

Aim:

Ramesh went to a general Store and picked two items of x and y prices. Write a java program to calculate the total amount for Ramesh to pay.

Sample Test Case:
 Enter x value: 10
 Enter y value: 20
 Sum: 30

Source Code:

q1148/IntExample.java

```
package q1148;
import java.util.Scanner;
public class IntExample {
    public static void main(String[] args) {
        //write your code here
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter x value: ");
        int x=sc.nextInt();
        System.out.print("Enter y value: ");
        int y=sc.nextInt();
        int a=x+y;
        System.out.println("Sum: "+a);
    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter x value:

10

Enter y value:

20

Sum: 30

Test Case - 2**User Output**

Enter x value:

12

Enter y value:

50

Sum: 62

Test Case - 3

User Output

Enter x value:

700

Enter y value:

654

Sum: 1354

Test Case - 4

User Output

Enter x value:

99

Enter y value:

100

Sum: 199

Aim:

A contractor need to build a house for which he has to buy Y number of bricks of price X each. Help him to display total amount required to buy Y number of bricks.

Write a java program based on above Scenario.

Assumptions:

- Price X may be integer or double datatype
- Price Y may be integer or double datatype
- Total amount should be in double datatype

Sample Test Case:

```
Enter price: 10.56
Enter quantity: 50.5
Total amount: 533.28
```

Source Code:

q1149/doubleExample.java

```
package q1149;
import java.util.Scanner;
public class doubleExample {
    public static void main(String[] args) {
        //write your code here
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter price: ");
        double a=sc.nextDouble();
        System.out.print("Enter quantity: ");
        double b=sc.nextDouble();
        double c=(a*b);
        System.out.println("Total amount: "+c);
    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter price:

100

Enter quantity:

20

Total amount: 2000.0

Test Case - 2**User Output**

Enter price:

10.56

Enter quantity:

50.5

Total amount: 533.28

Test Case - 3

User Output

Enter price:

87.5

Enter quantity:

65

Total amount: 5687.5

Test Case - 4

User Output

Enter price:

83.56

Enter quantity:

200.3

Total amount: 16737.068000000003

Aim:

A Class contains 60 Students. Mr. Vinod, class teacher, wants to send a message to the students congratulating them for clearing the final tests.

But Vinod doesn't have time to send messages to each student separately.

Write a program which takes the student name as input and displays Student's name along with the Vinod's message.

Assumptions:

- Student name doesn't contain empty spaces.

Sample Test Case:

```
Enter Student name: Vinod
Hello, Vinod. Congratulations on passing your last exams.
```

Source Code:

`q1152/StringExample.java`

```
package q1152;
import java.util.Scanner;
public class StringExample {
    public static void main(String[] args) {
        //write your code here
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter Student name: ");
        String s=sc.next();
        System.out.println("Hello, "+s+". Congratulations on passing your last
exams.");
    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter Student name:

Shiva

Hello, Shiva. Congratulations on passing your last exams.

Test Case - 2**User Output**

Enter Student name:

Vinod

Hello, Vinod. Congratulations on passing your last exams.

Test Case - 3**User Output**

Enter Student name:

Josh_Putnam

Hello, Josh_Putnam. Congratulations on passing your last exams.

Test Case - 4

User Output

Enter Student name:

Divya

Hello, Divya. Congratulations on passing your last exams.

Aim:

A student named Akash wants to find **sum** and **product** of two numbers. Every time Akash has to perform those tasks for different numbers.

Write a java program to help Akash which takes two numbers and prints sum and product of the given numbers.

Sample Test Case:

```
Enter the num1 value: 10
Enter the num2 value: 20
Sum of 10 and 20 = 30
Product of 10 and 20 = 200
```

Source Code:

[q1144/SumAndProductOfInt.java](#)

```
package q1144;
import java.util.Scanner;
public class SumAndProductOfInt
{
    public static void main (String[] args)
    {
        //write your code here
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the num1 value: ");
        int a=sc.nextInt();
        System.out.print("Enter the num2 value: ");
        int b=sc.nextInt();
        int c=a+b;
        System.out.println("Sum of "+a+" and "+b+" = "+c);
        int d=a*b;
        System.out.println("Product of "+a+" and "+b+" = "+d);
    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter the num1 value:

10

Enter the num2 value:

20

Sum of 10 and 20 = 30

Product of 10 and 20 = 200

Test Case - 2**User Output**

Enter the num1 value:

58

Enter the num2 value:

69

Sum of 58 and 69 = 127

Product of 58 and 69 = 4002

Test Case - 3

User Output

Enter the num1 value:

875

Enter the num2 value:

258

Sum of 875 and 258 = 1133

Product of 875 and 258 = 225750

Test Case - 4

User Output

Enter the num1 value:

235

Enter the num2 value:

548

Sum of 235 and 548 = 783

Product of 235 and 548 = 128780

Aim:

Suresh is a civil engineer. He wants to find the perimeter of the plots which helps him to make blueprints of the land.

Write a java program to find the perimeter of the plots which helps suresh to execute his plans easily.

Sample Test Case:

```
Enter the length of the plot: 100
Enter the width of the plot: 50
Perimeter of the plot is: 300
```

Source Code:

`q1153/RectanglePerimeter.java`

```
package q1153;
import java.util.Scanner;
public class RectanglePerimeter {
    public static void main(String[] args) {
        //write your code here
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the length of the plot: ");
        int a=sc.nextInt();
        System.out.print("Enter the width of the plot: ");
        int b=sc.nextInt();
        int c=2*(a+b);
        System.out.println("Perimeter of the plot is: "+c);
    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter the length of the plot:

100

Enter the width of the plot:

50

Perimeter of the plot is: 300

Test Case - 2**User Output**

Enter the length of the plot:

58

Enter the width of the plot:

44

Perimeter of the plot is: 204

Test Case - 3

User Output

Enter the length of the plot:

99

Enter the width of the plot:

87

Perimeter of the plot is: 372

Test Case - 4

User Output

Enter the length of the plot:

20

Enter the width of the plot:

10

Perimeter of the plot is: 60

Aim:

Raju is feverish. Although the thermometer only displays temperatures in Fahrenheit, his companion Ravi wants to check Raju's body temperature in Celsius.

Write a java program to help ravi which converts temperature from Fahrenheit to Celsius scale.

Use the formula:

$$\text{Celsius} = (\text{Fahrenheit} - 32) * \frac{5}{9}$$

Sample Test Case:

```
Enter temperature in Fahrenheit: 100.5
Temperature in Celsius: 38.055557
```

Source Code:

`q1146/TemperatureConverter.java`

```
package q1146;
import java.util.Scanner;
public class TemperatureConverter {
    public static void main(String[] args) {
        // write your code here
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter temperature in Fahrenheit: ");
        float a=sc.nextFloat();
        float c=(a-32)*5/9;
        System.out.println("Temperature in Celsius: "+c);

    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

```
Enter temperature in Fahrenheit:
```

```
100.5
```

```
Temperature in Celsius: 38.055557
```

Test Case - 2**User Output**

```
Enter temperature in Fahrenheit:
```

```
95.2
```

```
Temperature in Celsius: 35.11111
```

Test Case - 3**User Output**

```
Enter temperature in Fahrenheit:
```

```
99
```

Temperature in Celsius: 37.22222

Test Case - 4

User Output

Enter temperature in Fahrenheit:

96.58

Temperature in Celsius: 35.87778

Aim:

Ganesh owns a paper plant and recently received a large order to make paper plates with various radii. Ganesh needs to compute the area of every plate with a different radius each time.

Create a Java programme that calculates the area of the circular plates using the radius to aid Ganesh.

Note: Take **pivalue as 3.14**

Sample Test Case:

```
Enter the radius: 20
Area: 1256.0
```

Source Code:

[q1154/AreaOfCircle.java](#)

```
package q1154;
import java.util.Scanner;
public class AreaOfCircle {
    public static void main(String[] args) {
        //write your code here
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the radius: ");
        double a=sc.nextDouble();
        double pi=3.14;
        double area=pi*a*a;
        System.out.println("Area: "+area);
    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter the radius:

20

Area: 1256.0

Test Case - 2**User Output**

Enter the radius:

50

Area: 7850.0

Test Case - 3**User Output**

Enter the radius:

5

Area: 78.5

Test Case - 4

User Output

Enter the radius:

6

Area: 113.03999999999999

Aim:

Write a java program to Convert **int** to following datatypes

- int to long
- long to float
- float to double
- Print all the converted values.

Sample Test Case:

```
Enter an Integer: 78
After widening values are:
Long value: 78
Float value: 78.0
Double value: 78.0
```

Source Code:

[q1140/TypeConversion.java](#)

```
package q1140;
import java.util.Scanner;
public class TypeConversion {
    public static void main(String[] args) {
        //write your code here
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter an Integer: ");
        int a=sc.nextInt();
        System.out.println("After widening values are:");
        long b=(long)(a);
        System.out.println("Long value: "+b);
        float c=(float)(a);
        System.out.println("Float value: "+c);
        double d=(double)(a);
        System.out.println("Double value: "+d);
    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter an Integer:

78

After widening values are:

Long value: 78

Float value: 78.0

Double value: 78.0

Test Case - 2**User Output**

Enter an Integer:

-123

After widening values are:

Long value: -123

Float value: -123.0

Double value: -123.0

Test Case - 3

User Output

Enter an Integer:

-4

After widening values are:

Long value: -4

Float value: -4.0

Double value: -4.0

Test Case - 4

User Output

Enter an Integer:

265

After widening values are:

Long value: 265

Float value: 265.0

Double value: 265.0

Aim:

Imagine having a stock trading application that only shows double datatype stock prices. Users occasionally request estimated stock prices in the form of float, long, and int datatypes.

Write a java program which takes the stock price and converts the price to display approximately (Observe the Sample Test Case).

Sample Test Case:

```
Enter the number: 126.321
After narrowing values are:
Float value: 126.321
Long value: 126
Int value: 126
```

Source Code:

[q1141/ExplicitTypeConversion.java](#)

```
package q1141;
import java.util.Scanner;
public class ExplicitTypeConversion
{
    public static void main(String[] args)
    {
        //write your code here
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the number: ");
        double a=sc.nextDouble();
        System.out.println("After narrowing values are:");
        float b=(float)(a);
        System.out.println("Float value: "+b);
        long c=(long)(a);
        System.out.println("Long value: "+c);
        int d=(int)(a);
        System.out.println("Int value: "+d);

    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter the number:

126.321

After narrowing values are:

Float value: 126.321

Long value: 126

Int value: 126

Test Case - 2

User Output

Enter the number:

23.54

After narrowing values are:

Float value: 23.54

Long value: 23

Int value: 23

Test Case - 3

User Output

Enter the number:

89.546

After narrowing values are:

Float value: 89.546

Long value: 89

Int value: 89

Test Case - 4

User Output

Enter the number:

2.31

After narrowing values are:

Float value: 2.31

Long value: 2

Int value: 2

Aim:

You are working on a software project that involves handling user input for various calculations. The user enters numeric values as strings, and you need to convert these strings to decimals for the calculations.

Note: Use **Double.parseDouble(String)** to convert String to double.

Sample Test Case:

```
Enter a double value: 14.36
Double value entered: 14.36
```

Source Code:

q1157/TypeCasting.java

```
package q1157;
import java.util.Scanner;
public class TypeCasting{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        //write your code here
        System.out.print("Enter a double value: ");
        String b=scanner.nextLine();
        double a=Double.parseDouble(b);
        System.out.println("Double value entered: "+a);

    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

```
Enter a double value:
```

```
14.36
```

```
Double value entered: 14.36
```

Test Case - 2**User Output**

```
Enter a double value:
```

```
32.65
```

```
Double value entered: 32.65
```

Test Case - 3**User Output**

```
Enter a double value:
```

```
46.85
```

```
Double value entered: 46.85
```

Test Case - 4

User Output

Enter a double value:

1.254

Double value entered: 1.254

Aim:

You are developing software for an inventory management system in a retail store. Part of the system involves entering and processing product prices, which are represented as decimal values. To optimise storage and facilitate calculations, Write a program to convert these prices to integers, rounding them to the nearest cent.

Sample Test case:

```
Enter a double value: 12.35
Original Value before Casting: 12.35
After Type Casting to short: 12
After Type Casting to int: 12
```

Source Code:

q1159/NarrowingTypeCasting.java

```
package q1159;
import java.util.Scanner;
public class NarrowingTypeCasting {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a double value: ");
        double inputDouble = scanner.nextDouble();

        // write your code here
        System.out.println("Original Value before Casting: "+inputDouble);
        short b=(short)(inputDouble);
        System.out.println("After Type Casting to short: "+b);
        int a=(int)(inputDouble);
        System.out.println("After Type Casting to int: "+a);

    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter a double value:

12.35

Original Value before Casting: 12.35

After Type Casting to short: 12

After Type Casting to int: 12

Test Case - 2**User Output**

Enter a double value:

45.123

Original Value before Casting: 45.123

After Type Casting to short: 45

After Type Casting to int: 45

Test Case - 3

User Output

Enter a double value:

3.21

Original Value before Casting: 3.21

After Type Casting to short: 3

After Type Casting to int: 3

Test Case - 4

User Output

Enter a double value:

100.23

Original Value before Casting: 100.23

After Type Casting to short: 100

After Type Casting to int: 100

Aim:

You are working on a software project for an international language education program. The project involves creating a program that allows users to enter keyboard shortcuts for various language characters and symbols. These shortcuts are represented as **short** data types, and you need to convert them to **char** data types to display the corresponding characters.

Sample Test Case:

```
Enter a short value: 65
Original short value: 65
Converted char value: A
```

Source Code:

[q1161/ShortToCharConversion.java](#)

```
package q1161;
import java.util.Scanner;

public class ShortToCharConversion {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        //write your code here
        System.out.print("Enter a short value: ");
        short a=scanner.nextShort();
        System.out.println("Original short value: "+a);
        char b=(char)(a);
        System.out.println("Converted char value: "+b);

    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter a short value:

65

Original short value: 65

Converted char value: A

Test Case - 2**User Output**

Enter a short value:

67

Original short value: 67

Converted char value: C

Test Case - 3

User Output

Enter a short value:

78

Original short value: 78

Converted char value: N

Test Case - 4**User Output**

Enter a short value:

100

Original short value: 100

Converted char value: d

Aim:

A task is assigned to a student,in which the student has to categorise the metal blocks in different boxes based on their weight.He can weigh only two metal blocks, at a time.

Write a program which takes weights of two metal boxes and displays a list which includes all relations between the weights.(Observe the sample test case).

Sample Test Case:

```
Enter the weight of object1: 50.5
Enter the weight of object2: 30.6
Relation between object1 and object2:
object1 == object2 = false
object1 != object2 = true
object1 > object2 = true
object1 < object2 = false
object1 >= object2 = true
object1 <= object2 = false
```

Source Code:

[q1143/RelationalOperations.java](#)

```
package q1143;
import java.util.Scanner;
public class RelationalOperations{
public static void main(String []args){
    // write your code below
    Scanner sc=new Scanner(System.in);
    System.out.print("Enter the weight of object1: ");
    double object1=sc.nextDouble();
    System.out.print("Enter the weight of object2: ");
    double object2=sc.nextDouble();
    System.out.println("Relation between object1 and object2:");
    System.out.println("object1 == object2 = "+(object1==object2));
    System.out.println("object1 != object2 = "+(object1!=object2));
    System.out.println("object1 > object2 = "+(object1>object2));
    System.out.println("object1 < object2 = "+(object1<object2));
    System.out.println("object1 >= object2 = "+(object1>=object2));
    System.out.println("object1 <= object2 = "+(object1<=object2));
}
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the weight of object1:
50.5
Enter the weight of object2:
30.6
Relation between object1 and object2:

```
object1 == object2 = false
object1 != object2 = true
object1 > object2 = true
object1 < object2 = false
object1 >= object2 = true
object1 <= object2 = false
```

Test Case - 2

User Output

Enter the weight of object1:

20.896

Enter the weight of object2:

54.69

Relation between object1 and object2:

```
object1 == object2 = false
object1 != object2 = true
object1 > object2 = false
object1 < object2 = true
object1 >= object2 = false
object1 <= object2 = true
```

Test Case - 3

User Output

Enter the weight of object1:

55.5

Enter the weight of object2:

55.5

Relation between object1 and object2:

```
object1 == object2 = true
object1 != object2 = false
object1 > object2 = false
object1 < object2 = false
object1 >= object2 = true
object1 <= object2 = true
```

Test Case - 4

User Output

Enter the weight of object1:

452.369

Enter the weight of object2:

500.258

Relation between object1 and object2:

```
object1 == object2 = false
object1 != object2 = true
object1 > object2 = false
object1 < object2 = true
```

```
object1 >= object2 = false  
object1 <= object2 = true
```

Aim:

Raju is a farmer. He wants to take a loan from a bank that offers a low Simple Interest Rate on a certain amount of capital. Raju needs help in calculating S.I with the given parameters p,r,t (principal, rate, time).

Write a java program based on above scenario to help Raju.

Sample Test Case:

```
Enter the principal amount: 850000.56
Enter the rate of interest (in percentage): 3.5
Enter the time (in years): 2.3
Simple Interest: 68425.04508
```

Source Code:

q1169/SimpleInterestCalculator.java

```
package q1169;
import java.util.Scanner;

public class SimpleInterestCalculator {
    public static void main(String[] args) {

        //write your code here
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the principal amount: ");
        double p=sc.nextDouble();
        System.out.print("Enter the rate of interest (in percentage): ");
        double r=sc.nextDouble();
        System.out.print("Enter the time (in years): ");
        double t=sc.nextDouble();
        double i=(double)((p*r*t)/100);
        System.out.println("Simple Interest: "+i);
    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter the principal amount:

10000

Enter the rate of interest (in percentage):

3

Enter the time (in years):

2

Simple Interest: 600.0

Test Case - 2**User Output**

Enter the principal amount:

5000.658

Enter the rate of interest (in percentage):

5

Enter the time (in years):

5

Simple Interest: 1250.1645

Test Case - 3

User Output

Enter the principal amount:

560000

Enter the rate of interest (in percentage):

3

Enter the time (in years):

2

Simple Interest: 33600.0

Test Case - 4

User Output

Enter the principal amount:

850000.56

Enter the rate of interest (in percentage):

3.5

Enter the time (in years):

2.3

Simple Interest: 68425.04508

S.No: 15

Exp. Name: **Write 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.**

Date: 2023-10-10

Aim:

Write a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c$. Read in a, b, c and use the quadratic formula.

Source Code:

realSolutions.java

```
import java.util.Scanner;
class realSolutions{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter Value for a: ");
        double a=sc.nextDouble();
        System.out.print("Enter Value for b: ");
        double b=sc.nextDouble();
        System.out.print("Enter Value for c: ");
        double c=sc.nextDouble();
        double d=b*b-4.0*a*c;
        if(d>0.0)
        {
            double r1=(-b+Math.pow(d,0.5))/(2.0*a);
            double r2=(-b-Math.pow(d,0.5))/(2.0*a);
            System.out.println("Real Solutions are: "+r1+","+r2);
        }
        else if(d==0.0)
        {
            double r1=-b/(2.0*a);
            System.out.println("Real Solutions are: "+r1);
        }
        else{
            System.out.println("No Real Solutions");
        }
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter Value for a:

1

Enter Value for b:

4

Enter Value for c:

6

No Real Solutions

Test Case - 2

User Output

Enter Value for a:

1

Enter Value for b:

6

Enter Value for c:

6

Real Solutions are: -1.2679491924311228, -4.732050807568877

Aim:

Sarah is learning about the Fibonacci sequence in her maths class. Her teacher asked to start the sequence with two initial values, which are 0 & 1. Sarah is fascinated by this sequence and wants to explore it further.

Write a java program that generates the first **n** numbers in the Fibonacci sequence in Recursive(with using recursion) and Non-recursive way(without using recursion), starting with the initial values of 0 and 1, and then prints them out.

Sample Test Case:

```
Enter the range: 10
Fibonacci Sequence (Recursive):
0 1 1 2 3 5 8 13 21 34
Fibonacci Sequence (Non-Recursive):
0 1 1 2 3 5 8 13 21 34
```

Source Code:

```
q1172/Fibonacci.java
```

```

package q1172;
import java.util.Scanner;

public class Fibonacci {

    // Recursive function to calculate Fibonacci number
    public static int fibonacciRecursive(int n) {
        //write code with using recursion
        if(n<=1){
            return n;
        }
        return fibonacciRecursive(n-1)+fibonacciRecursive(n-2);
    }

    // Non-recursive (iterative) function to calculate Fibonacci number
    public static int fibonacciIterative(int n) {
        //write code without using recursion
        int a=0,b=1,c;
        if(n==0)
        {
            return a;
        }
        for(int i=2;i<=n;i++){
            c=a+b;
            a=b;
            b=c;
        }
        return b;
    }

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

        System.out.print("Enter the range: ");
        int n = scanner.nextInt();

        System.out.println("Fibonacci Sequence (Recursive):");
        for (int i = 0; i < n; i++) {
            System.out.print(fibonacciRecursive(i) + " ");
        }

        System.out.println("\nFibonacci Sequence (Non-Recursive):");
        for (int i = 0; i < n; i++) {
            System.out.print(fibonacciIterative(i) + " ");
        }
        System.out.println();
    }
}

```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter the range:

5

Fibonacci Sequence (Recursive):

0 1 1 2 3

Fibonacci Sequence (Non-Recursive):

0 1 1 2 3

Test Case - 2**User Output**

Enter the range:

10

Fibonacci Sequence (Recursive):

0 1 1 2 3 5 8 13 21 34

Fibonacci Sequence (Non-Recursive):

0 1 1 2 3 5 8 13 21 34

Test Case - 3**User Output**

Enter the range:

4

Fibonacci Sequence (Recursive):

0 1 1 2

Fibonacci Sequence (Non-Recursive):

0 1 1 2

Test Case - 4**User Output**

Enter the range:

9

Fibonacci Sequence (Recursive):

0 1 1 2 3 5 8 13 21

Fibonacci Sequence (Non-Recursive):

0 1 1 2 3 5 8 13 21

S.No: 17

Exp. Name: **Conditional Statements-if, if-else, Nested-if, switch**

Date: 2023-10-10

Aim:

Ravi is a pension officer. Ravi's job is to collect the person's name, age and prepare a data which should be in a below format :

```
Enter your name: srividya
Enter your age: 28
Hello, srividya!
You are 28 years old.
You were born in the year 1995.
You are an adult.
```

Ravi wants to declares them as

- Children if they are under 18 years of age,
- Adults if they are between 18 to 65.
- Remaining as senior citizens.

Assume you are ravi. How would you solve the problem?

Note: "You are a minor."

"You are an adult."

"You are a senior citizen." use these sentences in print statements appropriately.

Source Code:

```
q1158/ScannerWithIfElse.java
```

```

package q1158;
import java.util.Scanner;

public class ScannerWithIfElse
{
    public static void main(String[] args)
    {
        // Create a Scanner object to read user input
        Scanner scanner = new Scanner(System.in);

        // Prompt the user for their name
        System.out.print("Enter your name: ");
        String name = scanner.nextLine();

        // Prompt the user for their age
        System.out.print("Enter your age: ");
        int age = scanner.nextInt();

        // Calculate the year of birth
        int birthYear = java.time.Year.now().getValue() - age;

        // Complete below statements
        System.out.println("Hello, " + name + "!");
        System.out.println("You are " + age + " years old.");
        System.out.println("You were born in the year " + birthYear + ".");

        // Add conditional logic based on age
        if (age<18){
            System.out.println("You are a minor.");
        }
        else if(age>=18 && age<=65){
            System.out.println("You are an adult.");
        }
        else if(age>65){
            System.out.println("You are a senior citizen.");
        }
    }
}

```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter your name:

balu

Enter your age:

76

Hello, balu!
You are 76 years old.
You were born in the year 1947.
You are a senior citizen.

Test Case - 2

User Output

Enter your name:

harsha

Enter your age:

34

Hello, harsha!

You are 34 years old.

You were born in the year 1989.

You are an adult.

Test Case - 3

User Output

Enter your name:

preethi

Enter your age:

13

Hello, preethi!

You are 13 years old.

You were born in the year 2010.

You are a minor.

Test Case - 4

User Output

Enter your name:

srividya

Enter your age:

28

Hello, srividya!

You are 28 years old.

You were born in the year 1995.

You are an adult.

S.No: 18

Exp. Name: **Conditional Statements-if, if-else, Nested-if, switch**

Date: 2023-10-13

Aim:

Create a class InterviewProcess. Write a Java program that models an interview process with multiple rounds. The interview process consists of the following rounds: written test, group discussion, technical round, and HR round. The program should ask the candidate for their results at each stage and determine if they are eligible to proceed to the next round or if they should go home.

Sample Test Case:

```
Did you clear the written test? (yes/no): yes  
You are eligible for the group discussion round  
Did you pass the group discussion round? (yes/no): yes  
You are eligible for the technical round  
Did you pass the technical round? (yes/no): yes  
Congrats! You are eligible for the HR round
```

Note: You must write "Sorry, You can go home" in else block.

Source Code:

```
q1162/InterviewProcess.java
```

```

package q1162;
import java.util.Scanner;

public class InterviewProcess
{
    public static void main(String[] args)
    {
        // Create a Scanner object to read user input
        Scanner scanner = new Scanner(System.in);

        // write the code here
        System.out.print("Did you clear the written test? (yes/no): ");
        String a=scanner.next();
        if(a.equalsIgnoreCase("yes")){
            System.out.println("You are eligible for the group discussion round");
            System.out.print("Did you pass the group discussion round? (yes/no): ");
            String b=scanner.next();
            if(b.equalsIgnoreCase("yes")){
                System.out.println("You are eligible for the technical round");
                System.out.print("Did you pass the technical round? (yes/no): ");
                String c=scanner.next();
                if(c.equalsIgnoreCase("yes")){
                    System.out.println("Congrats! You are eligible for the HR
round");
                }
                else{
                    System.out.println("Sorry,You can go home");
                }
            }
            else{
                System.out.println("Sorry,You can go home");
            }
        }
        else{
            System.out.println("Sorry,You can go home");
        }
    }
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Did you clear the written test? (yes/no):
yes
You are eligible for the group discussion round
Did you pass the group discussion round? (yes/no):
no
Sorry,You can go home

Test Case - 2

User Output

Did you clear the written test? (yes/no):

yes

You are eligible for the group discussion round

Did you pass the group discussion round? (yes/no):

yes

You are eligible for the technical round

Did you pass the technical round? (yes/no):

yes

Congrats! You are eligible for the HR round

S.No: 20

Exp. Name: **Conditional Statements-if, if-else, Nested-if, switch**

Date: 2023-10-13

Aim:

Ravi possesses three geometric figures: a rectangle, a square, and a circle. He has the measurements for these shapes but lacks the knowledge to compute their respective areas. Create a program that assists Ravi in determining the areas of these different shapes.

Note: "Invalid choice. Please choose 1, 2, or 3." Use these sentences in default

Sample Test Case:

Choose a shape:

- 1. Rectangle
- 2. Square
- 3. Circle

Enter your choice (1/2/3): 1

Enter the length of the rectangle: 12

Enter the width of the rectangle: 13

The area of the rectangle is: 156.0

Source Code:

q1168/AreaCalculator.java

```

package q1168;
import java.util.Scanner;
public class AreaCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Choose a shape:");
        System.out.println("1. Rectangle");
        System.out.println("2. Square");
        System.out.println("3. Circle");
        System.out.print("Enter your choice (1/2/3): ");
        int choice = scanner.nextInt();

        switch (choice) {
            //write code here
            case 1:
                System.out.print("Enter the length of the rectangle: ");
                double l=scanner.nextDouble();
                System.out.print("Enter the width of the rectangle: ");
                double b=scanner.nextDouble();
                double area=l*b;
                System.out.println("The area of the rectangle is: "+area);
                break;

            case 2:
                System.out.print("Enter the side length of the square: ");
                double s=scanner.nextDouble();
                System.out.println("The area of the square is: "+(s*s));
                break;

            case 3:
                System.out.print("Enter the radius of the circle: ");
                double r=scanner.nextDouble();
                System.out.println("The area of the circle is: "+(Math.PI*r*r));
                break;

            default:
                System.out.println("Invalid choice. Please choose 1, 2, or 3.");
        }
    }
}

```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Choose a shape:

1. Rectangle
2. Square
3. Circle

Enter your choice (1/2/3):

9

Invalid choice. Please choose 1, 2, or 3.

Test Case - 2

User Output

Choose a shape:

- 1. Rectangle
- 2. Square
- 3. Circle

Enter your choice (1/2/3):

1

Enter the length of the rectangle:

20.56

Enter the width of the rectangle:

10.365

The area of the rectangle is: 213.1044

Test Case - 3

User Output

Choose a shape:

- 1. Rectangle
- 2. Square
- 3. Circle

Enter your choice (1/2/3):

2

Enter the side length of the square:

30.569

The area of the square is: 934.463761

Test Case - 4

User Output

Choose a shape:

- 1. Rectangle
- 2. Square
- 3. Circle

Enter your choice (1/2/3):

3

Enter the radius of the circle:

50.64

The area of the circle is: 8056.33036015514

Test Case - 5

User Output

Choose a shape:

1. Rectangle

2. Square

3. Circle

Enter your choice (1/2/3):

1

Enter the length of the rectangle:

256.25

Enter the width of the rectangle:

365.2

The area of the rectangle is: 93582.5

S.No: 21

Exp. Name: **Conditional Statements-if, if-else, Nested-if, switch**

Date: 2023-10-10

Aim:

Surya owns a digital watch and wishes to include a feature that provides information about the number of days in each month, ranging from 1 to 12. When he presses a single digit(month), the watch should display the corresponding number of days, such as 31. Assist Surya in implementing this update.

Note:

Use the below printing statements appropriately

"Enter a month number (1 to 12): "

"31 days in this month."

"30 days in this month."

"28 or 29 days in this month (leap year dependent)."

"Invalid month number. Please enter a number from 1 to 12."

Source Code:

```
q1170/DayscheckWithNumbers.java
```

```

package q1170;
import java.util.Scanner;

public class DayscheckWithNumbers
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a month number (1 to 12): ");
        int month = scanner.nextInt();

        //write a code here
        switch(month){
            case 1: case 3: case 5: case 7: case 8: case 10: case 12:
                System.out.println("31 days in this month.");
                break;
            case 4: case 6: case 9: case 11:
                System.out.println("30 days in this month.");
                break;
            case 2:
                System.out.println("28 or 29 days in this month (leap year
dependent).");
                break;
            default:
                System.out.println("Invalid month number. Please enter a
number from 1 to 12.");
        }

        //for your reference observe the below comment(you can use one printing statement for
multiple cases)
        //case 4: case 6: case 9: case 11:
        //System.out.println("30 days in this month.");
    }
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter a month number (1 to 12):
0
Invalid month number. Please enter a number from 1 to 12.
Test Case - 2
User Output
Enter a month number (1 to 12):
9
30 days in this month.

Test Case - 3

User Output

Enter a month number (1 to 12):

3

31 days in this month.

Test Case - 4

User Output

Enter a month number (1 to 12):

11

30 days in this month.

Aim:

Write a Java Program that prompts the user for an integer and then prints out all the prime numbers up to that Integer.

Sample Test Case:

```
Enter the end of the range: 10
2 3 5 7
```

Source Code:

`q1167/PrimeNumber.java`

```
package q1167;
import java.util.Scanner;
public class PrimeNumber {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the end of the range: ");
        // write your logic here
        int a=input.nextInt();
        for(int i=2;i<=a;i++){
            int count=0;
            for(int j=1;j<=i;j++){
                if(i%j==0){
                    count++;
                }
            }
            if (count==2){
                System.out.print(i+" ");
            }
        }
    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter the end of the range:

10

2 3 5 7

Test Case - 2**User Output**

Enter the end of the range:

20

2 3 5 7 11 13 17 19

Test Case - 3

User Output

Enter the end of the range:

100

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Test Case - 4

User Output

Enter the end of the range:

30

2 3 5 7 11 13 17 19 23 29

Aim:

Write a program for the below requirements

- * calculate sum and product of the given number
- * if sum equals to product print it is a spy number and print the given number in reverse order
- * if sum and product not equal, print it is not a spy number.

Note : use only while loop

Sample Test Case : 1

```
123
sum of the given number is: 6
product of the given number is: 6
it is a spy number
Given number in reverse order: 321
```

Sample Test Case : 2

```
456
sum of the given number is: 15
product of the given number is: 120
it is not a spy number
```

Source Code:

```
q1165/SpyNumber.java
```

```

package q1165;
import java.util.Scanner;
class SpyNumber {
    public static void main(String[] args) {
        Scanner s1 = new Scanner(System.in);

        // write your code here
        int n=s1.nextInt();
        int n1=n;
        int sum=0;
        int p=1;
        int lastdigit;
        while(n>0){
            lastdigit=n%10;
            sum=sum+lastdigit;
            p=p*lastdigit;
            n=n/10;
        }
        System.out.println("sum of the given number is: "+sum);
        System.out.println("product of the given number is: "+p);

        if(sum==p){
            System.out.println("it is a spy number");
            int reverse=0;
            while(n1!=0){
                int r=n1%10;
                reverse=reverse*10+r;
                n1=n1/10;
            }
            System.out.println("Given number in reverse order: "+reverse);
        }
        else{
            System.out.println("it is not a spy number");
        }
    }
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
123
sum of the given number is: 6
product of the given number is: 6
it is a spy number
Given number in reverse order: 321

Test Case - 2
User Output
456

sum of the given number is: 15

product of the given number is: 120

it is not a spy number

Test Case - 3

User Output

1124

sum of the given number is: 8

product of the given number is: 8

it is a spy number

Given number in reverse order: 4211

Test Case - 4

User Output

1234

sum of the given number is: 10

product of the given number is: 24

it is not a spy number

Aim:

Rani needs to create a Java program that operates a printing machine to produce a pyramid-shaped pattern using alphabetic characters. This program should take a parameter 'n' to determine the number of rows in the pyramid and then navigate the printing machine accordingly to achieve the desired pyramid pattern.

Sample Test Case:

```
4
 A
 A B
 A B C
A B C D
```

Source Code:

q1171/AlphabetPyramid.java

```
package q1171;
import java.util.Scanner;
public class AlphabetPyramid
{
    public static void main(String[] args)
    {
        Scanner s1=new Scanner(System.in);
        //write your logic here
        int n=s1.nextInt();
        for(int i=0;i<n;i++){
            int a=65;
            for(int j=n-i+1;j>1;j--){
                System.out.print(" ");
            }

            for(int j=0;j<=i;j++){
                System.out.print((char)(a+j)+" ");
            }

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

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

```
4
 A
 A B
 A B C
A B C D
```

Test Case - 2

User Output

3
A
A B
A B C

Test Case - 3

User Output

10
A
A B
A B C
A B C D
A B C D E
A B C D E F
A B C D E F G
A B C D E F G H
A B C D E F G H I
A B C D E F G H I J

Test Case - 4

User Output

7
A
A B
A B C
A B C D
A B C D E
A B C D E F
A B C D E F G

Aim:**Write a Program to remove an Element from given Array****Problem Statement:**

- The function should return the array excluding the given element

Function Rules:Fill the missing logic in **function** `RemoveElementFromArray` with **return type** `int[]` and **parameters** as listed below:

- `int[] arr`
- `int element`

Source Code:`q1145/RemoveElementFromArray.java`

```

package q1145;
import java.util.Arrays;
import java.util.Scanner;

public class RemoveElementFromArray {
    public static int[] removeElement(int[] arr, int element) {
        //write your code here
        int newSize=0;
        for(int value:arr){
            if(value!=element){
                newSize++;
            }
        }
        int[] newArray=new int[newSize];
        int index=0;

        for(int value:arr){
            if(value!=element){
                newArray[index]=value;
                index++;
            }
        }
        return newArray;
    }

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

        // Input the array size
        System.out.print("Enter the size of the array: ");
        int size = scanner.nextInt();

        int[] arr = new int[size];

        // Input array elements
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < size; i++) {
            arr[i] = scanner.nextInt();
        }

        // Input the element to be removed
        System.out.print("Enter the element to remove: ");
        int elementToRemove = scanner.nextInt();

        // Call the removeElement method to remove the element
        int[] modifiedArray = removeElement(arr, elementToRemove);

        // Display the modified array
        System.out.println("Array after removing " + elementToRemove + ":");
        for (int i = 0; i < modifiedArray.length; i++) {
            System.out.print(modifiedArray[i] + " ");
        }
        System.out.println();
    }
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the size of the array:
5
Enter the elements of the array:
2 3 4 5 6
Enter the element to remove:
5
Array after removing 5:
2 3 4 6

Test Case - 2
User Output
Enter the size of the array:
6
Enter the elements of the array:
3 4 5 6 7 77
Enter the element to remove:
1
Array after removing 1:
3 4 5 6 7 77

Test Case - 3
User Output
Enter the size of the array:
10
Enter the elements of the array:
1 2 3 4 5 6 7 8 9 11
Enter the element to remove:
9
Array after removing 9:
1 2 3 4 5 6 7 8 11

Test Case - 4
User Output
Enter the size of the array:
4
Enter the elements of the array:
5 6 89 63
Enter the element to remove:
5
Array after removing 5:

Aim:

A group of people, including Deeksha, attended an exhibition. Deeksha, the organizer, is concerned about the safety of the attendees. To address this concern, she needs a program that can identify the two individuals in the group with the highest and lowest ages.

Write a program that will assist Deeksha in ensuring the well-being of the exhibition's participants.(Observe the Sample Test Case)

Sample Test Case:

```
Enter the number of people: 5
Enter the age of each person:
55 69 52 20 10
Maximum age: 69
Minimum age: 10
```

Source Code:

[q1147/FindMinAndMax.java](#)

```
package q1147;
import java.util.Scanner;
public class FindMinAndMax {
    public static void main(String[] args) {
        //write your code here
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the number of people: ");
        int n=sc.nextInt();
        int age[]=new int[n];
        System.out.println("Enter the age of each person:");
        for(int i=0;i<n;i++){
            age[i]=sc.nextInt();
        }
        int max=age[0];
        int min=age[0];
        for(int i=1;i<age.length;i++){
            if(age[i]>max){
                max=age[i];
            }
            else if(age[i]<min){
                min=age[i];
            }
        }
        System.out.println("Maximum age: "+max);
        System.out.println("Minimum age: "+min);
    }
}
```

Execution Results - All test cases have succeeded!**Test Case - 1****User Output**

Enter the number of people:

5

Enter the age of each person:

55 69 52 20 10

Maximum age: 69

Minimum age: 10

Test Case - 2

User Output

Enter the number of people:

6

Enter the age of each person:

22 23 63 10 55 44

Maximum age: 63

Minimum age: 10

Test Case - 3

User Output

Enter the number of people:

2

Enter the age of each person:

56 45

Maximum age: 56

Minimum age: 45

Test Case - 4

User Output

Enter the number of people:

6

Enter the age of each person:

2 54 66 33 44 41

Maximum age: 66

Minimum age: 2

Aim:

Write a java program to print **Leaders** from the given array

Explanation:

In the context of an array, "leaders" are elements that are greater than all the elements to their right in the array. In other words, a leader is an element that does not have any element to its right that is greater than itself. Leaders "stand out" in the sense that they are the largest elements towards the right end of the array.

Here's an example to illustrate the concept of leaders in an array:

Suppose you have an array:

```
arr = [16, 17, 4, 3, 5, 2]
```

In this example, the leaders in the array are 17, 5, and 2.

The task of finding leaders in an array involves iterating through the array from right to left and identifying elements that are greater than or equal to the maximum element seen so far to their right. If an element meets this condition, it is considered a leader and can be printed.

Print the leaders in the **reverse order**(right to left) of the array.

Sample Test Case:

```
Enter the number of elements in the array: 5
Enter the elements of the array:
78 85 6 45 55
Leaders in the array:
55 85
```

Source Code:

```
q1160/LeadersInArray.java
```

```

package q1160;
import java.util.Scanner;

public class LeadersInArray {
    public static void findLeaders(int[] arr) {
        //write your code here
        int n=arr.length;
        int maxRight=arr[n-1];
        System.out.println("Leaders in the array:");
        for(int i=n-1;i>=0;i--){
            if(arr[i]>=maxRight){
                maxRight=arr[i];
                System.out.print(maxRight+" ");
            }
        }
        System.out.println();
    }

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

        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();

        int[] arr = new int[n];

        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
        findLeaders(arr);
    }
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the number of elements in the array:
5
Enter the elements of the array:
78 85 6 45 55
Leaders in the array:
55 85

Test Case - 2
User Output
Enter the number of elements in the array:

6

Enter the elements of the array:

85 69 54 71 65

Leaders in the array:

5 6 71 85

Test Case - 3

User Output

Enter the number of elements in the array:

3

Enter the elements of the array:

2 8 6

Leaders in the array:

6 8

Test Case - 4

User Output

Enter the number of elements in the array:

4

Enter the elements of the array:

-54 -8 -75 -7

Leaders in the array:

-7

Aim:**Problem Statement:**

The computer vision researchers are developing an advanced image enhancement algorithm for medical X-ray images. To achieve this, they need to transpose the pixel matrices of these X-ray images, which are stored in a database. Your task is to create a program that can efficiently transpose a given two-dimensional matrix, helping the researchers in their image enhancement process.

Sample Test Case:

```
Enter the number of rows: 3
Enter the number of columns: 2
Enter the elements of the matrix:
2 4
5 4
8 7
Transpose of the Matrix:
2 5 8
4 4 7
```

Source Code:**q1164/MatrixTranspose.java**

```
package q1164;
import java.util.Scanner;

public class MatrixTranspose {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of rows: ");
        int numRows = scanner.nextInt();
        System.out.print("Enter the number of columns: ");
        int numCols = scanner.nextInt();

        int[][] matrix = new int[numRows][numCols];

        System.out.println("Enter the elements of the matrix:");
        for (int i = 0; i < numRows; i++) {
            for (int j = 0; j < numCols; j++) {
                matrix[i][j] = scanner.nextInt();
            }
        }
        //write your code here
        System.out.println("Transpose of the Matrix:");
        for(int i=0;i<numCols;i++){
            for(int j=0;j<numRows;j++){
                System.out.print(matrix[j][i]+" ");
            }
            System.out.println();
        }

    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the number of rows:
3
Enter the number of columns:
3
Enter the elements of the matrix:
5 6 8
5 7 9
3 6 8
Transpose of the Matrix:
5 5 3
6 7 6
8 9 8

Test Case - 2
User Output
Enter the number of rows:
4
Enter the number of columns:
4
Enter the elements of the matrix:
5 6 9 8
6 3 9 8
5 7 2 1
4 4 2 7
Transpose of the Matrix:
5 6 5 4
6 3 7 4
9 9 2 2
8 8 1 7

Test Case - 3
User Output
Enter the number of rows:
2
Enter the number of columns:
2
Enter the elements of the matrix:
2 1
5 4
Transpose of the Matrix:
2 5

Test Case - 4**User Output**

Enter the number of rows:

3

Enter the number of columns:

2

Enter the elements of the matrix:

2 4

5 4

8 7

Transpose of the Matrix:

2 5 8

4 4 7

S.No: 30

Exp. Name: **Program to check the given String is Palindrome or not**

Date: 2023-10-13

Aim:

Create a class [PalindromeOrNot](#) with a **main** method. The method receives one command line argument. Check the given argument is palindrome or not.

For example:

```
Cmd Args : madam
The given string madam is a palindrome

Cmd Args : Godavari
The given string Godavari is not a palindrome
```

Note: Please don't change the package name.

Source Code:

[q11184/PalindromeOrNot.java](#)

```
package q11184;
import java.util.Scanner;
class PalindromeOrNot{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        String a=args[0];
        String b="";
        int n=a.length();
        for(int i=n-1;i>=0;i--){
            b=b+a.charAt(i);
        }
        if(a.equals(b)){
            System.out.println("The given string "+a+" is a palindrome");
        }
        else{
            System.out.println("The given string "+a+" is not a palindrome");
        }
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

The given string madam is a palindrome

Test Case - 2

User Output

The given string Godavari is not a palindrome

Test Case - 3

User Output

The given string malayalam is a palindrome

Test Case - 4

User Output

The given string 12345 is not a palindrome