Documentation

Let us roughly limit software documentation to that of two types: **internal** (comments in the code) and **external** (e.g. text files accompanying the code or pages inside a wiki system). Documentation is targeted at another developer which uses the code after you (or at yourself when you successfully forget everything). Below are the most important points for both types.

Internal documentation

Remember that **comments should help to understand the code.** If necessary information makes a overly long comment, it might be useful to write this information into external documentation and reference it in the comment.

Reasonable comments should include:

- **Statement of purpose** - for every file, structure, enumeration, type definition, class, public method of a class and instance variable of a class, unless it is really obvious. *Think twice of what is obvious and what is not, when you do not have context in your mind. Documenting private methods and local variables can also be very helpful.*
- **Contents of every file** at the beginning of the file, if it contains more than one entity.
- **Input and output for methods**, along with **pre- and postconditions**.
  - Preconditions are special conditions on the input of the method for it to function properly (e.g. *"fPtr file pointer points to opened file in "raw format" (see description.txt for details) and contains exactly numPoints*numAttributes float values"*).
  - Postconditions describe special conditions on the output and/or on called object after the method is finished, if the mentioned preconditions were satisfied.
- **Any complicated or tricky stuff**. Rule: more complicated - longer the comment.

External documentation

Reasonable minimum here is to create a file description.txt (name to be discussed), and briefly describe in it:

- **What** the code is supposed to do;
- **How** does the code work generally;
- **What input** does it takes;
- **What output** does it produce;
- **Which algorithms** are used.

With growth of the software complexity and importance, the next steps are to give more details on:

- Conceptual **general flow** of the program;
- **Complex algorithms** (provide explanations or references);
- **File contents for every file** (they are already inside files if you did internal documentation as recommended, you copy them here for overview);
- Most used **data structures**;
- **Global variables** (if any, and why it is necessary to have it global);
- **Major functions**, and functions which do **"the real work"**.