Convert Temperature from Celsius to Fahrenheit in Java

🧩Problem Statement

Write a Java program to convert a given temperature from **Celsius (°C)** to **Fahrenheit (°F)**.

The conversion formula is:

Fahrenheit = (Celsius × 9 / 5) + 32

The program should read a temperature value in Celsius, apply the formula, and display the corresponding temperature in Fahrenheit.

Input Format

- A single number representing the temperature in **Celsius**.
- The input can be an integer or a decimal value.

Edge Cases

- Very large or very small temperature values
- Zero (0°C)
- Negative temperatures (below freezing)

Output Format

- Print a single number representing the temperature in **Fahrenheit**.
- The output can be a decimal value.

Constraints

- **Range of Celsius value**: -273.15 ≤ C ≤ 1,000,000
- (-273.15°C is absolute zero)
- Input must be a valid numeric value
- No restriction on decimal precision unless specified

🧨Example Inputs and Outputs

Example 1 (Normal Case)

Input:

```
0
```

Output:

```
32.0
```

Example 2 (Negative Temperature – Edge Case)

Input:

```
-40
```

Output:

```
-40.0
```

Example 3 (Decimal Value)

Input:

```
36.5
```

Output:

```
97.7
```

🎷Normal Solution (Java)

```java
import java.util.Scanner;

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

        // Read temperature in Celsius
```

```
        double celsius = sc.nextDouble();

        // Convert Celsius to Fahrenheit
        double fahrenheit = (celsius * 9.0 / 5) + 32; // use floating-point
division explicitly

        // Print the result
        System.out.println(fahrenheit);

        sc.close();
    }
}
```

Optimized Solution (Java)

Since this problem involves a **single mathematical formula**, the normal solution is already optimal. However, we can slightly simplify the code by removing unnecessary objects when input handling is abstracted.

```
public class CelsiusToFahrenheit {
    public static void main(String[] args) {
        double celsius = Double.parseDouble(args[0]);
        double fahrenheit = (celsius * 9.0 / 5) + 32; // use floating-point
division explicitly
        System.out.println(fahrenheit);
    }
}
```

✅This version is useful when input is passed via **command-line arguments**.

🪡Step-by-Step Explanation

Normal Solution Logic

1. Read the temperature value in Celsius from the user.
2. Apply the conversion formula:
3. Multiply Celsius by 9
4. Divide the result by 5
5. Add 32
6. Store the result in a Fahrenheit variable.
7. Print the converted value.

Optimized Solution Logic

1. Accept Celsius value as a command-line argument.
2. Convert the string input into a double.

3. Apply the same conversion formula.
4. Print the output directly.

Time and Space Complexity

Normal Solution

- **Time Complexity:** O(1)
- **Space Complexity:** O(1)

Optimized Solution

- **Time Complexity:** O(1)
- **Space Complexity:** O(1)

The program always performs a constant number of operations.

Tips & Common Pitfalls

- ❌Using integer division **when both operands are integers** (e.g., `9 / 5` → `1`)
- ⚠️Writing `celsius * 9 / 5` *works only because* `celsius` *is* `double` *and Java promotes the expression*, but it is **not beginner-safe**
- ✅Best practice: **explicitly use floating-point literals** like `9.0 / 5`
- ❌Forgetting to handle decimal inputs
- ❌Not validating input when accepting user data
- ✅Always use `double` for temperature calculations
- ✅Remember the correct formula order: multiply → divide → add

---

Summary

- This is a **basic mathematical conversion problem**
- Ideal for beginners learning Java input/output and arithmetic operations
- Helps understand data types, formulas, and clean coding practices

Happy Coding 🚳