



Quadratic Equation

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

```
class Quadratic
```

```
{
```

```
    int a, b, c;
```

```
    double x1, x2, d;
```

```
    void input () {
```

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

```
        System.out.print println ("Enter the  
coefficients of a, b, c");
```

```
        a = s.nextInt();
```

```
        b = s.nextInt();
```

```
        c = s.nextInt();
```

```
    }
```

```
    void compute () {
```

```
        while (a == 0)
```

```
        {
```

```
            System.out.println ("Not a quadratic  
equation");
```

```
            System.out.println ("Enter a non-zero  
value for a");
```

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

```
            a = s.nextInt();
```

```
        }
```

$$d = b^2 - 4 * a * c;$$

if (d == 0)

$$r1 = -b / (2.0 * a);$$

System.out.println("Roots are real & equal");

System.out.println("Root 1 = Root 2 = " + r1);

else if (d > 0)

$$r1 = (-b + \text{Math.sqrt}(d)) / (2.0 * a);$$

$$r2 = (-b - \text{Math.sqrt}(d)) / (2.0 * a);$$

System.out.println("Roots are real & equal");

System.out.println("Root 1 = " + r1 + "
Root 2 = " + r2);

}

else if (d < 0)

{

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

$$r1 = -b / (2.0 * a);$$

$$r2 = \text{Math.sqrt}(\text{Math.abs}(d)) / (2.0 * a);$$

System.out.println("Roots = " + r1 + " +
i" + r2);

System.out.println("Roots = " + r1 + " -
i" + r2);

}

Class QuadraticMain

{

public static void main (String[] args)

{
 Quadratic q = new Quadratic();
 q = input();
 q = compute();
}

}

→ Enter the coefficient of a, b, c

1 5 6

Roots are real & distinct

Root 1 = -2.0 // Root 2 = -3.0

→ Enter the coefficient of a, b, c

1 2

2

3

Roots are imaginary

Root = -0.1 + 10.5385i

Roots = -0.1 - 10.5385i

→ Enter the coefficient of a, b, c

2

4

2

Roots are real & equal

Root 1 = Root 2 = -1.0

SCIPA of Student



Date: _____

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

```
class Student {
```

```
    String USn, name,
```

```
    int n;
```

```
    int c[], m[];
```

```
    void display() {
```

```
        Scanner s1 = new Scanner(System.in);
```

```
        System.out.println("Enter the USn of student");
```

```
        USn = s1.nextLine();
```

```
        System.out.println("Enter the name of student");
```

```
        name = s1.nextLine();
```

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

```
        n = s1.nextInt();
```

```
        c = new int[n];
```

```
        m = new int[n];
```

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

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

```
            c[i] = s1.nextInt();
```



```
System.out.println("Enter the marks of  
subject" + (i + 1) + " :");
```

```
m[i] = sl.nextInt();
```

```
}
```

```
}
```

```
int gp(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 sapa ()
```

```
{
```

```
int tC = 0, sum = 0;
```



Date: _____

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

```
    t = t * c + c[i];
```

```
    sum = sum + gp[m[i]] * c[i];
```

```
return (double) sum / t;
```

```
void disp()
```

```
{
    System.out.println("USN = " + usn);
    System.out.println("NAME = " + name);
```

```
}

class main
```

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

```
{
    Student s = new Student();
```

```
    s.a();
```

```
    s.disp();
```

```
    System.out.println("The SGPA
```

```
    = " + s.gpa());
```




Date: _____

Enter the usn of the student :

1BM22CS082

Enter the name of student

Enter the number of subjects

3

Enter marks of subject 1

95

Enter the number of credits

4

Enter the marks of subject 2

85

Enter the number of credits

3

Enter the marks of subject 3

75

Enter the number of credits

3

USN = 1BM22CS082

Name = Deekshith B

The SGPA = 8.60.

08/01/24

Book Database

```
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() {}
```

```
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 the book: ");
price =
num - pages = input.nextInt();
System.out.print("\n");
}
```

```
public String toString()
{
    String name, author, 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 Book Run
```

```
{
```

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

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

```
        /* String name, author;
           int priv, num - pages; */
```

```
        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");
```

```
        }
    }
}
```

Shape

Enter number of books : 2

Book 1 :

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

Enter name of author : J. K. Rowling

Enter price of the book : 418

Enter no of pages in book : - 341

Book 2 :

Enter name of book : Adventure of Nagakuni

Enter names of author : Jay

Enter price of the book : 300

Enter no of pages in the book : 250

Book 1 :

Book name : Harry Potter & The Chamber of secrets

author name : J. K. Rowling

price : 418 Rs

Number of page : 341 Page

[i] + "\n").



Date: _____

Book 2:

Book name: Adventures of the Nagaland

Author name: Khuanor

Price: 300 Rs

Number of pages: 250 pages

82/01/24



Date: _____

Shape

```
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");
```

```
    }
```



Date: _____

```
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");
```

```
}
```

```
}
```




Date: _____

Class Shape Run

{

public static void main (String [] args)

{

double x, y;

Scanner input = new Scanner (System.in);

System.out.println ("Enter length &
width of Rectangle:");

x = input.nextDouble();

y = input.nextDouble();

Rectangle rectangle = new Rectangle (x, y);
~~rectangle.printArea();~~

System.out.println ("Enter height & base
of Triangle:");

x = input.nextDouble();

y = input.nextDouble();

Triangle triangle = new Triangle (x, y);
~~triangle.printArea();~~

System.out.println ("Enter radius of
Circle:");

x = input.nextDouble();

Circle circ = new Circle(3);
 while - print Area();

input.close();

3

3

Output

Enter length and width of Rectangle

10

20

Area of Rectangle : 200.0 square

Enter height & base of Triangle

10

5

Area of Triangle : 25 square unit

Enter radius of circle

10

Area of circle : 314.2 square

22/01/24



Date : _____

Bank

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

```
class Amount  
{
```

```
    String customerName;  
    long amountNumber;  
    String amountType;  
    double balance;
```

```
    public Amount(String customerName, long  
amountNumber, String amountType, double balance)
```

```
{
```

```
    this.customerName = customerName;  
    this.amountNumber = amountNumber;  
    this.amountType = amountType;  
    this.balance = balance;
```

```
}
```

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

```
{
```

```
        balance += amount;
```

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

```
}
```




Date : _____

```
public void display Balanc ()  
{  
    System.out.println ("Amount  
Number : " + amount Number)  
    System.out.println ("Custom Name :  
" + customName);  
    System.out.println ("Amount Type : " +  
    amountType);  
    System.out.println ("Balance : " + balance);  
}
```

Class Sav Act extend Amount

```
public Sav Act (String custom Name,  
long amount Number, double balance)  
{  
    super (customName, amount Number,  
    "Savings", balance);  
}
```

```
public void calc as computeInterest  
Interest (double rate)
```

```
double interest = balance * rate / 100;  
balance + = interest;  
System.out.println ("Interest compute  
& deposited. Updated balance : " + balance);
```

```

public void withdraw(double amount)
{
    if (amount <= balance)
    {
        balance -= amount;
        System.out.println("Withdrawal successful.");
        updateSuccessful(balance, balance);
    }
    else
    {
        System.out.println("Insufficient funds. Withdrawal failed.");
    }
}

```

Class Current extends Account

```

{
    double minimumBalance;
    double serviceCharge;
}

```

public Current (String customerName, long amountNumber, double balance, double minimumBalance, double serviceCharge)



Date : _____

{

super (welcome Name, amount Number, "current" ^{balance}
this, minimum Balance = minimum Balance,
this, service Charge = service Charge,

g.

private void checkMinimumBalance()

{

if (balance < minimumBalance)

{

if (balance - ~~balance~~ service Charge
System.out.println ("Minimum balance
not maintained. service charge imposed.
Update balance: " + balance);

}

}

public void withdraw (double amount)

{

if (amount <= balance)

{

balance -> amount,
System.out.println ("Withdrawal
successful. Update balance: " + balance),
checkMinimumBalance(),

}



Date: _____

```
else  
{  
    System.out.println("Insufficient funds. Atleast  
    amount failed");  
}
```

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

```
    Scanner s1 = new Scanner(System.in);  
    System.out.print("Enter Customer name for  
    savings account: ");
```

```
    String SCN = s1.nextLine();  
    System.out.print("Enter initial balance for  
    savings account");
```

```
    String SAN = s1.nextLine();
```

```
    System.out.print("Enter initial balance for  
    savings account: ");
```

```
    double SIR = s1.nextDouble();
```

```
    SavingsAcct A = new SavingsAcct(SCN, SAN, SIR);
```



Date: _____

System.out.print("Enter custom name for Current Account: ");

String CCN = S1.next();

System.out.print("Enter account number for Current Account: ");

String CCN = S1.next();

System.out.print("Enter initial balance for Current Account: ");

double CIB = S1.nextDouble();

System.out.print("Enter minimum balance for Current Account: ");

double MB = S1.nextDouble();

System.out.print("Enter service charge for Current Account: ");

double SC = S1.nextDouble();

Current A = new Current(CCN, AN, CIB, MB, SC);

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

double SDA = S1.nextDouble();

SA.deposit(SDA);

System.out.print("Enter interest rate for Savings Account: ");



Date: _____

double SIR = SI. next Double();

SA. compute and Deposit Interest (SIR);

System.out.print ("Enter withdrawn amount for
Savings Account: ");

double SWA = SI. next Double();

SA. withdraw (SWA);

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

double CDA = SI. next Double();

CA. deposit (CDA);

System.out.print ("Enter withdrawn amount for
Current Account: ");

double CWA = SI. next Double();

CA. withdraw (CWA);

System.out.print ("In Fixed Balance: ");

System.out.print ("Savings Account ");

SA. display Balance();

System.out.print ("In Current Account: ");

CA. display Balance();

3
2

27/1/24



Packages

CIE package :- In CIE folder.

Student.java

Package CIE;

public class Student

String name;

String usn;

int sem;

public Student (String name, String usn, int sem)

{

this.name = name;

this.usn = usn;

this.sem = sem;

}

}



Date: _____

Internal.java

```
package CIE;
import CIE.Student;
```

```
public class Internal extends Student
{
```

```
    public int [] InternalMarks;
```

```
    public Internal (String name, String USN,
                    int sem, int [] Internal Marks)
    {
```

```
        super (name, USN, sem);
```

```
        this.InternalMarks = InternalMarks;
```

SEE package :- In SEE folder :-

```
package SEE;
```

```
import CIE.Student;
```

```
public class External extends CIE.Student
{
```

```
    public int [] SeeMarks;
```

```
    public External (String name, String USN, int sem,
                    int [] See Marks)
```




Date: _____

```
{ super (name, usn, sem),  
  this, seemarks = Seemarks,  
}  
}
```

Final marks.java

```
import CIE.Student;  
import CIE.Internal;  
import SEE.Externals;  
import java.util.Scanner;
```

```
public class Final Marks {
```

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

```
        Scanner s1 = new Scanner (System.in);
```

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

```
        int n = s1.nextInt();
```

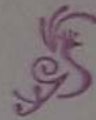
```
        String[] name = new String[n];
```

```
        String[] usn = new String[n];
```

```
        int[] sem = new int[n];
```

```
        int[][] Internal Marks = new int[n][5];
```

```
        int[][] Seemarks = new int[n][5];
```

Date : _____

```
for (int i = 0; i < n; i++) {
    System.out.println("Enter details of student " +
        (i + 1) + " : ");
    S.O.P (" | Name: ");
    name[i] = s1.next();
    S.O.P (" | USN: ");
    usn[i] = s1.next();
    S.O.P (" | Sem: ");
    sem[i] = s1.nextInt();
}
```

```
S.O.P ("Enter internal marks for course: ");
for (int j = 0; j < 5; j++) {
    S.O.P (" | Course " + (j + 1) + " : ");
    Internal marks[i][j] = s1.nextInt();
}
```

```
System.out.println("Enter See marks for
5 courses: ");
for (int j = 0; j < 5; j++) {
    S.O.P (" | Course " + (j + 1) + " : ");
    See marks[i][j] = s1.nextInt();
}
```



Date : _____

```
int [][] Finalmarks = new int [n] [5];
for (int i = 0; i < n; i++) {
    Internal I1 = new Internal (name[i],
    sem[i], Internal marks[i],
    External E1 = new External (name[i], sem[i],
    sem[i], Seemarks[i]);
    int sum = 0;
    for (int j = 0; j < 5; j++) {
        Finalmarks[i][j] = I1.Internalmarks
        [i] + E1.Seemarks[j];
    }
}
```

SOP ("In In Final marks for" + n +
"Student in 5 courses :");

BOP (name[k] + " : ");

```
for (int j = 0; j < 5; j++) {
    sum += Finalmarks[i][j];
}
```

System.out.println("sum");

s1.close();

29/1/29



Exception Handling

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

```
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 sendAge;
```

```
    public int receiveAge;
```

```
}
```

```
}
```




Date : _____

```
class Son extends Father {  
    private int sonAge;
```

```
    public Son(int FatherAge, int sonAge)  
        throws WrongAgeException {  
        super(fatherAge);
```

```
        if (sonAge >= fatherAge) {  
            throw new WrongException ("Son's age  
            should be less than Father's age");  
        }
```

```
        this.sonAge = sonAge;
```

```
    }  
    public int getSonAge() {  
        return sonAge;
```

```
    }  
}
```

```
class ExceptionHandling Demo {
```

```
    public static void main (String [] args)  
    {  
        Scanner scanner = new Scanner (System
```

```
        try {
```

```
            // System.out.print ("Enter Father's Age  
            int fatherAge = scanner.nextInt();
```



Date: _____

```
Father father = new Father ( fatherAge );  
System.out.println ( " Father's Age "  
+ father.getAge() );
```

```
System.out.println ( " Enter Son's Age " )  
int sonAge = scanner.nextInt ();  
Son son = new Son ( father.getAge(), sonAge )  
System.out.println ( " Son's Age " + son.getSonAge(),
```

}

```
catch ( WrongAgeException e ) {  
System.out.println ( " Exception caught " + e.  
getMessage() );
```

}

```
finally {
```

```
scanner.close();
```

}

}

}

Output

• Enter Father's age:

50

Enter Son's age:

25

Father's age: 40

Son's age: 25

Threads

```

class Thread1 extends Thread
{
    public void run()
    {
        while (true)
        {
            System.out.println("BMS College of Engineering");
            try
            {
                Thread.sleep(1000);
            }
            catch (InterruptedException e)
            {
            }
        }
    }
}

```

```

class Thread2 extends Thread
{
    public void run()
    {
        while (true)

```




Date: _____

{

System.out.println("(SE"),

try
{

Thread.sleep(2000),

}

catch (InterruptedException e)

{

}

}

}

}

public class Threads

{

public static void main (String [] args)

{

Thread t1 = new Thread(),

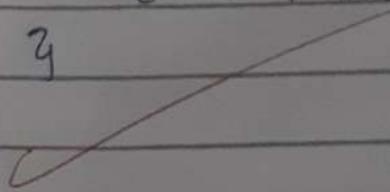
Thread t2 = new Thread(),

t1.start(),

t2.start(),

}

}





Date : _____

Bms College of Engineering

CSE

CSE

CSE

CSE

CSE

Bms College of Engineering

CSE

CSE

CSE

CSE

CSE

~~Bms College of Engineering~~

19.02.24