**Clean Code Check List**

*NOTE: Kindly download the document to your local to view the text on Hyperlink mouse hover.*

***Naming***

Use [**descriptive**](#_top)and meaningful names

☑︎ [**Variables & Properties**](#_top): use **Nouns or short phrases with adjectives**.

*For example: user, product, customer, database, transaction etc*.

☑︎ For **storing boolean values** use a **short phrase with an adjective**

*For example: isValid, didAuthenticate, isLoggedIn, emailExists etc.*

☑︎ [**Functions and Methods**](#_top): **Verbs or short phrases with adjectives**

*For example: login(), createUser(), database.insert(), log() etc.*

☑︎ [**Classes**](#_top): Noun

*For example: User, Product,RootAdministrator, Transaction, Payment etc.*

***Note :***

* Be as **specific** as necessary and possible
* **Avoid misleading** names
* Be **consistent** with your names (e.g. stick to get... instead of fetch...)
* Avoid generic names

***Comments & Formatting***

* + Most comments are [bad](#_top) - avoid them!
  + Some [good](#_top) comments are acceptable

☑︎ Legal comments

☑︎ Warnings

☑︎ Helpful explanations (e.g. for Regex)

☑︎ To dos (don't overdo it though)

* Use vertical formatting:

☑︎ Keep related concepts close to each other (vertical density)

☑︎ Add spacing / distance (e.g. blank linkes) between concepts that are not directly related (vertical distance)

☑︎ Write code top to bottom: Called functions should come below calling functions (if possible)

* Use horizontal formatting:

☑︎ Avoid long lines - break them into multiple lines instead

☑︎ Use indentation to express scope of code

***Functions***

* + Limit the number of parameters your functions use - less is better!
  + Consider using objects, dictionaries or arrays to group multiple parameters into one parameter
  + Functions should be small and do one thing

☑︎ Levels of abstraction inside the function body should be one level below the level implied by the function name

☑︎ Avoid mixing levels of abstractions in functions

☑︎ But: Avoid redundant splitting!

* + Stay DRY (Don't Repeat Yourself)
  + Avoid unexpected side effects

***Control Structures & Errors***

* + Prefer positive checks
  + Avoid deep nesting

☑︎ Consider using "Guard" statements

☑︎ Consider using polymorphism and factory functions

☑︎ Extract control structures into separate functions

* + Consider using "real" errors (with error handling) instead of "synthetic errors" built with if statements

***Objects & Classes***

* + Focus on building "real objects" or data containers / structures
  + Build small classes - focus on a single responsibility (which does not mean "single method"!)

☑︎ Build classes with high cohesion

* + Follow the "Law of Demeter" for "real objects"
  + Especially when doing OOP: Follow the SOLID principles
  + Especially SRP(Single Responsibility Principle) and OCP(Open Close Principle) will help a lot with writing clean code (= readable code)

\*\*\*\* Thank You\*\*\*