Documentation

Premise:

- The developer writing the code knows what the code is doing (or is supposed to do)
- Other people will [someday] have to read, understand, modify, test your code
- Documentation will help

Documentation

Two Primary Types of Documentation

(often designers and developers are not the same person.)

- External to your code
  - A feature/design specification document
  - A “wiki” type of website, document repo
- Internal to your code
  - Comments
  - Good coding habits

Documentation

What must we document?

- What is this “data”
  - – A database
  - – Transaction data from user input/screen
  - – Validation Rules
  - – Source / Destination

Documentation

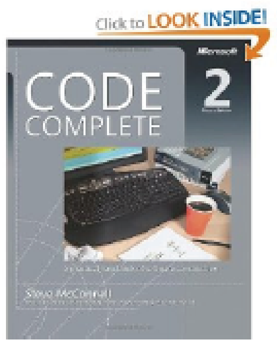
What must we document?

- What is the code doing to the data
- The meaning of variables
- Functions, methods, called procedures

Documentation

- “Self-Documenting” code
  - Program structure
  - Variable Naming
  - Class and Method Names
  - Named Constants instead of Literals
  - Minimized control flow
  - Reduced data structure complexity

Self-Documenting Code



- Code should be written for humans
  - Compiler will keep the machine happy
- Quality Comments
  - Bad comments are worse than none at all!
- Naming Scheme
  - Make a name count!
- Coding Style
  - Decide on one and enforce its use
- Documentation Extraction Systems
  - JavaDoc, Doxygen, rdoc, etc…

Effective comments **DO NOT** repeat the code!

Documentation

- Your Organization should define and enforce
  - Standards for names
    - Variables – lower case, words separated by “_”
    - Methods/Functions – CamelCase, no “_”
  - Avoid numbers as differentiators
    - grade1, grade2, grade3
  - Use a name that is self-explanatory – no comments needed
  - Standard abbreviations
    - “dept” for “department”
    - “cust” for “customer”

Documentation

- Code Structure
  - Consistent Indentation
  - Using braces {…}
  - Where to declare variables
  - Make it modular

Documentation

Fundamental Principles

- *The best documentation is the code itself.*
- *Make the code self-explainable and self-documenting, easy to read and understand.*
- *Do not document bad code, rewrite it! (Refactoring)*

Documentation

- Source code documentation generator tools
- Generate formatted, browsable, and printable documentation from specially-formatted comment blocks in source code.
- This allows for developer documentation to be embedded in the files where it is most likely to be kept complete and up-to-date.

- **Javadoc** for Java
- **Doxygen** for

  C++, C, Java, Objective-C, Python, VHDL, PHP,C#

Documentation

How it works:

- Write comments in special format
  - Include html formatting
  - Use tags to specify specific kinds of documentation
- Leave the rest to the tool

Documentation

A quick look at Doxygen:

https://www.stack.nl/~dimitri/doxygen/manual/docblocks.html#cppblock

The Real world

Internal Documentation Mention in Programmer Guidelines Doc.txt (Command Line)