            catch (InterruptedException e)
            {
                e.printStackTrace();
            }
        }
    }
}
```

```
class Thread 2 extends Thread
{
    public void run()
    {
        for (int i=0; i<25; i++)
        {
            System.out.println("CSE");
            try
            {
                Thread.sleep(2000);
            }
            catch (InterruptedException e)
            {
                System.out.println("");
                e.printStackTrace();
            }
        }
    }
}
```

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

Thread 1. t1 = new Thread1();

Thread 2 t2 = new Thread2();

t1.start();

t2.start();

83

Q:- BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

DPP
19/2/24

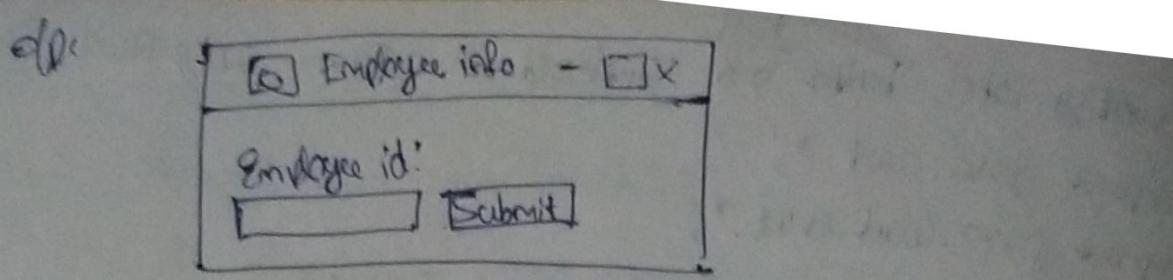
1. Creating label, button and TextField in a frame using AWT.

```
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);
        t.setBounds(20, 100, 80, 30);
        b.setBounds(100, 100, 80, 30);
        f.add(b);
        f.add(l);
        f.add(t);
        f.setSize(100, 300);
        f.setTitle("Employee info");
        f.setLayout(null);
        f.setVisible(true);
    }
}
```

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

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



2. Create a button and add a ActionListener for mouse click.

import java.awt.*;

import java.awt.event.*;

public class EventHandling extends WindowAdapter implements ActionListener

{

Frame f;

TextField tf;

EventHandling()

{}

// Create Components

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);

// register listener

b.addActionListener(this);

f.add(b);

f.add(tf);

f.setSize(300, 300);

f.setVisible(true);

f.setLayout(null);

}

```
public void actionPerformed(ActionEvent e) {
```

```
    f.f.setText("welcome");
```

```
}
```

```
public void windowClosing(WindowEvent e)
```

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

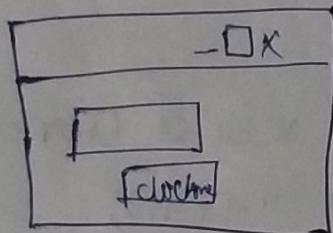
```
}
```

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

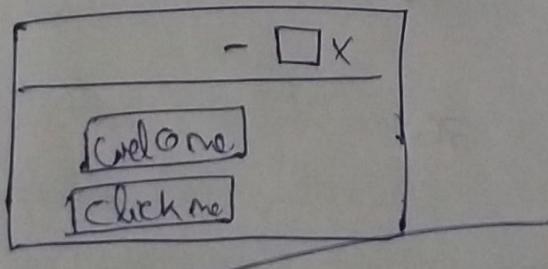
```
{ new EventHandling();
```

```
};
```

dp:



After clicking on "click me"



Programs on IO

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

Output:
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. Public class FileEx {

```
public static void main (String [] args) throws IOException {
    FileInputStream fin = new FileInputStream ("Example.txt");
    int Content;
    System.out.println ("Remaining bytes that can be read : "
        + fin.available ());
    Content = fin.read ();
    System.out.print ((char) Content + " ");
    System.out.print (Content + " ");
}
```

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

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

Q3

Q3: Remaining bytes that can be read : 11

H

Remaining bytes that can be read : 10

Remaining bytes that can be read : 10

Example at xt

Hello World

3- import java.io.FileInputStream;
import java.io.IOException;

Example at xt
hello world

public class FileEx2 {

 public static void main (String args[]) throws IOException {
 FileInputStream fin = new FileInputStream ("Example.txt");

 byte [] bytes = new byte [20];

 int i;

 char c;

 i = fin.read (bytes);

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

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

 for (byte b : bytes)

 c = (char) b;

 System.out.print (c);

23

System.out.println();

o/p: Number of bytes read: 11

Bytes read : hello world

~~100~~
 $2^{13}/2^7$