NC-LAB Pseudocode Standard
Revision 1.0

November 19, 2008

Introduction
This document describes the requirements which must be met by all pseudocode written for the Natural Computation Lab (NC-LAB). The use of this coding standard is to promote reliable, readable, and easily maintainable pseudocode.

Pseudocode must be clear and concise. The reader of any pseudocode should immediately be able to tell what the writer intended. If this is not the case, the pseudocode should be rewritten to make the algorithm more clear. This style guide will ensure that all pseudocode is consistent and readable.

Variables
1. Variable assignment must be done using a single arrow.
   Ex: \( \text{size} \leftarrow 15 \)
2. Use a single equal sign for comparison.
3. All variable names must be in title case, with the exception of the first letter of the name, which should be in lower case.
   Ex: velocity, minVelocity, maxRecordedVelocity, etc.
4. All variables must be shown in italics.

Functions
1. Function preconditions will use the boldfaced keyword \textbf{“Require”}.
2. Function postconditions will use the boldfaced keyword \textbf{“Ensure”}.
3. Function names must use title case formatting.
4. Function prototypes and return types must be within the pseudocode title.
Algorithm 1: function $\text{SumOf}(x, y)$ returns $x + y$

1: $sum \leftarrow x + y$
2: return $sum$

Loops

1. Loop invariants are displayed with the boldfaced keyword “Require” at the beginning of the loop and boldfaced “Ensure” shown at the end of the loop. Each keyword should be followed by the description of the invariant.

2. Arrays indices should start with the number 1.

3. There are four different styles of looping that are encouraged. The following examples are examples of For-loops, While-loops, Repeat until condition, and infinite loops.

Algorithm 2 For-loop example

1: for $i = 1$ to 10 do
2:  $i \leftarrow i + 1$
3: end for

Algorithm 3 While-loop example

1: while $i = 1$ to 10 do
2:  $i \leftarrow i + 1$
3: end while

Algorithm 4 Repeat until condition example

1: repeat
2:  $i \leftarrow i + 1$
3: until $i = 10$

Algorithm 5 Infinite loop example

1: loop
2:  $i \leftarrow i + 1$
3: end loop
Style

1. Comments should typically be avoided; if the pseudocode gets really complex then comment directly above the code block with C-style comments.
   Ex: //This is what a comment should look like.

2. All keywords must be in bold.

3. All constructs should be ended with “end <construct>”.

4. Indention expresses scope.

5. Every line in the pseudocode must be numbered.

6. Each line in the pseudocode should not wrap to the next line.

7. Conditional symbols should be used where necessary.
   Ex: \( \leq, \geq, \neq \)

8. All conditional statements (including the if-statement) should maintain the style mentioned in this section.

9. Arrays will use title case formatting.
   Ex: NumberArray[\text{size}]

\begin{algorithm}
\textbf{Algorithm 6} : \textbf{function} CalcExponential(\( x, n \)) \textbf{returns} \( y = x^n \)
\begin{algorithmic}
\Require \( n \geq 0 \) and \( x \neq 0 \)
\Ensure \( y = x^n \)
\State \( y \leftarrow 1 \)
\If {\( n < 0 \)}
\State \( xVal \leftarrow 1/x \)
\State \( nVal \leftarrow -n \)
\Else
\State \( xVal \leftarrow x \)
\State \( nVal \leftarrow n \)
\EndIf
\Require \( n \geq 0 \)
\While {\( nVal \neq 0 \)}
\If {\( nVal \) is even}
\State \( xVal \leftarrow xVal \times xVal \)
\State \( nVal \leftarrow nVal/2 \)
\Else \{nVal is odd\}
\State \( y \leftarrow y \times xVal \)
\State \( nVal \leftarrow nVal - 1 \)
\EndIf
\EndWhile
\Ensure \( y = x^n \)
\end{algorithmic}
\end{algorithm}