



# Clean Code

Rawlabs Academy

```
public class UpdateController {  
    int i = 5;  
    String nm = "John Doe";  
    String e = "johndoe@gmail.com";  
    String yyymmddstr = "1996-01-12";  
  
    public String search(String i) {  
        ...  
    }  
}
```



**What do you  
think **about the  
code ?****

# What is **Clean Code** ?

**Clean Code** is term for code that is 'readable', 'understood' and 'altered' by programmers



“ Working code isn't necessary  
good code. Your code also need to  
be easy to read, understand, and  
modify

~ **Brandon Gregory**

”

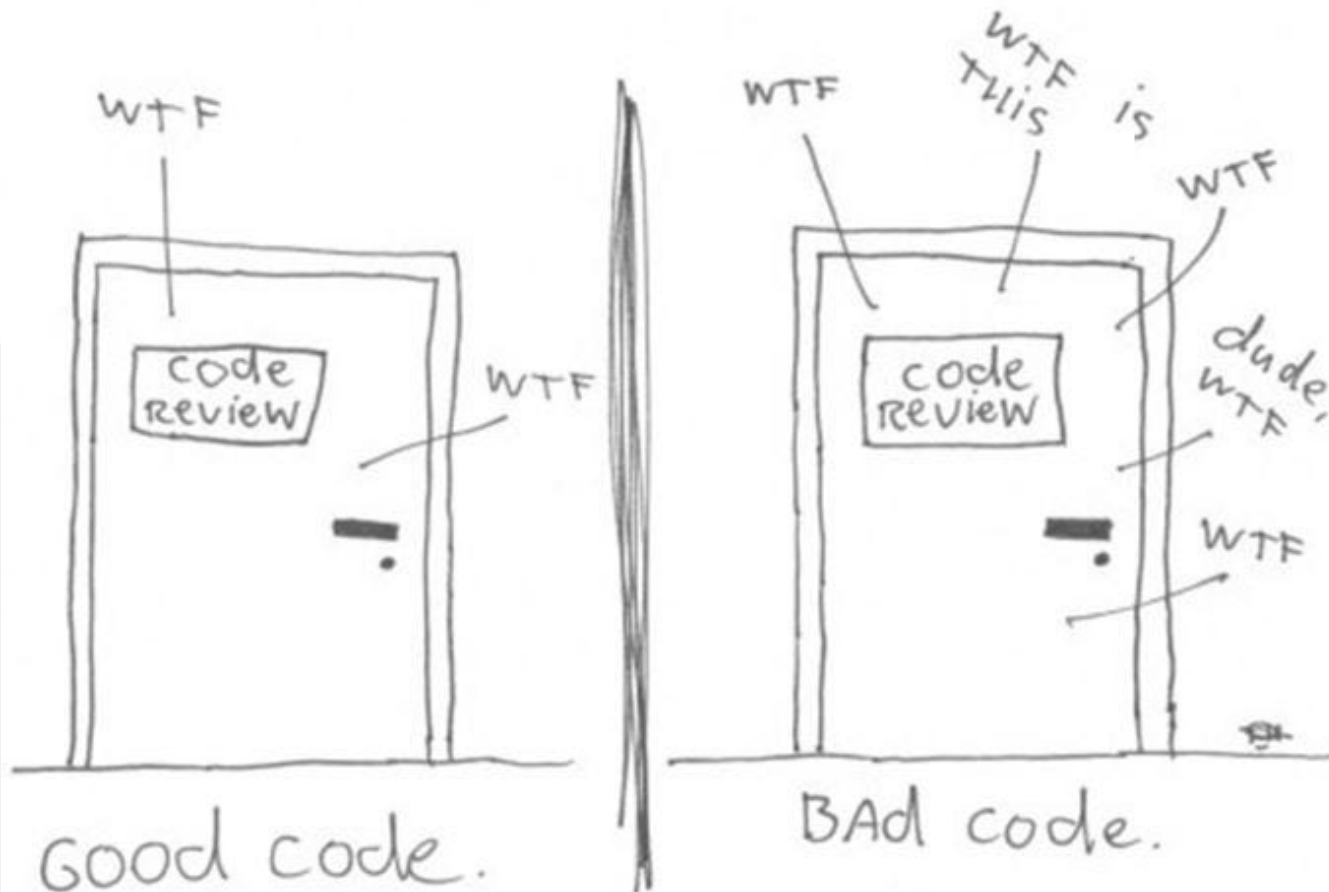


“ Any fool can write code that a computer can understand. Good programmers write code that humans can understand.

~ **Martin Flower**

”

The ONLY VALID MEASUREMENT  
OF CODE QUALITY: WTFs/MINUTE



# Why Clean Code?



Work Collaboration



Feature Development



Faster Development



# **Clean Code**

## **Characteristic**

# Easy to Understand - **DON'T DO THIS**

```
double b = 125.0;
int[] data = {2, 3, 5, 7};
List<String> locations = List.of("New York", "Sydney",
    "Texas", "San Francisco");

locations.forEach((l) -> {
    doStuff();
    doSomeOtherStuff();
    ...
    // Wait, `l` for what?
    doAnotherThing(l);
});
```

# Easy to Understand - **DO THIS**

```
double balance = 125.0;
int[] primeNumbers = {2, 3, 5, 7};
List<String> locations = List.of("New York", "Sydney",
    "Texas", "San Francisco");

locations.forEach((location) -> {
    doStuff();
    doSomeOtherStuff();
    ...
    doAnotherThing(location);
});
```

# Example Style Guide



Javascript : <https://github.com/airbnb/javascript>

Python : <https://google.github.io/styleguide/pyguide.html>

# Suggestion **Formatting?**

1. Line width code 80-120
2. One Class 300-500 lines
3. Lines of code that are related to each other
4. Keep the function close to its caller
5. Declaration of adjacent variables to their users
6. Pay attention to indentation
7. Using **prettier** or **formatter**

# Principle **Clean Code**

## KISS

Keep It So Simple

Avoid creating functions created to perform A, while modifying B, checking C functions, etc.

# Tips for always **KISS**

- Functions or classes should be small
- Functions created to perform a single task only
- Don't use too many arguments on functions
- Care must be taken to achieve a balanced, small and minimal number of conditions

# Principle **Clean Code**

## **DRY**

Don't Repeat Yourself

Code duplication occurs because of frequent copy and paste. To avoid duplication of code create functions that can be used repeatedly



# Refactoring



**Refactoring** is the process of restructuring the code created, by changing the internal structure without changing the external behavior. The principle of **KISS** and **DRY** can be achieved by refactoring.

# Technique **Refactoring**

- Creating an abstraction
- Breaking down code with functions/classes
- Fix code naming and location
- Detection of duplicated code

# Task

Create summary  
about Clean Code and  
some example  
implementation.

