

# Lab Program 1

## Quadratic Equation:

```
import java.util.Scanner;  
class Quadratic {  
    double a, b, c, d, r1, r2;  
    void calculate() {  
        d = b * b - 4 * a * c;  
        if (d > 0)  
        {  
            System.out.println("The roots are real and distinct");  
            r1 = (-b + Math.sqrt(d)) / (2 * a);  
            r2 = (-b - Math.sqrt(d)) / (2 * a);  
            System.out.println("root1 = " + r1 + " and root2 = " + r2);  
        }  
        else if (d == 0)  
        {  
            System.out.println("The roots are real and equal");  
            r1 = -b / (2 * a);  
            System.out.println("root1 = root2 = " + r1);  
        }  
        else  
        {  
            System.out.println("The roots are imaginary");  
        }  
    }  
}
```

```
class main {
```

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

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

```
        Quadratic obj = new Quadratic();
```

```
        System.out.println("enter value of a");
```

```
        obj.a = s.nextDouble();
```

```
        System.out.println("enter value of b");
```

```
        obj.b = s.nextDouble();
```

```
sout (" enter value of c");
```

```
obj.c = S.nextDouble();
```

```
obj.calculate();
```

```
}
```

```
}
```

output:-

enter the value of a:

1

enter the value of b:

4

enter the value of c:

4

the roots are real and equal

$\sqrt{0.1} = \sqrt{0.2} = 2.0$

## Lab Program - 2

08/01/2024

```
import java.util.Scanner;  
class Student {  
    String USN, name;  
    int marks[] = new int[6];  
    double percentage;  
    Scanner s = new Scanner(System.in);  
  
    void calc() {  
        double total = 0;  
        for (int i=0; i<marks.length; i++) {  
            System.out.print("Enter subject " +  
                (i+1) + " marks");  
            marks[i] = s.nextInt();  
            if (marks[i] > 100)  
                System.out.println("Enter valid  
                    marks!");  
        }  
        this.calc();  
    }  
}
```

```
void setDetails() {  
    System.out.print("USN:");  
    USN = s.next();  
    System.out.print("Name:");  
    name = s.next();  
    this.setMarks();  
}
```

```
void Display () {  
    System.out.println ("");  
    System.out.println (" USN: " + USN +  
        " Name: " + name);  
    for (int i=0; i<marks.length; i++)  
        System.out.print ("Subject " + (i+1) + ":" +  
            marks[i]);  
    System.out.println (" Percentage: " + percentage);  
}
```

```
public class Lab1 {  
    public static void main (String args []) {  
        Student s [] = new Student [5];  
        for (int i=0; i<s.length; i++)  
            s[i] = new Student ();  
        System.out.println (" Enter student " +  
            (i+1) + " details:- ");  
        s[i].setDetails ();  
    }  
}
```

```
for (int i=0; i<s.length; i++)  
{  
    s[i].Display ();  
}  
}
```

Output:-

Enter student 1 details :- USN

1BM22CS069

Name : BHUPENDRA

BHUPENDRA

Enter subject 1 marks : 12

1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5 6 7 8 9 10 11 12

Enter student 2 details - USN :

1BM22CS209

Name :

AUVURTH

Enter subject 1 marks : 13

1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5 6 7 8 9 10 11 12

Percentage :- 13.03

Percentage :- 11.0

## Lab Program 3

```
import java.util.Scanner;  
class Books {  
    String name;  
    String author;  
    int price;  
    int numPages;  
    Books (String name, String author, int price,  
           int numPages) {  
        this.name = name; this.author = author;  
        this.price = price; this.numPages = numPages;  
    }  
    public String toString () {  
        return "Book Name" + this.name + "\n" +  
               "Author Name" + this.author + "\n" +  
               "Book Price : " + this.price + "\n" +  
               "Number of Pages" + this.numPages + "\n";  
    }  
}  
class Main2 {  
    public static void main (String [] args) {  
        System.out ("Bhupendra Singh");  
        System.out ("IBM22CS069");  
        Scanner s = new Scanner (System.in);  
        int n;  
        String name;  
        String author;  
        int price;  
        int numPages;  
        System.out ("Enter the number of Books:-");  
    }  
}
```

```

n = s.nextInt();
s.nextLine();
books[] b;
b = new Books[n];
for (int i=0; i<n; i++) {
    System.out.println("Book " + (i+1) + ":");
    System.out.print("Enter the book name:");
    name = s.nextLine();
    System.out.print("Enter the author:");
    author = s.nextLine();
    System.out.print("Enter the Price:");
    price = s.nextDouble();
    System.out.print("Enter the number of Pages:");
    numPages = s.nextInt();
    s.nextLine();
    b[i] = new Books(name, author, price, numPages);
}
System.out.println("Book " + (i+1) + " is " + b[i]);
}

```

## Output :-

Enter the number of books

3

Book 1

Enter the book name

DS is c

Enter the author

Reema Theriaq

Enter the pris.

Open Book - 28

Properties of

Enter the number of pages

600

Book 2:

Enter the book name

Gulliver's Travels

Enter the author name

Jonathan Swift

Enter the price

300

Enter the number of pages

1000

Book 1:

Book name: Oscar

Author name: Roena Thengi

Book Price: 200

Number of Pages: 600

Book 2

Book name is Gulliver's Travels

Author name: Jonathan Swift

Book Price: 300

Number of Pages: 1000

Lab Program 4

abstract class Shape {  
  // Inheritance with method

  protected int dimension1; // 000

  protected int dimension2; // 000

}

  // Using abstract method

  // Inheritance with method

public shape (int dimension1, int dimension2)

{  
  // Inheritance with method

  this.dimension1 = dimension1; // 000

  this.dimension2 = dimension2; // 000

}

  // Using abstract method

  // Inheritance with method

abstract void printArea(); // 1. defn.

}

Class Rectangle extends Shape {  
  // Inheritance with method

public Rectangle (int length, int width)

{  
  // Inheritance with method

  super (length, width); // 000

  // Inheritance with method

  // Inheritance with method

class Triangle extends Shape {

public Triangle (int base, int height)

{  
  // Inheritance with method

  super (base, height); // 000

  // Inheritance with method

  // Inheritance with method

  // Inheritance with method

void printArea() {

double area = 0.5 \* dimension1 \* dimension2;

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

}

Class Circle extends Shape {

public Circle(int radius) {

super(radius, 0);

}

void printArea() {

double area = Math.PI \* dimension1 \* dimension2;

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

}

3) (Inheritance) Inheritance Example

public class Main {

public static void main (String [] args) {

Y

Rectangle rectangle = new Rectangle(5, 10);

Triangle triangle = new Triangle(3, 8);

Circle circle = new Circle(4);

rectangle.printArea();

triangle.printArea();

circle.printArea();

3

3

Output:- Area of rectangle : 50

Area of Triangle = 120

Area of Circle = 50.2654

```
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;
```

```
}
```

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

```
balance += amount;
```

```
System.out.println ("Deposit successful.
```

```
Updated balance " + balance)
```

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

```
System.out.println ("Balance for account "
```

```
accountNumber " + " " + balance)
```

```
}
```

```
public void depositInterest (double rate) {
```

```
    if ("Savings".equals(accountType)) {
```

$$\text{double interest} = \text{balance} * (\text{rate}/100)$$

```
    balance += interest;
```

```
System.out.println ("Interest deposited.
```

```
    Updated balance: " + balance);
```

```
} else {
```

```
System.out.println ("No interest for current  
account");
```

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

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

```
        balance -= amount; now: 300.0
```

```
        System.out.println ("Withdrawal successful.
```

```
        Updated balance: " + balance);
```

```
}
```

```
else {
```

```
System.out.println ("Insufficient funds  
for withdrawal");
```

```
}
```

```
class SavAcc extends Account {  
    public SavAcc (String customerName,  
        long accountNumber, double balance) {  
        super (customerName, accountNumber,  
            "Savings", balance);  
    }  
}
```

```
class Current extends Account {  
    double minBalance;  
    double serviceCharge;  
    public Current (String customerName,  
        long accountNumber, double balance,  
        double minBalance, double serviceCharge) {  
        this.minBalance = minBalance;  
        this.serviceCharge = serviceCharge;  
    }  
}
```

```
public void withdraw (double amount) {  
    if (amount <= balance - minBalance) {  
        balance -= amount;  
        System.out.println ("Withdraw successful.  
        Updated balance : ", +balance);  
    }  
    else  
        System.out.println ("Insufficient funds for  
        withdrawal.");  
}
```

```
3
```

public void checkMinimumBalance()  
{  
if (balance < minBalance)  
{

`borrow = "ServiceCharge"`  
`send ("Service charge imposed for  
falling below minimum balance.  
Update balance : " + balance)"`

public class Bank {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 SavingsAccount account = new SavingsAccount("John Doe", 123456789, 1000);  
 }  
}

Our Act current Account = new Contact  
("Jane Doe", 9876543  
2000, 500.50);

Sous ("Saving Account Operations");  
savings Account · display Balance();  
savings Account · deposit(500);  
savings Account · defact Interest(5.0);  
savings Account · withdraw(200,0);

```
cont ("In Current Account Operations");  
Current Account. displayBalance();  
Current Account. deposit(1000);  
Current Account. withdraw(500);  
Current Account. checkMinimumBalance();  
Scanner. close();
```

Output :-

Balance for account 1234 56789 : 1000

Deposit successful. Updated balance 1500

Interest deposited. Updated balance 1575

Withdraw successful. Updated

balance 1075.0

Current Account Operations:

Balance for account 987654321:

2000

Deposit successful. Updated balance 3000

Withdraw successful. Updated balance 2000

27/12/24

29-01-2024

## Lab Program - 6

### Package

FOLDER CIE

Student.java

package CIE;

public class Student {

    public String usn;

    public String name;

    public int sem;

}

Internals.java

package CIE;

public class Internals extends Student {

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

}

Folder SEE

package SEE;

import CIE.Student;

public class External extends Student {

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

}

Main.java

import CIE.\*;

import SEE.External;

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

```
public class Main {
```

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

```
    {  
        int n; Scanner sc = new Scanner (System.in);
```

```
        int [] finalMarks = new int [5];
```

```
        sout ("Enter the no. of students:-");
```

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

```
        Internals [] final = new Internals [n];
```

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

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

```
            final [i] = new Internals ();
```

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

```
            sout ("Enter Name of student " + (i+1));
```

```
            final [i].name = sc.next();
```

~~```
sout ("Enter USN of Student " + (i+1));
```~~~~```
final [i].USN = sc.next();
```~~~~```
sout ("Enter the current semester of  
student " + (i+1));
```~~~~```
final [i].sem = sc.nextInt();
```~~

sout (" Enter marks obtained in Internals  
(for all 5 courses) for student")

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

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

final[i].marksInternals[j] = sc.nextInt();

}

sout (" Enter the marks obtained in  
Externals (for all 5 courses)");

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

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

final[E(i)].marksExternals[j] = sc.nextInt();

}

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

final[E(i)].marksExternals[j] /= 2;

}

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

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

finalMarks[j] = final[i].marksInternals[j] +  
final[E(i)].marksExternals[j]

3

sout ("Final marks for " + final[i].name

" of USN " + final[i].USN + "

studying in sem : "final[i].sem

+ " has total marks : ");

for ( ent i=0; j<5; i++)

sout ( final Marks(j) );

Output :

Enter the number of students : 2

Enter the name of student 1 :- Agust

USN :- LBM2234

current semester :- 3<sup>rd</sup>

marks obtained in Internals for student 1

14

12

40

25

16

marks obtained into Externals

78

90

89

86

94

Final marks for student 2 - USN 1BMS2CS36

Enter name of student 2 Busso

USN 1BMS2CS36

Current Semester :- 3<sup>rd</sup>

marks obtained in Internals :- 25

36

12

10

26

marks obtained in Externals :- 86

78

66

96

99

Final marks for ayash of USN 1BMS2CS34

student in sem 3 has total marks:

53

57

84

73

62

Final marks for busso of USN 1BMS2CS36

student in sem 3 has total marks

69

75

45

29

75

29 11 29 8

19-02-2024

## Lab Program 7

Class WrongAgeException extends Exception {  
 public WrongAgeException (String message) {  
 super (message);  
 }  
}

Class Father {

private int age;

public Father (int age) throws  
 WrongAgeException {  
 if (age < 0) {  
 throw new WrongAgeException ("Age cannot  
 be negative");  
 }  
 this.age = age;  
}

public int getAge () {

return age;

}

Class Son extends Father {

private int sonAge;

public Son (int fatherAge, int sonAge)

throws WrongAgeException {

super (fatherAge);

if (sonAge >= fatherAge) {

throw new WrongAgeException ("

Son's age cannot be greater  
than or equal to father's age");  
}

this.sonAge = sonAge;  
}]

public int getSonAge() {  
 return sonAge;  
}

}

public class Main {

public static void main (String [] args)

try {

int fatherAge = 45;

int sonAge = 20;

Son son = new Son (fatherAge, sonAge);

sout ("Both father and son ages are  
 valid");

} catch (WrongAgeException e) {

sout ("Error" + e.getMessage());

}

}

}

Output:

Both father and son ages are  
valid

## Lab Program 8

Threads.java

Class BMSDisplayThread extends Thread {

public void run() {

while (true) {

sout ("BMS College of Eng.");

try {

Thread.sleep (10000);

}

} catch (InterruptedException e) {

e.printStackTrace();

}

}

Class CSEDisplayThread extends Thread {

public void run() {

while (true) {

sout ("CSE");

try {

Thread.sleep (2000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

public class Threads {

public static void main (String [] args)

main() -> main() -> main() -> main() -> main()

BMS Display Thread bmsThread = new

BMSDisplayThread();

CSE Display Thread csethread = new CSEDisplay  
Thread();

bmsThread.start();

cseThread.start();

}

3. Output:

BMS College of Engineering

CSE College of Engineering

CSE College of Engineering

CSE College of Engineering

CSE College of Engineering

(108, 89, 601, 293) character

## Program 9:-

Create Label, button and Text Field in a frame using AWT and Event Listener.

```
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();
```

```
t.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);
```

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

```
f.setLayout(null);
```

```
f.setVisible(true);
```

```
}
```

public void actionPerformed (WindowEvent e)

System.out.println ("button clicked")

}

public static void main (String args)

{

AWTExample obj = new AWTExample () ;

obj.setVisible (true) ;

}

Output :-

Employee id

|     |        |
|-----|--------|
| 123 | Submit |
|-----|--------|

Submit

2. Create a button and add an 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 (60, 50, 170, 20);
```

```
Button b = new Button ("click me");
```

b.addActionListener (this);

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

```
f.add (tf);
```

```
f.setLayout (null);
```

```
f.setVisible (true);
```

```
public void actionPerformed (ActionEvent)
```

```
{  
    tf.setText ("Welcome");
```

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

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

Output :-

blupending

click me

welcome

click me

Ari  
26.02.94