Turiba University

(Mehak Singh)

(Friendly coding)

PROFESSIONAL BACHELOR'S DEGREE

Study programme (Bachelor's in Computer Systems)

Author: **Mehak Singh**

Thesis Advisor: **(Jānis Pekša, Dean of IT Dept.)**

Riga, 2025

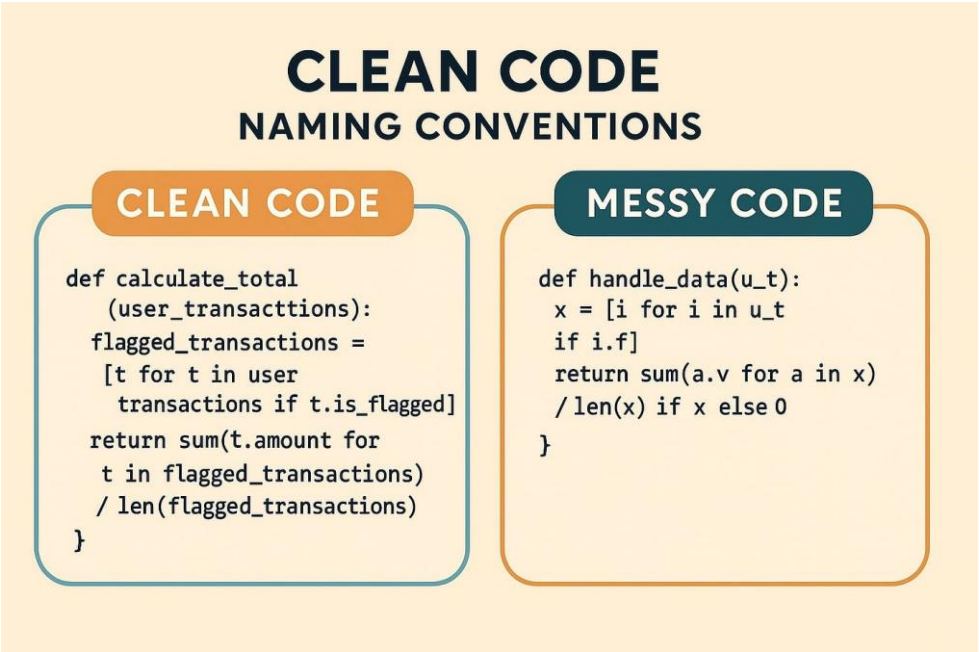
1. Introduction

Coding isn't only about getting a computer to follow commands — it's also a way of expressing ideas to people. Whenever we write code, we're communicating with two groups:

• The machine, which strictly follows instructions, and

• Other humans (including our future selves), who must read, update, and maintain that code later.

This is where the idea of Friendly Coding becomes important. Friendly coding means writing programs that feel natural, clear, and easy to follow. It makes your code "talk" in a simple, friendly manner — just like a person explaining something in a comfortable way. In simple terms, friendly coding means readable code. It helps prevent confusion, boosts productivity, and builds confidence and smooth teamwork among developers.

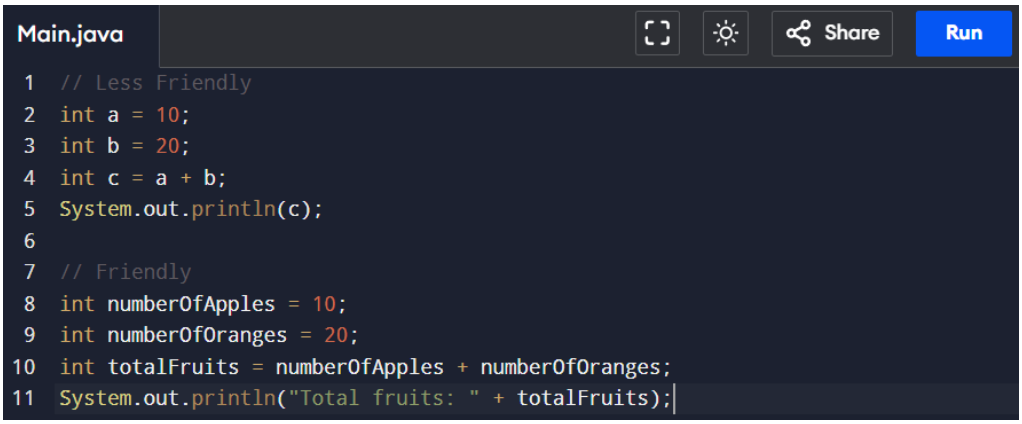


2. What is Friendly Coding?

Friendly coding is a way of writing programs that puts clarity and readability first. It's about creating code that anyone — including you, months later — can understand without getting confused or having to read it repeatedly.

A friendly coder doesn't aim to write code that only runs correctly — they aim to write code that also makes sense. It should explain itself naturally and be simple to follow.

Here’s a small example:



Even though both snippets work the same, the second one is friendlier because the naming makes it self-explanatory.

3. Why Friendly Coding Matters

Writing friendly code might not feel very important when you’re working by yourself, but in real-life software development, you’re almost always part of a team. That’s when friendly coding becomes truly valuable.

Here’s why it matters so much:

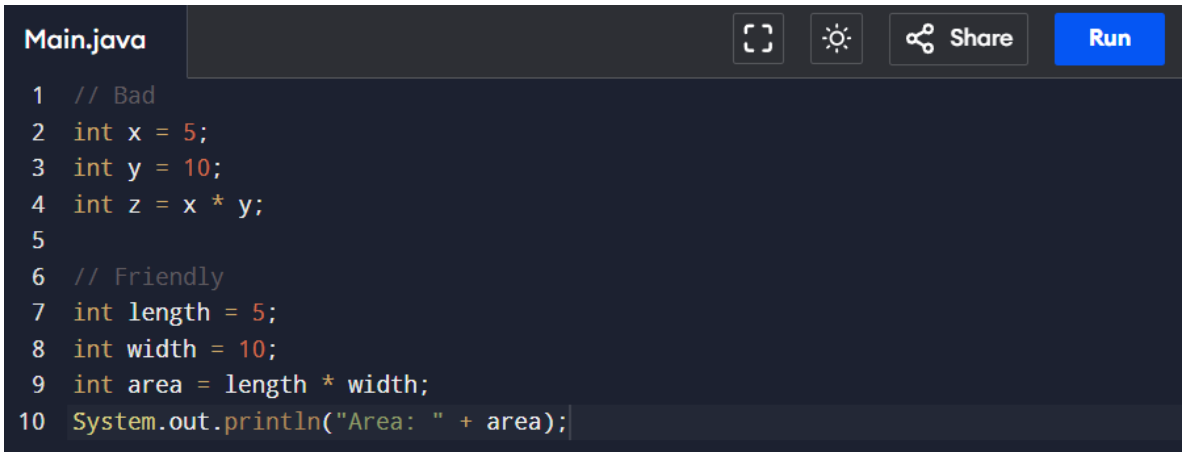
1. **Better Communication:**
   When your code has clear names, a clean layout, and simple comments, other people can understand your thinking without any struggle.
2. **Easier Fixing and Updating:**
   Readable code makes problems easier to spot. If the code makes sense at first glance, debugging becomes much quicker.
3. **Saves a Lot of Time Later:**
   When the logic is clear, adding new features or making changes takes much less effort.
4. **Helps Build Trust in a Team:**
   When your teammates can easily understand your code, they feel more confident working with it — and with you.
5. **Looks Professional:**
   Clean, well-written, friendly code shows that you take your work seriously and follow good coding practices.

4. Key Principles of Friendly Coding

Let’s look at the main habits that make code friendly — especially in Java.

4.1 Meaningful Variable and Method Names

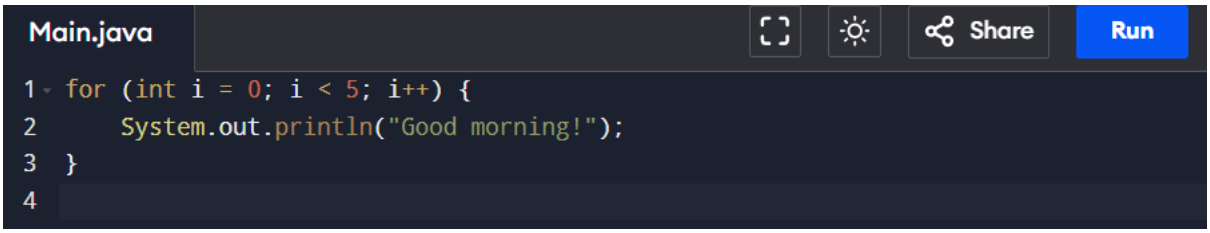
Your variable and method names should describe their purpose clearly.



This may seem small, but meaningful names make a massive difference.

4.2 Proper Indentation and Spacing

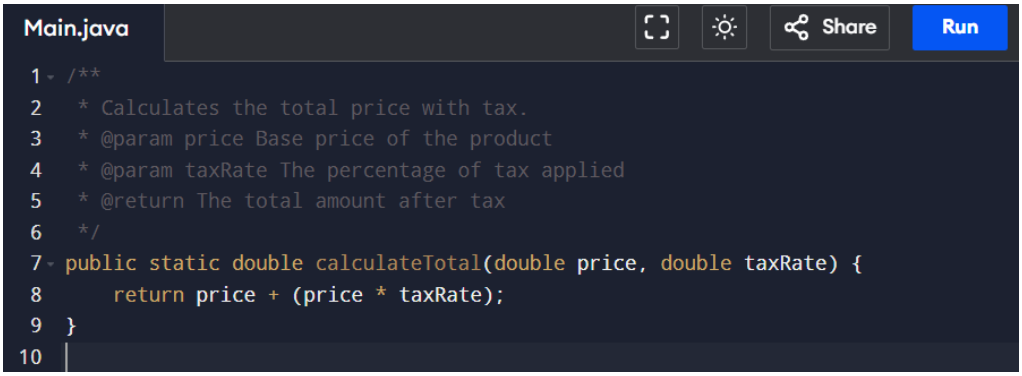
Neat formatting makes code easier to read and understand.
Follow Java conventions with consistent indentation (usually 4 spaces per block).



Bad indentation often hides logical errors and confuses readers.

4.3 Add Comments and Documentation

Comments explain why something is happening, not just what.
In Java, Javadoc comments are perfect for documenting methods.



This comment block helps anyone immediately understand the method's purpose.

4.4 Avoid Repetition (Use Functions)

If you're writing the same logic multiple times — make it a method.

This makes your code modular, cleaner, and reusable.`

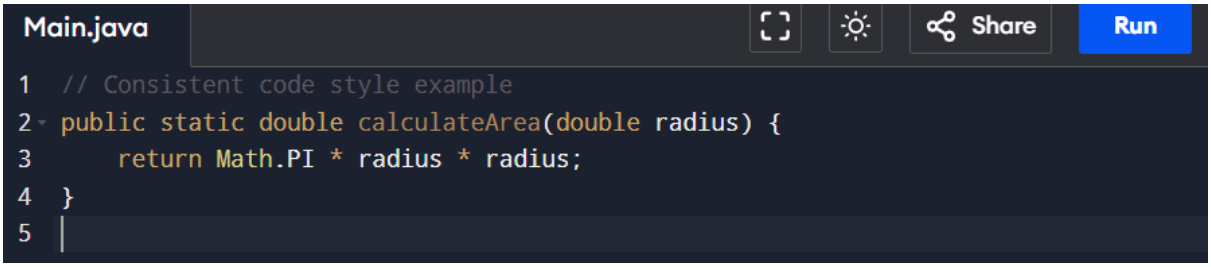
4.5 Handle Errors Gracefully

Friendly code should handle exceptions, not crash unexpectedly.

This makes your program user-friendly and error-resistant.

4.6 Consistency is Key

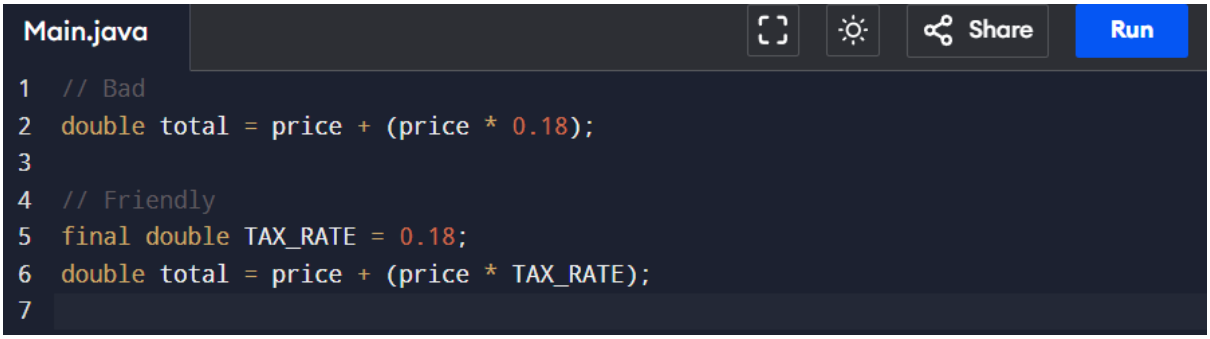
Use consistent formatting — the same bracket style, naming style, and spacing throughout your project.



Even small details like consistent bracket placement make code more readable.

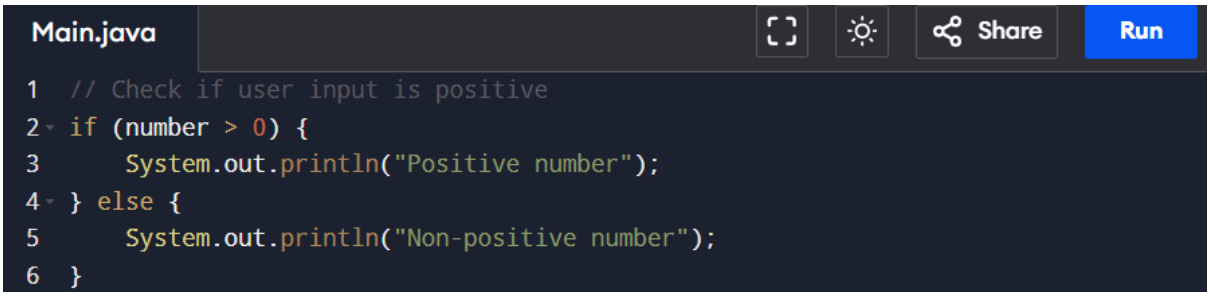
4.7 Use Constants Instead of Magic Numbers

Magic numbers make code confusing. Use final constants instead.



Now the meaning of “0.18” is clear to everyone.

4.8 Use Comments to Explain Logic



Even a simple one-line comment can save time when revisiting code after weeks.

5. Comparison: Friendly vs Unfriendly Code

Feature	Unfriendly Code	Friendly Code
Variable Names	a, b, c	length, width, height
Comments	None	Short and clear Javadoc
Structure	Messy, hard to read	Proper indentation and spacing
Error Handling	Ignored	Graceful try-catch
Reusability	Repetitive	Uses functions
Constants	Magic numbers	Named constants

6. Real-Life Analogy: Coding as a Conversation

Imagine explaining how to make tea to your friend.
If you say, "Just do it somehow," your friend will be lost.
But if you say, "Boil water, add tea leaves, then pour milk," your instructions are clear.

Coding is the same. Friendly coding explains things step by step — it's like giving clear instructions to your team and your future self.

7. Conclusion

Friendly coding is not about fancy tricks or shortcuts — it's about kindness, clarity, and collaboration**.**
Good code is like good writing: the more human it feels, the better it works.

In Java, friendly coding practices — meaningful names, consistent style, reusable methods, and graceful error handling — make your work shine not only technically but also professionally.

When you write friendly code, you're not just solving a problem — you're making the coding world a little more welcoming for everyone who reads your work.