CODING STANDARDS

DONE BY:

SRUTHI.S

1. **Limited use of globals:**

   These rules tell about which types of data can be declared global and which can't be. This limits the data that needs to be defined with global scope.

2. **Standard headers for different modules:**
   The header of different modules should have a standard format and information for ease of understanding and maintenance. It should contain: Name of the module, the Date on which the module was created, the Author's name, the Modification history, the Synopsis of the module, Different functions supported, and Global variables accessed/modified by the module.
3. **Naming conventions for local variables, global variables, constants and functions:**
   Local variables should be named using camel case lettering starting with a small letter (e.g. localData) whereas Global variable's names should start with a capital letter (e.g. GlobalData). Constant names should be formed using capital letters only (e.g. CONSDATA).
4. **Indentation:**

   For making the code readable, programmers should use White spaces properly i.e. Indentation. Every nested block should be properly indented and spaced and there should be indentation at the beginning and end of every block.

5. **Conventions regarding error return values and exception handling mechanisms:**
   The way error conditions are reported by different functions in a program should be standard within an organisation.

CODING GUIDELINES:

1. **Avoid using a coding style that is too difficult to understand:**
   The code should be easily understandable. The complex code makes maintenance and debugging difficult and expensive.

2. **Avoid using an identifier for multiple purposes:**
   Each variable should be given a descriptive and meaningful name indicating the reason behind using it. This is not possible if an identifier is used for multiple purposes and thus it can lead to confusion for the reader.
3. **Code should be well documented:**
   The code should be properly commented on for understanding easily. Comments regarding the statements increase the understandability of the code.
4. **The length of functions should not be very large:**
   Lengthy functions are very difficult to understand. So functions should be small enough to carry out small work and lengthy functions should be broken into small ones for completing small tasks.
5. **Try not to use the GOTO statement:**
   The GOTO statement makes the program unstructured, thus it reduces the understandability of the program and also debugging becomes difficult.