**Clean Code**

Principle of clean code includes:

1. Follow standard conventions.
2. Keep it simple stupid. Simpler is always better. Reduce complexity as much as possible.
3. Boy scout rule. Leave the campground cleaner than you found it.
4. Always find root cause. Always look for the root cause of a problem.

Understandability tips:

1. Be consistent. If you do something a certain way, do all similar things in the same way.

Use explanatory variables.

1. Encapsulate boundary conditions. Boundary conditions are hard to keep track of. Put the processing for them in one place.
2. Avoid logical dependency. Don't write methods which works correctly depending on something else in the same class.
3. Avoid negative conditionals.

Design rules:

1. Keep configurable data at high levels.
2. Prefer polymorphism to if/else or switch/case.
3. Separate multi-threading code.
4. Prevent over-configurability.
5. Follow Law of Demeter. A class should know only its direct dependencies.

Some steps to write clean code:

1. **Naming:**

The variables, constants, functions and anything with name must be description and unambiguous name.

A white background with black text

Description automatically generated

1. **Function:**

Must be small, do one thing.

A white background with black text

Description automatically generated

1. **Comment:**

Explain my code in a few words (small description).

1. **Loops:**

Should detect my jobs and choose for loop or while or do while.

Iteration variables are better to use I, j, k because they are known by programmers.

A screenshot of a computer code

Description automatically generated

1. **Condition:**

Choosing if or switch case depended on my function.

1. **Objects and data structures:**

Hide internal structure, do one thing and avoid hybrids structure.

My resources:

* <https://maharatech.gov.eg/mod/hvp/view.php?id=11788>
* [Summary of 'Clean code' by Robert C. Martin (github.com)](https://gist.github.com/wojteklu/73c6914cc446146b8b533c0988cf8d29)