Pseudocode Guidelines

The class pseudocode will be presented as a text-friendly numbered list. It will contain all elements of the procedural and object-oriented programming paradigms; however, it eliminates data abstraction, modularity, and error handling. In general, the pseudocode conventions are

- Every statement represents a simple operation and is numbered.
- Indentation will indicate a block structure (body statements). Each block structure begins its numbering from 1. Once a block structure is exited, the numbering continues from where it was before entering the block structure.
- The assignment operator is `<-`
- The arithmetic operators are the standard operators (`+, -, *, /, %`).
- The logical operators are `and`, `or`, and `not`.
- The relational operators are `=`, `!`, `<`, `>`, `<=`, `>=`.
- The selection statement syntaxes are

  1. `if` *condition* `, then` *body*
  2. `if` *condition* `, then` *body* `else` *body*
  3. `if` *condition* `, then` *body* `elif` *condition* `, then` *body*

  where syntax 3 can contain multiple `elif` statements and can end with a `else` statement.
- The loop statement syntaxes are

  1. `while` *condition* `, do` *body*
  2. `for` *initialization* `to` *end* `by` *step* `, do` *body*
  3. `foreach` *variable* `in` *sequence* `, do` *body*

  where syntax 2 can omit the `by` clause which means steps are 1.
- Variables are local. They are global if they begin with an underscore.
- Arrays can access their elements with the subscript operator (`[]`) as expected with indices ranging from [0,n) where n is the size of the array. Additionally, arrays have the attribute `length` that is accessed with the dot operator (`.`) as follows

  Example:
  `A.length` provides the size of the array A
- Objects contain both fields and methods, which are accessed using the dot operator. They behave as pointers; that is, assigning an object to another makes the objects aliases. Likewise, an object can be assigned `nil`.
- All function parameters are passed by value, except for arrays and objects.