

1) Java Program to print an Integer Entered by the user

Algorithm:

Step 1 : Start

Step 2 : Import java.util.Scanner

Step 3 : Take Input from the user / create reader instance

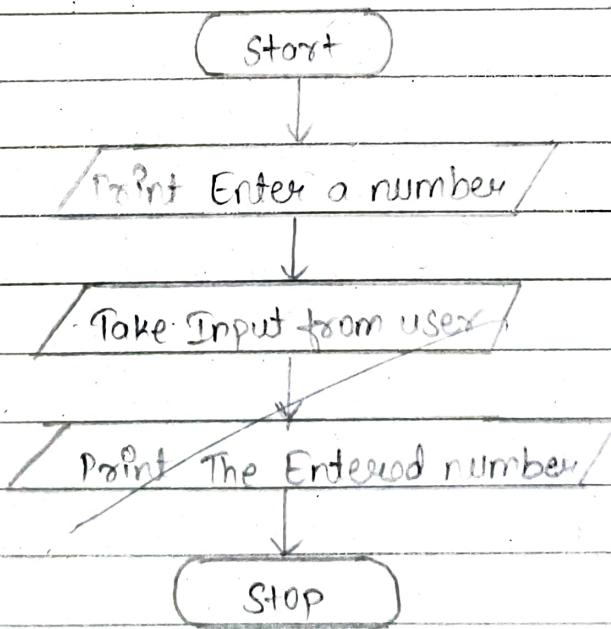
Step 4 : Print Enter a number

Step 5 : Read the number entered by user

Step 6 : Print the entered number

Step 7 : Stop

flowchart:



Code :

```
import java.util.Scanner;  
public class HelloWorld  
{  
    public static void main (String [] args)  
    {  
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter a number: ");  
        int number = sc.nextInt();  
        System.out.print ("You entered: " + number);  
    }  
}
```

Output :

Akshorra Singa

IBM22CS029

Enter a number 29

You entered: 29

```
C:\Users\bnsce\Desktop\1bm22cs029>javac HelloWorld.java
```

```
C:\Users\bnsce\Desktop\1bm22cs029>java HelloWorld
```

Akshara Singa

1BM22CS029

Enter a number29

You entered: 29

2) Java program to check whether a number is even or odd

### Algorithm

Step 1: Start

Step 2: Print enter a number

Step 3: Take input from user and store in variable num

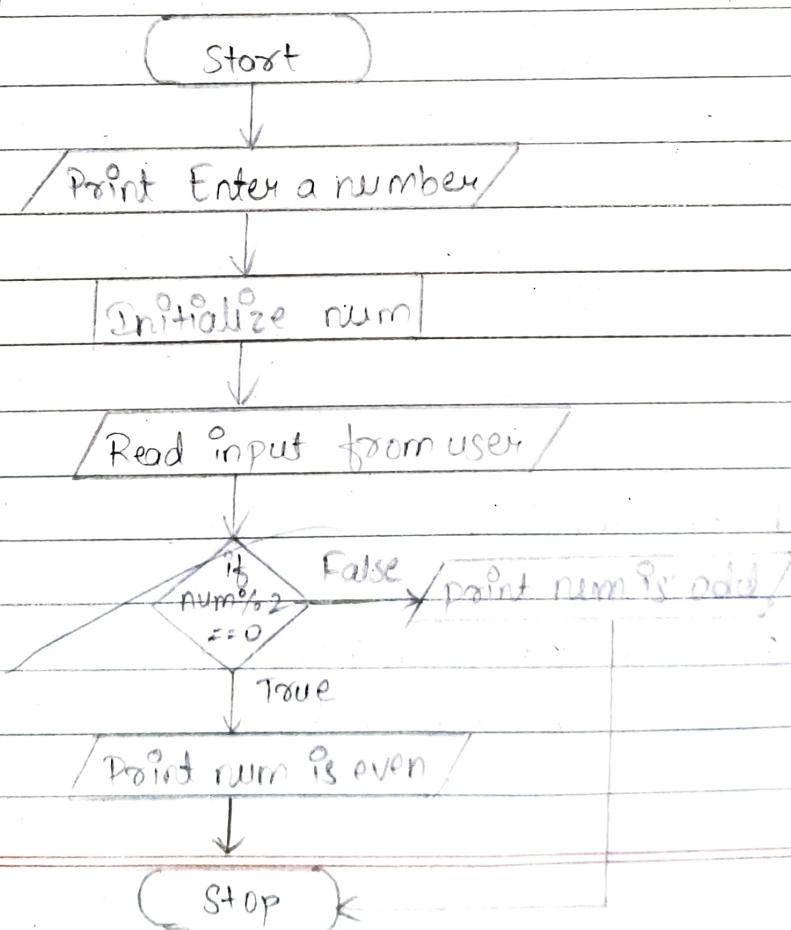
Step 4: Check if  $\text{num} \% 2$  gives remainder 0

Step 5: If true print number is even and goto Step 7 else goto Step 6

Step 6: Print number is odd

Step 7: Stop

### Flowchart



Code

```
import java.util.Scanner;
public class JavaExample
{
    public static void main (String [] args)
    {
        int num;
        System.out.print("Enter an Integer number:");
        Scanner sc = new Scanner (System.in);
        num = sc.nextInt();
        if (num % 2 == 0)
            System.out.println (num + " is an even no");
        else
            System.out.println (num + " is an odd number");
    }
}
```

Output:

Aksh Akshara Singh

1BM22CS029

Enter an Integer number: 29

29 is an odd number

```
C:\Users\bmsce\Desktop\1bm22cs029>javac JavaExample.java
```

```
C:\Users\bmsce\Desktop\1bm22cs029>java JavaExample
```

Akshara Singa

1BM22CS029

Enter an Integer number:29

29 is an odd number.

3) Java program to print right triangle star pattern with eight rows

### Algorithm

Step 1: Start

Step 2: Initialize row, column and numberofrows = 8

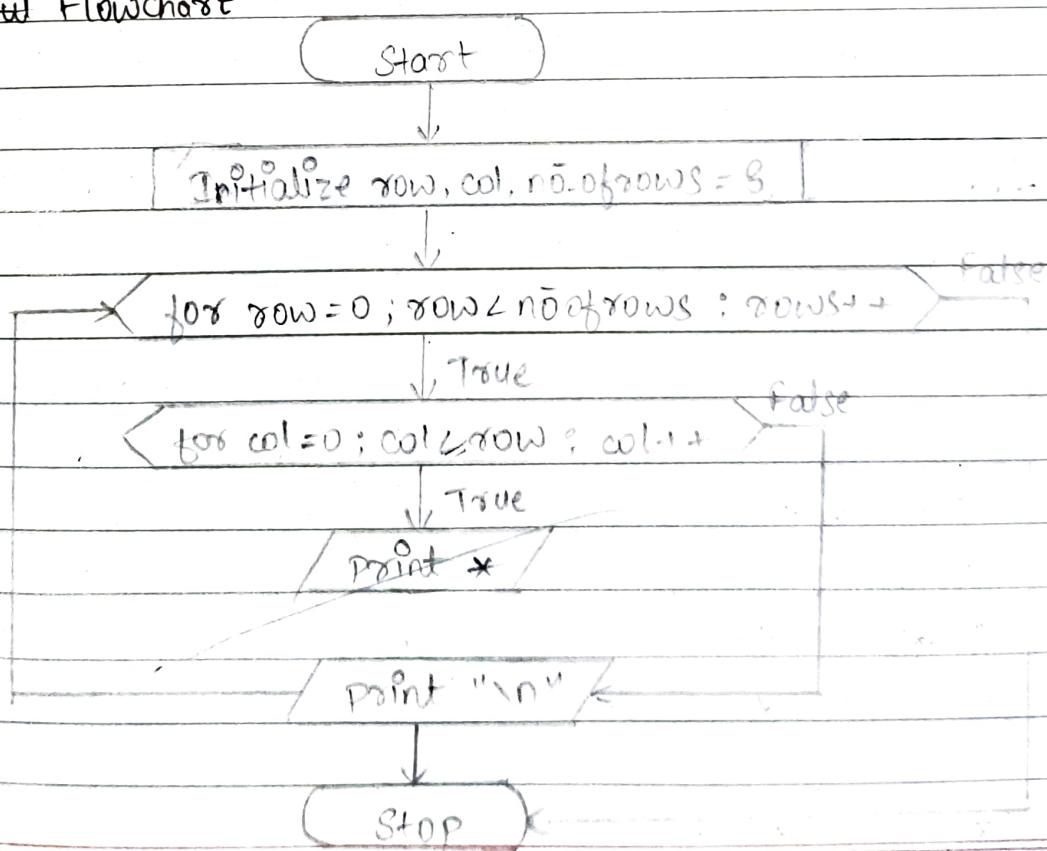
Step 3: for row less than numberofrows increase row by one else goto Step 6

Step 4: for column less than numberofrows increase column by one else goto step 3 and print newline

Step 5: Print \*

Step 6: Stop

### Q4 Flowchart



Code

```
public class RightTriangle
```

```
{
```

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

```
{
```

```
int row, column, numberofrows = 8;
```

```
for (row = 0; row < numberofrows; row++)
```

```
{
```

```
for (column = 0; column < numberofrows; column++)
```

```
{
```

```
System.out.print ("*");
```

```
g
```

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

```
g
```

```
{
```

Output

\*

\* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

```
C:\Users\bmsce\Desktop\1bm22cs029>javac RightTriangle.java
```

```
C:\Users\bmsce\Desktop\1bm22cs029>java RightTriangle
```

Akshara Singa

1BM22CS029

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

4) Java program to find quotient and remainder of 15 and 2

Algorithm

Step 1 : Start

Step 2 : Initialize num1 = 15 and num2 = 2

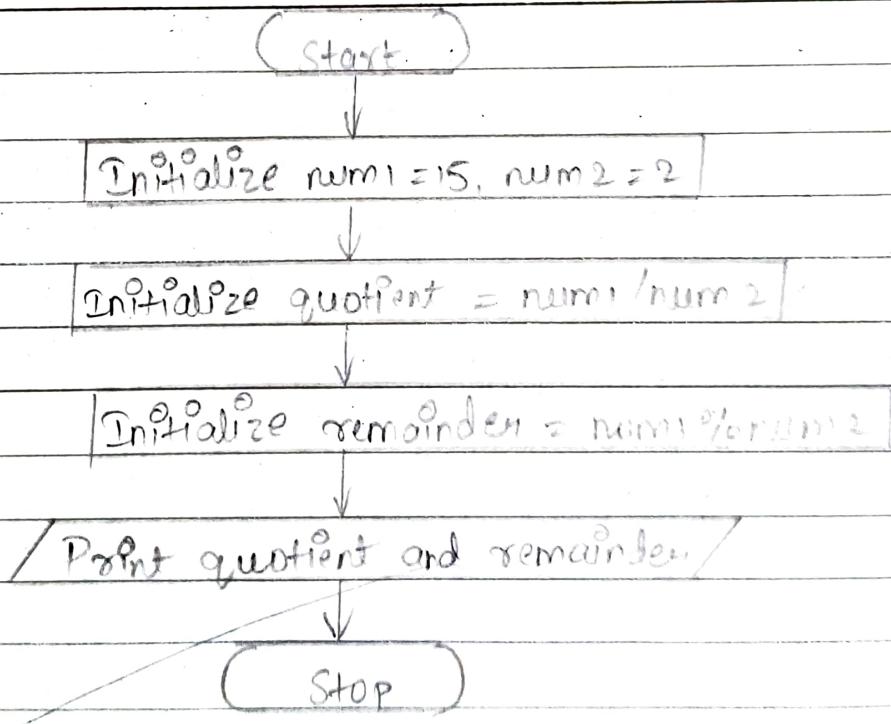
Step 3 : Set quotient = num1 / num2

Step 4 : Set remainder = num1 % num2

Step 5 : Print quotient and remainder

Step 6 : Stop

Flowchart



Code

```
public class QuotientAndRemainder  
{
```

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

```
        int num1 = 15, num2 = 2;
```

```
        int quotient = num1 / num2;
```

```
        int remainder = num1 % num2;
```

```
        System.out.println ("Quotient is: " + quotient);
```

```
        System.out.println ("Remainder is: " + remainder);
```

```
}
```

```
}
```

Output

Quotient is : 7

Remainder is : 1

```
C:\Users\bmsce\Desktop\1bm22cs029>javac QuotientAndRemainder.java
```

```
C:\Users\bmsce\Desktop\1bm22cs029>java QuotientAndRemainder
```

```
Quotient is: 7
```

```
Remainder is: 1
```

```
Akshara Singa
```

```
1BM222CS029
```

## 5) Java Program to multiply two numbers

Algorithm

Step 1 : Start

Step 2 : Print Enter first number

Step 3 : Set entered number to num 1

Step 4 : Print Entered second number

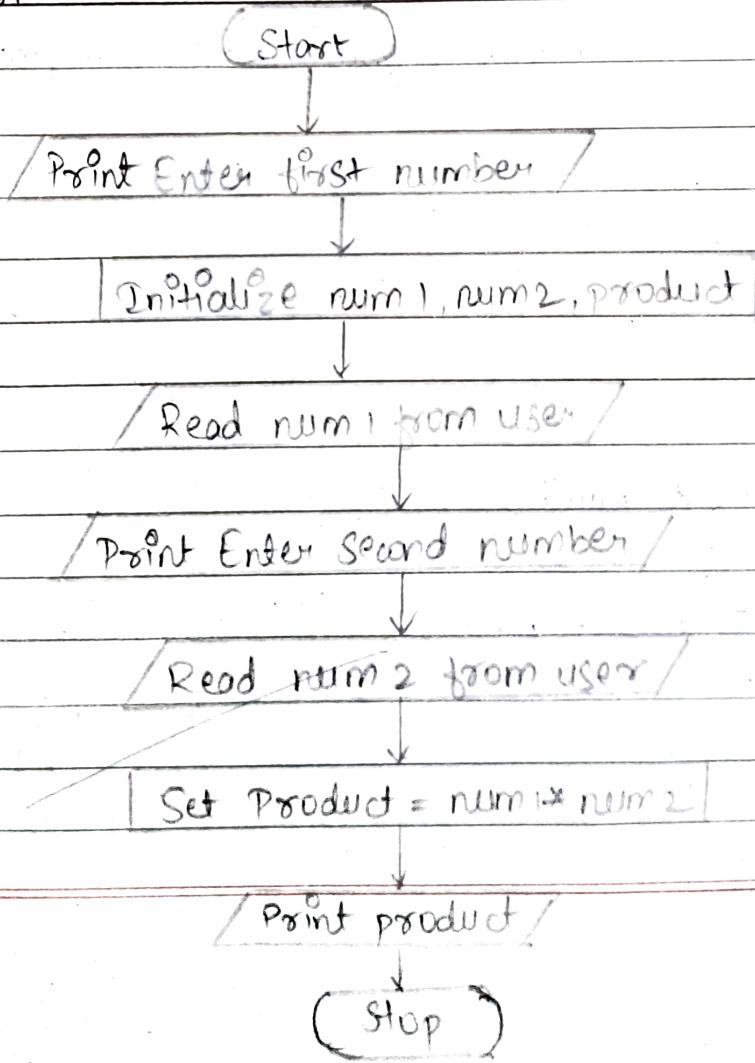
Step 5 : Set entered number to num 2

Step 6 : Calculate <sup>and set</sup> product = num1 \* num2

Step 7 : Print product

Step 8 : Stop

Flowchart



Code

```
import java.util.Scanner;  
public class Multiplication  
{  
    public static void main (String [] args)  
    {  
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter first number:");  
        int num1 = sc.nextInt();  
        System.out.println ("Enter second number:");  
        int num2 = sc.nextInt();  
        sc.close();  
        int product = num1 * num2;  
        System.out.println ("Output: " + product);  
    }  
}
```

Output

Enter first number:

2

Enter second number:

3

Output : 6

```
C:\Users\bmsce\Desktop\1BM22CS029>javac Multiplication.java
```

```
C:\Users\bmsce\Desktop\1BM22CS029>java Multiplication
```

```
Enter first number:
```

```
2
```

```
Enter second number:
```

```
3
```

```
Output: 6
```

```
Akshara Singa
```

```
1BM22CS029
```

6) Swap the floating point numbers 1.2 and 2.45 using a temporary variable

### Algorithm

Step 1 : Start

Step 2 : Initialize num1 = 1.2, num2 = 2.45 , temp

Step 3 : Set temp = num1

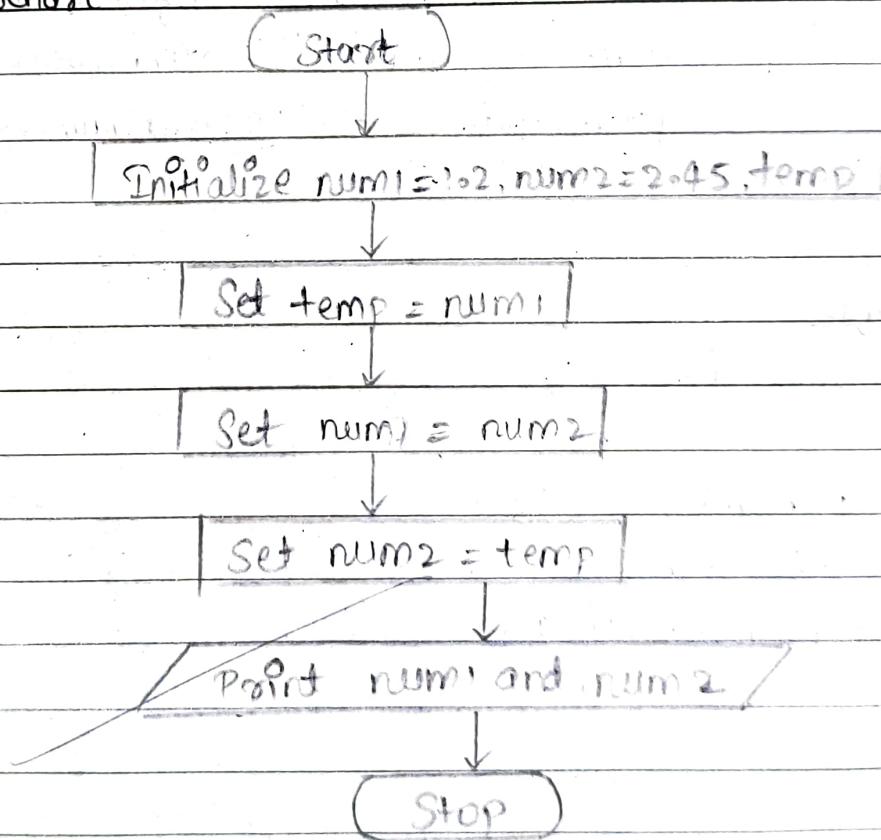
Step 4 : Set num1 = num2

Step 5 : Set num2 = temp num1 and then num1 = temp

Step 6 : Print num1 and num2

Step 7 : Stop

### Flowchart



Code

public class Swapnumbers

{

public static void main (String [ ] args)  
{

float first = 1.20f , second = 2.45f;

System.out.println ("-- Before Swap --");

System.out.println ("First number = " + first);

System.out.println ("Second number = " + second);

float temporary = first;

first = second;

second = temporary;

System.out.println ("-- After Swap --");

System.out.println ("First number = " + first);

System.out.println ("Second number = " + second);

}

3

Output

-- Before Swap --

First number = 1.2

Second number = 2.45

-- After Swap --

First number = 2.45

Second number = 1.2

```
C:\Users\bmsce\Desktop\1bm22cs029>javac SwapNumbers.java
```

```
C:\Users\bmsce\Desktop\1bm22cs029>java SwapNumbers
```

--Before swap--

First number= 1.2

Second number= 2.45

--After swap--

First number = 2.45

Second number= 1.2

Akshara Singa

1BM22CS029

7) Develop a Java program that prints all real solutions to the quadratic equation  $ax^2 + bx + c = 0$ . Read in  $a, b, c$  and use the quadratic formula. If the discriminant  $b^2 - 4ac$  is negative, display a message stating that there are no real solutions.

### Algorithm

Step 1 : Start

Step 2 : Initialize float  $a, b, c, d$  and double  $x_1, x_2$ ;

Step 3 : Print enter the coefficients

Step 4 : Read the values of  $a, b, c$

Step 5 : Calculate  $d$  value where  $d = b^2 - 4ac$

Step 6 : if  $d > 0$

calculate  $x_1$  and  $x_2$ , print  $x_1, x_2$

Step 7 : else if  $d = 0$

calculate  $x_1 = x_2 = -b/2a$  print  $x_1, x_2$

Step 8 : else

print roots are imaginary

calculate  $x_1 = -b/2a$

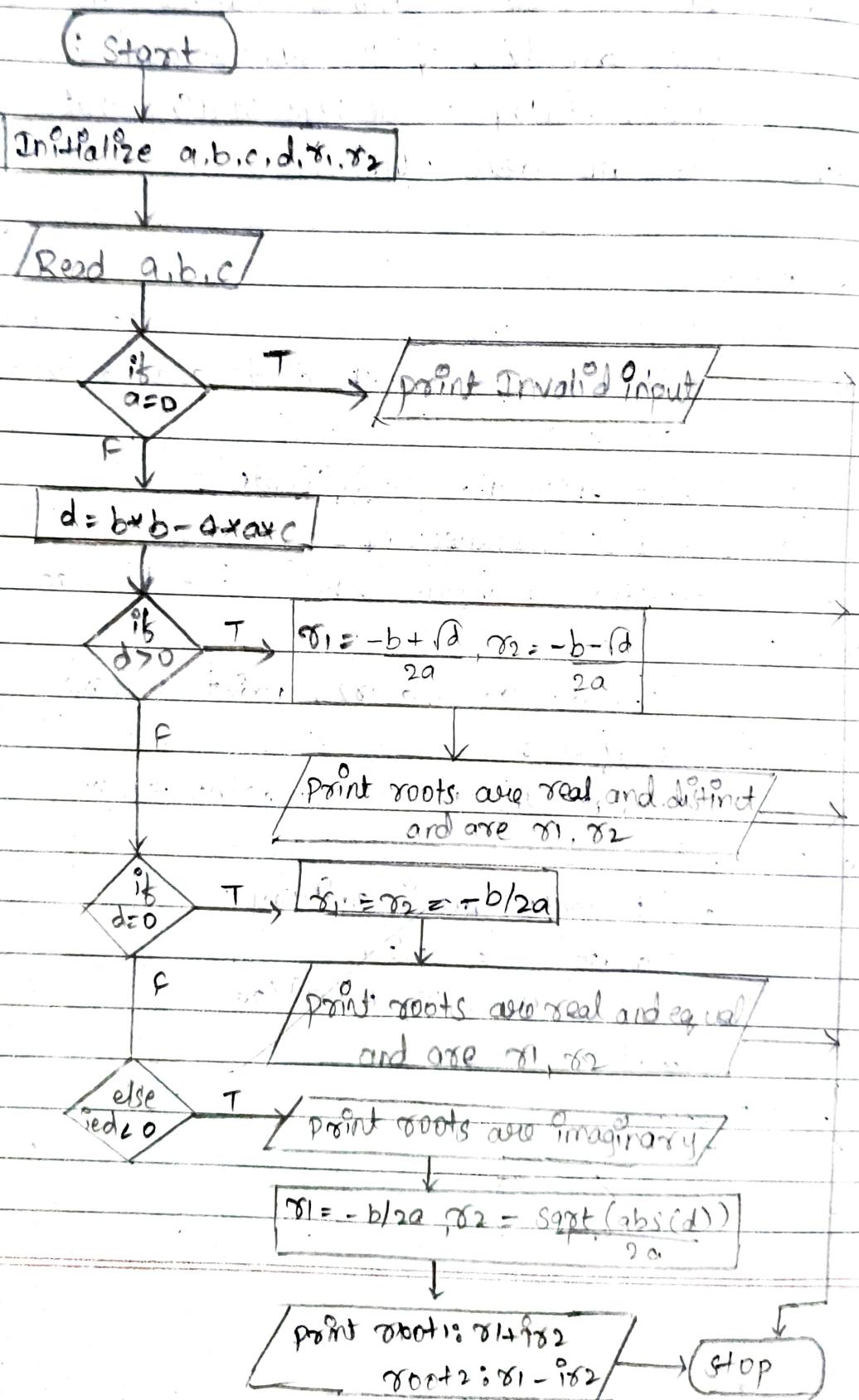
$x_2 = \text{Sqrt}(d)/2a$

print  $x_1$  and  $x_2$

Step 9 : Stop

## Algorithm

### Flowchart



```

import java.util.Scanner;
code import static java.lang.Math.sqrt;
public class QuadraticEquation {
    public static void main (String [ ] args) {
        float a, b, c, d;
        double x1, x2;
        Scanner scan = new Scanner (System.in);
        System.out.println ("Enter coefficients");
        a = Scan.nextFloat ();
        b = Scan.nextFloat ();
        c = Scan.nextFloat ();
        if (a == 0)
            System.out.println ("Invalid input");
        else {
            d = b * b - 4 * a * c;
            if (d > 0)
                System.out.println ("Roots are real & distinct");
            x1 = (-b + Math.sqrt (d)) / (2 * a);
            x2 = (-b - Math.sqrt (d)) / (2 * a);
            System.out.println ("root 1: " + x1 + " root 2: " + x2);
            else if (d == 0)
                System.out.println ("Roots are real and equal");
            x1 = x2 = -b / (2 * a);
            System.out.println ("root1: " + x1 + " root2: " + x2);
            else
                System.out.println ("Roots are Imaginary");
            x1 = -b / 2a;
            x2 = sq (abs) sqrt (d) / (2a);
            System.out.println ("root1: " + x1 + " + " + x2 + "i" + "root2: "
                + x1 + " - " + x2 + "i");
        }
    }
}

```

Output

① enter coefficients

$$2 \quad 1 \quad 1$$

Roots are imaginary

$$\text{root 1: } -0.25 + i0.6614378277$$

$$\text{root 2: } -0.25 - i0.6614378277$$

② enter coefficients

$$1 \quad 2 \quad 1$$

Roots are real and equal

$$\text{root 1: } -1.0$$

$$\text{root 2: } -1.0$$

③ enter coefficients

$$1 \quad 3 \quad 1$$

Roots are real and distinct

$$\text{root 1: } -0.38196601125$$

$$\text{root 2: } -2.6180339887$$

④ enter coefficients

~~$$0 \quad 1 \quad 1$$~~

Invalid quadratic equation

Feb 22/2/23

```
C:\Users\STUDENT\Desktop\1bm22cs029>javac QE.java
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>java QE
```

```
Akshara 1BM22CS029  
enter coefficients
```

```
2
```

```
1
```

```
1
```

```
Roots are imaginary
```

```
root1:-0.25+i0.6614378277661477
```

```
root2:-0.25-i0.6614378277661477
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>javac QE.java
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>java QE
```

```
Akshara 1BM22CS029  
enter coefficients
```

```
1
```

```
2
```

```
1
```

```
Roots are real and equal
```

```
root1:-1.0
```

```
root2:-1.0
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>javac QE.java
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>java QE
```

```
Akshara 1BM22CS029  
enter coefficients
```

```
1
```

```
3
```

```
1
```

```
Roots are real and distinct
```

```
root1:-0.3819660112501051
```

```
root2:-2.618033988749895
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>javac QE.java
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>java QE
```

```
Akshara 1BM22CS029  
enter coefficients
```

```
0
```

```
1
```

```
1
```

```
Invalid quadratic equation
```

Q Calculate the SGPA of a given person by creating a method to accept values and calculate SGPA, Also display values

Import java.util.Scanner;

public class Student

{

String usn;

String name;

private static int credit[] = {4, 4, 4, 3, 3, 2, 1, 1}

int marks[] = new int[8];

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

~~public void getDetails()~~

~~{~~

~~System.out.println("Enter your USN:");~~

~~USN = s.next();~~

~~System.out.println("Enter your name:");~~

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

~~{~~

~~public void setMarks()~~

~~{~~

~~System.out.println("Enter your marks")~~

~~In order considering 91-100 as 10, 81 to 90 as 9);~~

~~for (int i=0; i<8; i++)~~

~~{~~

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

~~}~~

```
for (int i=0; i<8; i++)
```

```
{  
    public double sgpa ()
```

```
    double Sgpa=0, temp=0;
```

```
    for (int i=0; i<8; i++)
```

```
{
```

```
    temp = credit[i] * marks[i];
```

```
}
```

```
    Sgpa = temp / 22;
```

```
} // & return sgpa;
```

```
public void display()
```

```
{
```

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

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

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

```
}
```

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

```
{
```

```
Student s1 = new Student ();
```

```
s1.getDetails ();
```

```
s1.setMarks ();
```

```
s1.display ();
```

```
}
```

## Algorithm.

start

- 1) generate a class student with variables usn, name and credits array with values { 9, 4, 93, 3, 2, 1, 1 } and marks array,
- 2) receive details name and usn from user in the get-details function
- 3) create set-marks function to set the marks from user by accepting values one after the other
- 4) create function Sgpa where create temporary variable temp which stores the value of credits \* marks for every individual subject
- 5) calculate sgpa by dividing total credit with temp
- 6) generate a display function which will display name, usn and sgpa
- 7) Inside the main function create an object of class student call the functions get-details, set-marks and display
- 8) stop

## Algorithm

( Start

Output

enter your usn

IBM22CS029

enter name

AKSHARA

enter your marks % is same order as credits

100

100

97

89

96

95

87

99

Name: AKSHARA

Usn : IBM22CS029

Sgpa : 9.81818

29/12/23  
Akshara

```
C:\Users\STUDENT\Desktop\1bm22cs029>javac Student.java
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>java Student
```

```
enter your usn
```

```
1BM22CS029
```

```
enter name:
```

```
AKSHARA
```

```
enter your marks in the same order as credits
```

```
100
```

```
100
```

```
97
```

```
89
```

```
96
```

```
95
```

```
87
```

```
99
```

```
Name:AKSHARA
```

```
Usn:1BM22CS029
```

```
Sgpa:9.818181818181818
```

s Create a class Book which contains 4 members, book name, author, price, num-pages. Include a constructor to set the values of the members. Include methods to set and get the details of the objects. Include a testing method that could display the complete details of the book. Develop a java program to create n book objects.

- import java.util.Scanner;

class book

{

String name;

String author;

float price;

int num-pages;

void get details()

{

Scanner sc = new Scanner(System.in);

System.out.println("enter bookname, author, price, num-pages");

name = sc.next();

author = sc.next();

price = sc.nextFloat();

num-pages = sc.nextInt();

}

void getDetails()

{

String details = toString();

System.out.println(details);

}

public String toString()

{

return "the book" + name + "was  
written by " + author + "it consists of "  
+ numPages + " pages and costs around " + price;

}

public static void main(String[] args)

{

Scanner scan = new Scanner(System.in);  
System.out.println("enter no of books  
you want to generate");

int n = scan.nextInt();

book b[] = new book[n];

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

{

b[i] = new book();

b[i].setDetails();

{

System.out.println("book details");

System.out.println();

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

{

b[i].getDetails();

{

}

## Algorithm

~~Start~~

- Step 1: Create a class book with members name, author, price and number of pages
- Step 2: get the details of the book through the method set-details , accept values from the user
- Step 3: display the details of the book using to string method
- Step 4: create a to string method which would return the details of the book in one string
- Step 5: create the main class method now
- Step 6: ask the user for the number of books to be entered .
- Step 7: Create an array of type book consisting n books
- Step 8: for i ranging from 0 to n initialize all the n books and set-details of all the books individually.
- Step 9: Now for all books print the details of every book.
- Step 10: Stop

Akshara Singa 1BM22CS029

enter bookname,author,price,num\_pages

Python

ABCDEF

234

567

enter bookname,author,price,num\_pages

JAVA

XYZ

565

567

the book Java was written by Strange it consists of 9857 pages and costs 243.0rupees.

the book Python was written by ABCDEF it consists of 567 pages and costs 234.0rupees.

the book JAVA was written by XYZ it consists of 567 pages and costs 565.0rupees.

Q Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide 3 classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

→ import java.util.Scanner;

abstract class Shape :

{

int x, y;

abstract void area();

public static void main (String args[])

{

Shape obj1 = new Circle();

obj1.area();

Shape obj2 = new Triangle();

obj2.area();

Shape obj3 = new Rectangle();

obj3.area();

}

}

class Rectangle extends Shape

{ System.out.println ("Enter length and breadth of the rectangle");

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

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

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

```
void public void area()
```

```
{ System.out.println ("The area of  
rectangle is , x*y); }
```

```
}
```

```
class Circle extends Shape
```

```
{ System.out.println ("Enter radius of  
the circle"); }
```

~~```
Scanner sc = new Scanner (System. in)
```~~~~```
x = sc.nextInt();
```~~~~```
y = x;
```~~

```
public void area()
```

```
{ System.out.println ("The area of  
circle is 3.14*x*y); }
```

```
}
```

```
class Triangle extends Shape
```

~~```
{ System.out.println ("Enter base and  
height of the Triangle"); }
```~~~~```
Scanner sc = new Scanner (System. in)
```~~~~```
x = sc.nextInt();
```~~~~```
y = sc.nextInt();
```~~

```
public void area()
{
    System.out.println("The area of
    Triangle is 0.5*x*y");
}
```

### Algorithm

Step 1: Start

Step 2: Create an abstract class Shape

Step 3: Initialize two integers

Step 4: Initialize an abstract method area

Step 5: create a class rectangle extending  
class Shape

Step 6: Initialize value of x and y to z.

Step 7: construct method area returning ~~area~~

Step 8: Create a class circle extending  
class Rectangle

Step 9: Initialize value of x and y to l and b

Step 10: construct method area returning ~~area~~

Step 11: Create a class triangle

Step 12: Initialize value of x and y to  
b and h

Step 13: print area =  $\frac{1}{2}bh$

Step 14: Stop

Final Output

Akshara 1BM22CS029

enter the radius of the circle

1

area of circle is 3.14

enter the length and breadth of the rectangle

2

2

area of rectangle is 4

enter the base and height of the triangle

2

2

area of triangle is 2.0

Q Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides CI and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a min balance and if the balance falls below this a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements.

Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customer and update the balance
- b) Display the balance
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance Check for the min balance, impose penalty if necessary and update the balance

```
- import java.util.Scanner;  
class Account {  
    String customerName;  
    long accno;  
    String accountType;  
    double balance;  
    public Account (String CustomerName,  
    long accno, String accountType)  
    { this.accno = accno;  
        this.accountType = accountType;  
        this.balance = 0.0;  
    }  
    public void displayBalance()  
    { System.out.println ("Account Number : " + accno);  
        System.out.println ("Customer Name : " + customerName);  
        System.out.println ("Account Type : " + accountType);  
        System.out.println ("Balance : $" + balance);  
    }  
}  
class Curr_Acc extends Account {  
    double minBalance;  
    double serviceCharge;  
    public Curr_Acc (String customerName, long accno)  
    { super (customerName, accno, "Current");  
        this.minBalance = 500;.  
        this.serviceCharge = 50;  
    }  
}
```

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

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

```
{ balance -= amount;
```

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

```
Current Balance: $ " + balance); }
```

```
else
```

```
{ System.out.println ("Insufficient funds  
Withdrawal not allowed."); }
```

```
}
```

```
}
```

```
public void imposeServiceCharge ()
```

```
{ if (balance < minBalance)
```

```
{ balance -= serviceCharge;
```

```
System.out.println ("Service charge
```

```
imposed. Current Balance: RS " + balance);
```

```
}
```

```
}
```

```
class Sav-Acc extends Account
```

```
{ double interestRate;
```

```
public Sav-Acc (String customerName, long accno)
```

```
{ super (customerName, accno, "Savings")
```

```
this.interestRate = 0.05;
```

```
}
```

```
compound
```

```
public void depositInterest (double initialAmount, int term)
```

```
{ double interest = balance * interestRate;
```

```
balance += interest;
```

```
double compoundInterest = initialAmount * Math.pow ((1 +
```

```
interestRate), term) - initialAmount;
```

```
balance += compoundInterest;
```

System.out.println ("Interest deposited.  
Current Balance : RS." + balance);

3

8

public class Bank {

    public static void main (String [] args)

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

        System.out.println ("Choose account type:");

        System.out.println ("1. Current");

        System.out.println ("2. Savings");

        System.out.print ("Enter choice (1 or 2):");

        int choice = scanner.nextInt();

        System.out.print ("Enter customer name:");

        String customerName = scanner.next();

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

        long accno = scanner.nextLong();

        if (choice == 1)

        { CurrAcct currAccount = new CurrAcct

            (customerName, accno);

            System.out.print ("Enter initial balance: \$");

            double initialBalance = scanner.nextDouble();

            currAccount.balance = initialBalance;

            System.out.print ("Enter withdrawal amount: \$");

            double withdrawalAmount = scanner.nextDouble();

            currAccount.withdraw (withdrawalAmount);

```
    curAccount. imposeServiceCharge();
```

```
    curAccount. displayBalance();
```

```
} else if (choice == 2)
```

```
{ SavAccount savAccount = new SavAcct(customerName,  
   accno);
```

```
System.out.print("Enter Initial balance: $");
```

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

```
savAccount.balance = initialBalance;
```

```
System.out.print("Enter withdrawal amount: $");
```

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

```
savAccount.balance -= withdrawalAmount;
```

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

```
Current Balance: $" + savAccount.balance);
```

```
System.out.print("Enter interest rate: ");
```

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

```
savAccount.interestRate = interestRate;
```

```
savAccount.displayBalance();
```

```
System.out.print("Enter term (in years) for  
Compound Interest calculation: ");
```

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

```
savAccount.compoundInterest(initialBalance, term);
```

```
savAccount.displayBalance();
```

```
}
```

```
else {
```

```
System.out.println("Invalid choice");
```

```
}
```

```
}
```

```
}
```

## Algorithm

Step 1: Start

Step 2: Create class account with attribute

Step 3: Assign values to attribute

Step 4: Create displayBalance method

Step 5: Create current account class extending the Account class

Step 6: Create a withdraw method

Step 7: Create a method for imposing penalty

Step 8: Create savings account class extending the Account class

Step 9: Create a method to calculate the Compound Interest where

$$\text{total} = \text{Initial} * \left(1 + \frac{\text{Interest rate}}{100}\right)^{\text{nt}}$$

Step 10: Create a main function which accept the details of the customer and type of account

Step 11: Ask for user the type of account and call for the required class

Step 12: Stop

Proc  
19/1

Output

Choose account type:

- 1. Current
- 2. Savings

Enter choice (1 or 2): 1

Enter customer name: Akshara

Enter account number: 7892858259

Enter initial balance: \$ 10000

Enter withdrawal amount: \$ 2000

Withdrawal successful. Current Balance: \$ 8000.0

Account number: 7892858259

Customer Name: Akshara

Account Type: Current

Balance: \$ 8000.0

Qall

```
1BM22CS029
Akshara Singa
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 1
Enter customer name: Akshara
Enter account number: 7892858259
Enter initial balance: $10000
Enter withdrawal amount: $2000
Withdrawal successful. Current Balance: $8000.0
Account Number: 7892858259
Customer Name: Akshara
Account Type: Current
Balance: $8000.0

C:\Users\STUDENT\Desktop\1bm22cs029>javac Bank.java

C:\Users\STUDENT\Desktop\1bm22cs029>java Bank
1BM22CS029
Akshara Singa
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 2
Enter customer name: Akshara
Enter account number: 7892858259
Enter initial balance: $10000
Enter withdrawal amount: $2000
Withdrawal successful. Current Balance: $8000.0
Enter interest rate: 0.05
Account Number: 7892858259
Customer Name: Akshara
Account Type: Savings
Balance: $8000.0
Enter term (in years) for compound interest calculation: 2
Compound Interest deposited. Current Balance: Rs.9025.0
Account Number: 7892858259
Customer Name: Akshara
Account Type: Savings
Balance: $9025.0
```

Q Create a package CIE which has two classes Student and Internals. The class Personal has members like usn, name, Sem. The class Internals has an array that stores the internal marks scored in 5 courses of the current sem of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in 5 courses of the current Sem of the student.

package CIE;

import java.util.\*;

public class Student

{

    public int sem;

    public String usn;

    public String name;

    public void accept()

    { Scanner scan = new Scanner (System.in)

        System.out.println ("Enter usn, name  
        and Sem.\n");

        usn = scan.nextLine();

        name = scan.nextLine();

        sem = scan.nextInt();

    }

}

```
package CIE;
public class Internals
```

```
{  
    public int pm[] = new int[5];
```

```
package SEE;
import CIE.Student;
public class External extends Student
```

```
{  
    public int sm[] = new int[5];
```

```
?  
import java.util.*;  
import SEE.*;  
import CIE.*;  
public class Final_Marks
```

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

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

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

```
        System.out.println ("Enter n: ");
```

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

```
        SEE.External st[] = new SEE.External[n];
```

```
        CIE.Internals s[] = new CIE.Internals[n];
```

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

```
{  
    st[i] = new SEE.External();
```

```
    s[i] = new CIE.Internals();
```

```
s[i] = new CIEInternals();
```

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

```
st[i].accept();
```

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

```
{
```

```
System.out.println("Enter 3m and  
sm of sub" + (j+1));
```

```
st[i].im[j] = sc.nextInt();
```

```
st[i].sm[j] = sc.nextInt();
```

```
fm[j] = s[i].im[j] + st[i].sm[j];
```

```
}
```

```
System.out.println("final marks of" +  
st[i].name);
```

```
for (int k=0; k<5; k++)
```

```
{
```

```
System.out.println("course" + (k+1) + " = "  
+ fm[k]);
```

```
}
```

```
j
```

```
3
```

## Algorithm:-

- Start
- Create a package CIE
- Create a class Student in this package with members USN, name, sem
- Create another class Internal in the same
- In student class create a get method to accept the values
- Create another package SEE
- Create a class External extending Student
- In this class create an array of size 5 to store the Semester marks
- Inside the main function create an array of final marks of n students in 5 subjects
- Create an array st of type External of size n initially for Internal as s
- Accept the details of all n students  
accept the Internal and external marks for all n students
- Calculate the final marks for each subject and store in final marks array
- Display the total final marks of all the Students combinedly (sum of marks of every ~~of~~ subject of all the students)
- Stop

Output

\* enter no of students: 1

enter details:

Enter sem, usn

2

1BM22CS029

Enter internal and see marks of sub) 99 99

" " sub 2 45 45

" " sub 3 49 49

" " sub 4 47 47

" " sub 5 43 43

" " sub 6 46 46

Final marks of 1BM22CS029

Course 1 = 90

Course 2 = 98

Course 3 = 94

Course 4 = 86

Course 5 = 92

90 98 94 86 92

C:\Users\STUDENT\Desktop\1bm22cs029\ooj>javac -d . finalMarks.java

C:\Users\STUDENT\Desktop\1bm22cs029\ooj>java finalMarks

Akshara

1bm22cs029

enter no of students:

2

Enter details1

Enter sem,usn and name:

2

1bm22cs029

Enter internal and see marks of sub1

45

45

Enter internal and see marks of sub2

49

49

Enter internal and see marks of sub3

47

47

Enter internal and see marks of sub4

43

43

Enter internal and see marks of sub5

46

46

Final marks of 1bm22cs029

Course1=90

Course2=98

Course3=94

Course4=86

Course5=92

course=91

Enter details2

Enter sem,usn and name:

3

1bm22cs027

Enter internal and see marks of sub1

45

45

Enter internal and see marks of sub2

46

46

Enter internal and see marks of sub3

47

47

Enter internal and see marks of sub4

48

48

Enter internal and see marks of sub5

49

49

Final marks of 1bm22cs027

Course1=90

Course2=92

Course3=94

Course4=96

Course5=98

& Create Father class and a Son class extending father class with appropriate exceptions

→ import java.util.Scanner;

class WrongAge extends Exception

{

public WrongAge (String message) {

super (message);

}

}

class Father {

int fatherAge;

public Father (int fatherAge) throws WrongAge

{

if (fatherAge < 0)

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

this.fatherAge = fatherAge;

}

}

class Son extends Father {

int sonAge;

public Son (int fatherAge, int sonAge) throws

WrongAge {

try { super (fatherAge); }

catch (WrongAge a)

{ System.out.println ("Wrong father's age!"); }

};

```
if (sonAge >= fatherAge)
    {
        throw new WrongAge ("Son's age
must be less than Father's age");
    }
    this.sonAge = sonAge;
}
```

```
public class FatherSon {
    public static void main (String [] args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter father's age
and Son's age");
        int fa = sc.nextInt ();
        int sa = sc.nextInt ();
        try {
            Son s = new Son (fa, sa);
            System.out.println ("Father's age : " + s.fatherAge);
            System.out.println ("Son's age : " + s.sonAge);
        }
        catch (WrongAge e) {
            System.out.println ("Error" + e.getMessage ());
        }
    }
}
```

## Algorithm

### Start

- Create wrongage class extending exception class
- Pass the message from this wrongage class to the main exception class
- Create father class and throw this exception when father's age is less than 0
- Create son's class extending father's class check son's age is greater than father's age if yes throw Wrong Age error
- Create main function , enter the user input values for son's and father's age . Check for conditions and print the values of son's and father's age
- Stop

### Output

- Enter father's age and son's age:

22

6

Father's age : 22

Son's age : 6

Q Write a program which creates 2 threads  
one thread displaying "BMS College of  
Engineering" after every <sup>10 sec</sup>  
and the second thread  
displaying "CSE" every 2 seconds



### Algorithm

- Create a class A extending Thread
- Create a constructor A where set  $t_1 = 0$   
and maximum time limit as  $time_1 = 100000$  ms
- Now create a run method which  
prints BMSCE and increase  $t_1$  by 10000
- Create a class B extending Thread
- Create constructor B set  $t_2 = 0$  and  
max time limit as  $time_2 = 20000$  msec
- Now create run method printing CSE  
and increase  $t_2$  by 2000
- Inside the main function create 2  
threads one of each class and start  
the threads
- Stop

class A extending Thread

```
{  
A( ) {  
    int t1 = 0;  
    int time1 = 100000;  
}  
    public void run() {  
        while (t1 <= time1) {  
            System.out.println("BMS College  
of Engineering");  
            try {  
                sleep(10000);  
            }  
            catch (Exception e) {  
                System.out.println("e.getMessage()");  
            }  
            t1 += 10000;  
        }  
    }  
}
```

class B extending Thread

```
{  
B( ) {  
    int t2 = 0;  
    int time2 = 200000;  
}
```

```
public void run()
{
    while (t2 <= time2)
    {
        System.out.println ("CSE");
        t2 += 2000;
        try {
            Thread.sleep (2000);
        } catch (InterruptedException e) {
            System.out.println ("e.getMessage()");
        }
    }
}
```

```
public class Demo
```

```
public static void main (String args[])
{
    new A().start();
    new B().start();
}
```

OUTPUT

BMSCE

CSE

CSE

CSE

CSE

BMSCE

CSE

CSE

CSE

CSE

CSE

BMSCE

BMSCE

BMSCE

BMSCE

BMSCE

BMSCE  
BMSCE  
BMSCE  
BMSCE  
BMSCE

Jai  
16.02.24

C:\Users\STUDENT\Desktop\1bm22cs029>javac demo.java

C:\Users\STUDENT\Desktop\1bm22cs029>java demo

Akshara.Singa

1BM22CS029

BMS COLLEGE OF ENGINEERING

CSE

CSE

CSE

CSE

CSE

BMS COLLEGE OF ENGINEERING

CSE

CSE

CSE

CSE

CSE

BMS COLLEGE OF ENGINEERING

CSE

BMS COLLEGE OF ENGINEERING

```
C:\Users\STUDENT\Desktop\1bm22cs029>java fatherson
Akshara Singa
1BM22CS029
Enter father's age and son's age:
22
6
Father's age: 22
Son's age: 6
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>javac fatherson.java
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>java fatherson
Akshara Singa
1BM22CS029
Enter father's age and son's age:
20
30
Error: Son's age must be less than Father's age
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>javac fatherson.java
```

```
C:\Users\STUDENT\Desktop\1bm22cs029>java fatherson
Akshara Singa
1BM22CS029
Enter father's age and son's age:
-9
2
Error: Age cannot be negative
```

Q Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide Button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberformatException. If Num2 were zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

→ import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class SwingDemo {

SwingDemo() {

JFrame jfrm = new JFrame ("Divide App");

jfrm.setSize(275,150);

jfrm.setLayout(new FlowLayout());

jfrm.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JLabel jlb = new JLabel ("Enter divisor & dividend:");

JTextField ajf = new JTextField(8);

JTextField bjt = new JTextField(8);

JButton button = new JButton ("Calculate");

```
JLabel cur = new JLabel();  
JLabel alab = new JLabel();  
JLabel blab = new JLabel();  
JLabel anslab = new JLabel();
```

```
frm.add(cur);  
frm.add(alab);  
frm.add(ajtf);  
frm.add(bjtf);  
frm.add(button);  
frm.add(alab);  
frm.add(blab);  
frm.add(anslab);
```

```
ActionListener l = new ActionListener() {  
    public void actionPerformed (ActionEvent evt)  
    { System.out.println ("Action event from  
    a text field"); }  
};  
ajtf.addActionListener (l);  
bjtf.addActionListener (l);
```

```
button.addActionListener (new ActionListener () {  
    public void actionPerformed (ActionEvent evt) {  
        try {  
            int a = Integer.parseInt (ajtf.getText());  
            int b = Integer.parseInt (bjtf.getText());  
        } catch (Exception e) {}  
    }  
});
```

```
int ans = a/b;  
alab.setText("In A = " + a);  
blab.setText("In B = " + b);  
anslab.setText("In Ans = " + ans);
```

{

```
catch (NumberFormatException e) {
```

```
alab.setText(" ");  
blab.setText(" ");  
anslab.setText(" ");
```

```
err.setText("Enter Only Integers!");
```

{

```
catch (ArithmaticException e) {
```

```
alab.setText(" ");  
blab.setText(" ");  
anslab.setText(" ");
```

```
err.setText("B should be NonZero!");
```

{

{};

```
form.setVisible(true); }
```

```
public static void main (String args[]) {  
SwingUtilities.invokeLater(new Runnable () {
```

```
public void run () {
```

```
new SwingDemo();
```

{};

{}

Output

Enter the divisor and dividend

|    |   |           |
|----|---|-----------|
| 34 | 2 | calculate |
|----|---|-----------|

$$A = 34 \quad B = 2 \quad \text{Ans} = 17$$

~~Ans~~  
29.02.24



Divider App



Enter the divider and divident:

34

2

Calculate

**A = 34 B = 2 Ans = 17**

C:\Users\aksha\Downloads&gt;cd java

C:\Users\aksha\Downloads\java&gt;javac SwingDemo.java

C:\Users\aksha\Downloads\java&gt;java SwingDemo.java

Akshara Singa

1BM22CS031