

Q1 Simple Calculator

```
import java.util.Scanner;
public class void main (String[] args)
{
    Scanner scanner = new Scanner (System.in);
    System.out.print (" Enter the first no: ");
    double n1 = scanner.nextDouble();
    System.out.print (" Enter the second no: ");
    double n2 = scanner.nextDouble();
    System.out.println (" select operation: ");
    System.out.println (" 1. Addition (+) ");
    System.out.println (" 2. Subtraction (-) ");
    System.out.println (" 3. Multiplication (*) ");
    System.out.println (" 4. Division (/) ");
    System.out.print (" Enter choice (1-4): ");
    int choice = scanner.nextInt();
    double result = 0;
    switch (choice)
    {
        case 1 :
            result = n1 + n2;
            break;
        case 2 :
            result = n1 - n2;
        case 3 :
            result = n1 * n2;
        case 4 :
            if (n2 != 0)
                result = n1 / n2;
            else
```

```

    {
        System.out.println("Error: cannot divide by zero");
        return;
    }
    break;
default:
    System.out.println("Invalid choice. Please enter
        a number between 1 & 4");
    return;
}

System.out.println("Result : " + result);
scanner.close();
}
}

```

Q2 To check whether a no is prime or not.

```

import java.util.Scanner;

public class PrimeNumberCheck
{
    public static void main (String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number");
        int number = scanner.nextInt();
        if (

```

Output:

Enter the first number : 5
 Enter the second number : 6

Select Operation

1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)

Enter choice(1-4)=1
 Result : 11.0

28 Prime Number

```
class Main {  
    public static void main (String[] a)  
    {  
        int n = 5;  
        boolean flag = false;  
        for (int i = 2; i <= num/2; ++i)  
        {  
            if (num % i == 0)  
            {  
                flag = true;  
                break;  
            }  
        }  
        if (flag == true)  
            System.out.println(n + " is not prime");  
        else  
            System.out.println(n + " is a prime no");  
    }  
}
```

Output 5 is a prime number

38 Fibonacci series

```
class Main {  
    public static void main (String[] args)  
    {  
        int n = 10, firstTerm = 0, secondTerm = 1;  
        System.out.println ("Fibonacci series till " + n + "  
                                terms:");  
        for (int i = 1; i <= n; ++i)  
        {  
            System.out.print (firstTerm + ", ");  
            int nextTerm = firstTerm + secondTerm;  
            firstTerm = secondTerm;  
            secondTerm = nextTerm;  
        }  
    }  
}
```

20/10/20

}

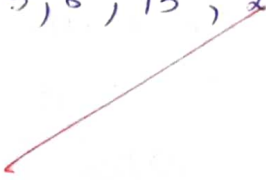
}

}

Output

Fibonacci Series fill 10 terms:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34,



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.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 eqn");
```

```
            System.out.println("Enter a non-zero value for 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("roots1 = roots2 = " + r1);
```

```
        } else if (d > 0)
```

```
{
```

```
            r1 = ((-b) + (Math.sqrt(d))) / (double)(2*a);
```

```
            r2 = ((-b) - (Math.sqrt(d))) / (double)(2*a);
```



```

System.out.println("Roots are equal & distinct");
System.out.println("root1 = " + r1 + " root2 = " + r2);
}
else if (d < 0)
{
    System.out.println("Roots are imaginary");
    r1 = (-b) / (2 * a);
    System.out.println("roots = " + r1 + " + i" + r2);
    System.out.println("roots = " + r1 + " - i" + r2);
}
}
}

```

```

class QuadraticMain
{
    public static void main (String[] arg)
    {
        Quadratic q = new Quadratic();
        q.input();
        q.compute();
    }
}

```

output:

Enter the coefficients of a, b, c

1
-1
1

Roots are not real

Enter ~~the~~ coefficients of a b c

1
9
-1

Roots are: -0.1125 and 8.887

08/10/21 Roots are real and distinct

2] Students

```
import java.util.*;  
class student  
{
```

```
String usn;
```

```
String name;
```

```
int credits[];
```

```
int marks[];
```

```
void marks[];
```

```
void accept()
```

```
{
```

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

```
System.out.println("Enter the USN");
```

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

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

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

```
System.out.println("Enter the no of subjects");
```

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

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

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

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

```
{
```

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

```
credits[i] = s.nextInt();
```

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

```
marks[i] = s.nextInt();
```

```
}
```

```
void display()
```

```
{
```

```
System.out.println | "usrn : "+ usn);  
System.out.println("name: "+ name);  
for (int i=0; i < credits.length; i++)  
{
```

```
    System.out.println("Subject "+ (i+1) + " credits [" + credits[i] + "];  
    }  
}
```

```
double calculateGpa()  
{
```

```
    double sgpa = 0;
```

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

```
        sgpa = credits[i] * marks[i];  
    }
```

```
    sgpa /= credits.length;
```

```
    return sgpa;
```

```
    }  
}
```

```
public class Main  
{
```

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

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

```
        s.accept();
```

```
        s.display();
```

```
        System.out.println("sgpa : " + s.calculateGpa());
```

```
    }  
}
```


Output:

Enter number of students: 2

Enter USN: IBM22CS077

Enter name: chaitra

Enter marks for 6 subjects:

98

96

97

85

90

95

Enter USN: IBM22CS075

Enter name: Chahat

Enter marks for 6 subjects.

90

95

98

85

80

82

Student Details:

USN: IBM23CS076

Name: Chaitanya

Marks: 98, 96, 97, 85, 90, 95.

Percentage: 93.5%.

USN: IBM22CS076

Name: clara

Marks: 90, 95, 98, 85, 80, 82

Percentage: 80.0%.

3] Book database

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

```
class Books
```

```
{
```

```
    String name;
```

```
    String author;
```

```
    int price;
```

```
    int numPages;
```

```
    Books()
```

```
    {}
```

```
    Books(String name, String author, int price, int  
           numPages)
```

```
    {
```

```
        this.name = name;
```

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

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

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

```
    }
```

```
    public String toString()
```

```
    {
```

```
        String name, author, price, numPages;
```

```
        name = "Book name:" + this.name + "\n";
```

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

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

```
        numPages = "Number of pages:" + this.numPages  
                   + "\n";
```

```
        return name + author + price + numPages;
```

```
    }
```

```
}
```

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 ("Enter the number of books:");

n = s.nextInt();

Books b[];

b = new Books[n];

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

{

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

System.out.print ("Enter name of book:");

name = s.next();

System.out.print ("Enter author:");

author = s.next();

System.out.print ("Enter price:");

price = s.nextInt();

System.out.print ("Enter no of pages:");

numPages = s.nextInt();

b[i] = new Books (name, author, price, numPages);

}

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

System.out.print ("Book : " + (i+1) + " : \n" + b[i]);

}

}

Output:

Enter the number of book : 2

Book 1 :

Enter the name of the book = Jungle - Book

Enter the author of the book = Rudyard - Kipling.

Enter the price of the book : 1000

Enter the number of pages of the book : 500

Book 2 :

Enter the book name of the book : ~~Enter~~ ~~of~~ ~~Game of Thrones~~

Enter the author of the book : George R.R. Martin.

Enter the price of the book : 900

Enter the number of ~~pages~~ of the book : 400

Book 1 :

Bookname : Jungle - Book

Author : Rudyard - Kipling

Price : 1000

Number of pages : 500

Book 2 :

Book name : ~~Enter~~ Game of Thrones

Author of the book : George R.R. Martin

Price : 900

Number of pages : 400

2) Shapes

```
abstract class Shape {
```

```
    protected int length;
```

```
    protected int breadth;
```

```
    public Shape(int length, int breadth)
    {
```

```
        this.length = length;
```

```
        this.breadth = breadth;
```

```
    }
```

```
    public abstract void printArea();
}
```

```
class Rectangle extends Shape {
```

```
    public Rectangle(int length, int breadth)
    {
```

```
        super(length, breadth);
```

```
    }
```

```
    public void printArea()
    {
```

```
        int area = length * breadth;
```

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

```
    }
```

```
}
```

```
class Triangle extends Shape {
```

```
    public Triangle(int length, int breadth)
    {
```

```
        super(length, breadth);
```

```
    }
```



```
public void printArea() {
```

```
    double area = 0.5 * length * breadth;  
    System.out.println("Triangle Area: " + area);  
}
```

```
class Circle extends Shape  
{
```

```
    public Circle(int radius)  
    {
```

```
        super(radius, 0);  
    }
```

```
    public void printArea()  
    {
```

```
        double area = Math.PI * radius * radius;  
        System.out.println("Circle Area: " + area);  
    }
```

```
public class Main {
```

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

```
        Rectangle rectangle = new Rectangle(5, 10);
```

```
        Triangle triangle = new Triangle(4, 8);
```

```
        Circle circle = new Circle(4);
```

```
        rectangle.printArea();
```

```
        triangle.printArea();
```

```
        circle.printArea();  
    }
```

Output:

Rectangle Area: 50

Triangle Area: 10.0

Circle Area: ~~113.0973~~

2/ 22/01/24

Bank Account

```
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 ("Account Number: " + accountNumber);
```

```

    System.out.println ("Customer Name:" + customerName);
    System.out.println ("Account Type:" + accountType);
    System.out.println ("Balance:" + balance);
}
}

```

```

class SavAcct extends Account
{

```

```

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

```

```

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

```

```

    public void computeAndDepositInterest (double rate)
    {

```

```

        double interest = balance * rate / 100;

```

```

        balance += interest;

```

```

        System.out.println ("Interest computed &
                             deposited. updated
                             balance:" + balance);
    }

```

```

    public void withdraw (double amount)
    {

```

```

        if (amount <= balance)
        {

```

```

            balance -= amount;

```

```

            System.out.println ("Withdrawal
                                Successful. updated balance:" + balance);
        }

```

```

    else
    {

```

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

```
class CurrAcct extends Account
```

```
double minimumBalance;
```

```
double serviceCharge;
```

```
public CurrAcct (String customerName, long accountNumber,  
double balance, double minimumBalance, double  
serviceCharge)
```

```
{
```

```
super(customerName, accountNumber, "Current", balance);
```

```
this.minimumBalance = minimumBalance;
```

```
this.serviceCharge = serviceCharge;
```

```
}
```

```
private void checkMinimumBalance()
```

```
{
```

```
if (balance < minimumBalance)
```

```
{
```

```
balance -= serviceCharge;
```

```
System.out.println("Minimum balance  
not maintained. Service charge imposed.  
Updated balance: " + balance);
```

```
}
```

```
}
```

public void withdraw (double amount)

{

if (amount <= balance)

{

balance -= amount;

System.out.println (" withdrawal success,
Updated balance: " +
balance);

checkMinimumBalance();

}

else

{

System.out.println (" Insufficient funds
withdrawal failed!

}

}

}

public class Bank

{

psvm (String[] args)

{

Scanner SI = new Scanner (System.in);

SOI (" Enter customer name for
savings Account: ");

String SCN = SI.nextLine();

System.out.println (" Enter account no for
savings account: ");

~~String~~ SCN Long SAM = SI.nextLong();

SOP (" Enter Initial balance for
savings Account: ");

double SIB = SI.nextDouble();


```

SavAcctSA = new SavAcct (SCN, SAN, SIB);
SOP("Enter customer name for current Account:");
String CCN = SI.next();
SOP("Enter account number for current Account:");
String en long CAN = SI.nextLong();
SOP("Enter initial balance for current Account");
double CIB = SI.nextDouble();
CurAct CA = new CurAct (CCN, CAN, CIB, MB, SC);
SOP("Enter deposit amount for savings
Account:");
double SDA = SI.nextDouble();
SA.deposit(SDA);
SOP("Enter interest rate for savings
Account:");
double SIR = SI.nextDouble();
SA.computeAndDepositInterest(SIR);
SOP("Enter deposit amount for
current Account:");
double CDA = SI.nextDouble();
CA.deposit(CDA);
SOP("Enter withdrawal amount
for current Account:");
double CWA = SI.nextDouble();
CA.withdrawal(CWA);
SOP("Final Balances:");
SOP("Savings Account:");
SA.displayBalance();
}
}

```

Output

Enter customer name for savings Account : Raj

Enter account no for savings Account : 2021

Enter initial balance for savings Account : 10000

Enter customer name for current Account : Raj

Enter account number for current Account : 202123

Enter initial balance for current Account : 1000

Enter minimum balance for current Account : 5000

Deposit Successful

Updated balance : 15000

Enter Interest rate for saving Account : 4

Interest computed and deposited

Updated balance : 15600.0

Enter withdrawal amt for saving Account : 70

Withdrawal Successful

Updated balance : 14900.0

```
package class Student
{
```

```
    protected public String usn;
    protected public String name;
    protected public int sem;
```

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

```
        this.usn = usn;
```

```
        this.name = name;
```

```
        this.sem = sem;
```

```
    }
}
```

```
package CIE ;
```

```
public class Internals extends Student
{
```

```
    public int[] internal Marks;
```

```
    public Internals (String usn, String name,
        int sem, int[] internalMarks)
```

```
    {
```

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

```
        this.internalMarks = internalMarks;
```

```
    }
```

```
}
```

```
package SEE;
```

```
import CIE.Student;
```

```
public int[] seeMarks;
```

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

```
{
```

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

```
    this.seeMarks = seeMarks;
```

```
}
```

```
}
```

```
import CIE.internals;
```

```
import SEE.External;
```

```
public class Main
```

```
{
```

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

```
{
```

```
        int[] internalMarks = {75, 80, 90, 85,  
                                88};
```

```
        internals student1 = new internals
```

```
        ("1ABC123", "John", 3,  
         internalMarks);
```

```
        int[] seeMarks = {70, 85, 88, 92, 78};
```

```
        External student2 = new External ("2XYZ456",
```

"Jane", 3, seeMarks);

```
S.O.P ("Student 1:");  
S.O.P ("USN:" + student1.usn);  
S.O.P ("Name:" + student1.name);  
S.O.P ("Semester " + student1.sem);  
S.O.P ("Internal Marks:" + java.util.Arrays.  
toString(student1.internalMarks));
```

```
S.O.P ("In Student 2:");  
S.O.P ("USN:" + student2.usn);  
S.O.P ("Name:" + student2.name);  
S.O.P ("Semester:" + student2.sem);  
S.O.P ("SEE Marks:" + java.util.Arrays.toString(  
student2.seeMarks));  
}  
}
```

Output :

Student 1:

USN: 1ABC123

Name: John

Semester : 3

Internal Marks (75, 80, 90, 85, 88)

Student 2:

USN: 2X42 456

Name : Jane

Semester : 3

SEE marks = [70, 85, 88, 92, 78]

27/11/24

19/02/2024

Q. Write a programme which creates two threads one thread displaying "BMS College of Engineering" once every ten seconds and another thread displaying "CSE" once every 2 seconds

```
class BMS implements Runnable {
```

```
    public void run ()
```

```
    {
```

```
        while (true)
```

```
        {
```

```
            try
```

```
            {
```

```
                System.out.println("BMS College of Engg");
```

```
                Thread.sleep(10000);
```

```
            } catch (InterruptedException e)
```

```
            {
```

```
                e.printStackTrace();
```

```
            }
```

```
        }
```

```
class CSE implements Runnable
```

```
{
```

```
    public void run()
```

```
    {
```

```
        while (true)
```

```
        {
```

```
            try {
```

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

```
                Thread.sleep(2000);
```

```
            }
```

```
            catch (InterruptedException e)
```

```
            {
```

```
e. printStackTrace();  
}
```

```
public class Main  
{
```

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

```
Thread t1 = new Thread(new BMS College of Engg());
```

```
Thread t2 = new Thread(new CSE());
```

```
t1.start();
```

```
t2.start();
```

```
}
```

Output: BMS College of Engg

CSE

CSE

CSE

CSE

CSE

BMS college of Engg

CSE

CSE

CSE

CSE

CSE

BMS college of Engg

CSE

CSE

CSE

CSE

CSE

BMS College of Engg

Q Write a programme that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends base class. In father class, implement a constructor which takes the age and throws the exception wrongAge() when input age less than 0. In son class implement a constructor that takes both father and son's age and throws an exception if son's age is \geq father's age.

```
import java.util.Scanner;  
class wrongAge extends Exception  
{  
    public wrongAge (String message)  
    {  
        super (message);  
    }  
}
```

```
class Father  
{
```

```
    int fatherAge;  
    Father () throws wrongAge  
    {
```

```
        Scanner s = new Scanner (System.in);  
        System.out.println ("Enter Father's Age :");  
        fatherAge = s.nextInt();  
        if (fatherAge < 0)  
        {
```

```
throw new WrongAge("Age cannot be negative");  
}
```

```
}
```

```
void display()
```

```
{
```

```
    System.out.println("Father age is : " + fatherAge);
```

```
}
```

```
}
```

```
class Son extends Father
```

```
{
```

```
    int SonAge;
```

```
    Son() throws WrongAge
```

```
{
```

```
    Super();
```

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

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

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

```
    if (SonAge > FatherAge)
```

```
{
```

```
        throw new WrongAge("Son's age  
cannot be greater than father's  
age");
```

```
}
```

```
    else if (SonAge == FatherAge)
```

```
        throw new WrongAge("Son's age  
cannot be equal to father's age");
```

```
}
```

```
    else if (SonAge < 0)
```

```
{  
    throw new WrongAge ("Age cannot be  
        negative");  
}
```

```
}
```

```
void display()  
{
```

```
    Super.display()
```

```
    System.out.println ("Son's age is "  
        sonAge);  
}
```

```
}
```

```
public class Main
```

```
{
```

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

```
    {
```

```
        try {
```

```
            son = new Son();
```

```
            s.display();  
        }
```

```
        catch (WrongAge e)
```

```
        {
```

```
            System.out.println (e.getMessage());  
        }
```

```
    }
```

```
}
```

```
}
```


output: Enter Father's age : 40
Enter Son's age : 20
Father's Age is : 40
Son's Age is : 20

Enter Father's age :
Enter Son's age :

Son's age cannot be equal to Father's age

Enter Father's age = -30
~~Age cannot be negative~~

Am
19.02.24

26/02/2022

AWT

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

public class AWTExample extends WindowAdapter {
    Frame f;

    AWTExample()
    {
        f = new Frame();
        f.addWindowListener(this);
        Label label = new Label(text: "Employee id:");
        Button b = new Button(label: "Submit");
        TextField t = new TextField();
        label.setBounds(x: 20, y: 80, width: 80, height: 30);
        t.setBounds(x: 20, y: 100, width: 80, height: 30);
        b.setBounds(x: 100, y: 100, width: 80, height: 30);
        f.add(b);
        f.add(label);
        f.add(t);
        f.setSize(width: 400, height: 300);
        f.setTitle(title: "Employee info");
        f.setLayout(null);
        f.setVisible(b: true);
    }

    public void windowClosing(WindowEvent e)
```

System.exit(status:0);

}

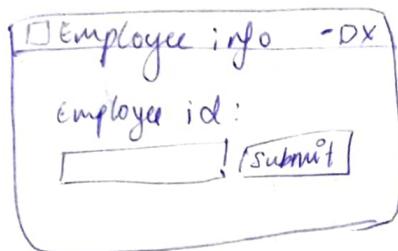
```
public static void main(String[] args)
{
```

```
    AWTExample awt = new AWTExample();
```

```
}
```

```
}
```

Output :



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

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

```
public class EventHandling extends WindowAdapter implements
    ActionListener {
```

```
    Frame f;
```

```
    TextField tf;
```

```
    EventHandling()
```

```
{
```

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

```
    f.addWindowListener(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: 80, height: 30);
        b.addActionListener(this);

        f.add(b);
        f.add(tf);

        f.setSize(width: 300, height: 300);
        f.setTitle(title: "Employee info");
        f.setLayout(mgr: null);
        f.setVisible(b: true);
    }

```

```

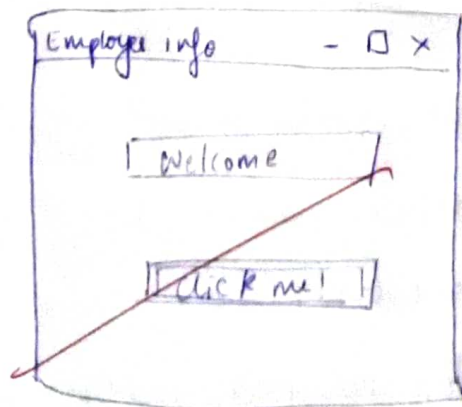
public void actionPerformed(ActionEvent e)
{
    tf.setText(t: "Welcome");
}

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

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

```

Output:



26.02.24