RULES OF NAMING AND NAMING CODE ELEMENTS

Be sure to refer to these naming rules anytime you are coding

1 RULES OF NAMING

Reveal Intention. The name of a variable, function, or class should tell you three things: why it exists, what it does, and how it is used. If a name requires a comment, then the name does not reveal its intent.

Don't be Cute. Choose Clarity over Humor. Say what you mean. Mean what you say.

Searchable Names. Names should be easy to locate across a body of text. If a name occurs in multiple places, it is imperative to give it a search-friendly name. Single-letter names should ONLY be used as local variables inside short methods.

Avoid Encoding. Avoid Hungarian notation, member prefixes, etc.

One Word, One Meaning. Be consistent throughout your code by using one word per abstract concept. To eliminate confusion, use words that only have one meaning.

Meaningful Distinctions. When attempting to differentiate code, make distinctions that are meaningful without changing the searchability or intention of the code. For example, don't change the spelling because the name is already taken.

Use Pronounceable Names. It is easier to remember and discuss names that you can pronounce. For example: genymdhms versus generationTimeStamp.

2 NAMING CODE ELEMENTS

Class and Object Names

- Use noun phrase names
- Don't use verbs

Method Names

- Use verb phrase names

Method and Class Length

- Long scope = Short name
- Short method scope = Long name

Solution Domains

Use programming language such as:

- Computer Science terms
- Algorithm names
- Design Pattern names

Problem Domains

- Use simple terms that clearly identify the problem
- Avoid programmer language

Variables and Parameters

- Long scope = Long name
- Short scope = Short name
- Single-letter names should be used only for:
  - Counter variables for simple for loops
  - Exception instances in Catch blocks
  - Arguments of very short functions