Doxygen Tutorial

For : COP 3330.
Object oriented Programming (Using C++)
http://www.compgeom.com/~piyush/teach/3330

Piyush Kumar

Documentation Generators

- A **documentation generator** is a programming tool that generates documentation intended for programmers (API documentation) or end users (End-user Guide), or both, from a set of specially commented source codes.

Commenting programs

- **Doxygen** is a documentation generator for C++, C, Java, Objective-C, Python, and to some extent PHP, C# and D.
- Its highly portable. (Windows/unix/linux/mac)
- KDevelop has builtin support for Doxygen.

Why?

- Doxygen is very useful for maintaining and understanding your own larger projects as well as useful documentation for others who use your code.

How?

- For each project that uses Doxygen, you must create a configuration file.
- "`doxygen -g`" creates a example configuration file called "`Doxyfile`"
- "`doxygen [configfile]`" will create the documentation for your code as per your configuration specs.

Configuration File

- Well documented, you just need to fill in the blanks.
- Main things to set
  - `PROJECT_NAME = MyProject`
  - `OUTPUT_DIRECTORY = ./doc`
  - `INPUT = ./src ./include`
  - `FILE_PATTERNS = *.cpp *.hpp`
  - `GENERATE_HTML = YES`
  - `EXTRACT_ALL = YES`

Documenting the source.

- Beginning of file:

```
/*! \file dpoint.hpp
    \brief d-dimensional point class

    A d-dimensional point class which is written carefully
    using templates. It allows for basic operations on points
    in any dimension. Orientation tests for 2 and 3 dimensional
    points are supported using
    <a href="http://www.cs.berkeley.edu/~jrs/">Jonathan's</a>
    code. This class forms the building block of other classes
    like dplane, dsphere etc.

    \author <a href="www.compgeom.com/~piyush">Piyush Kumar</a>
    \bug    No known bugs.

*/
```

HTML allowed

Documenting the source.

- Beginning of function.

```
/*! \brief Prints character ch at the current location
 *         of the cursor.
 *
 *  If the character is a newline ('\n'), the cursor should
 *  be moved to the next line (scrolling if necessary).  If
 *  the character is a carriage return ('\r'), the cursor
 *  should be immediately reset to the beginning of the current
 *  line, causing any future output to overwrite any existing
 *  output on the line.  If backsapce ('\b') is encountered,
 *  the previous character should be erased (write a space
 *  over it and move the cursor back one column).  It is up
 *  to you how you want to handle a backspace occurring at the
 *  beginning of a line.
 *
 *  \param ch the character to print
 *  \return The input character
 */
int putbyte( char ch );
```

Creating the frontpage

- Example that creates first page of documentation: (You can add it to the main.cpp or main source code file)

```
/**
@mainpage COP 3330 Project 1

@author Me and Myself

Here you should tell us about how your project works.  How to run,
any special things you have, etc.  Also, explain any non-trivial
design decisions you make. If you are working with a partner, clearly
State what is each person's contribution. You should
also comment on the stability of your code.  Any big bugs should be listed
here.  Basically, anything that you think we need to know in general about
your project should go here.

Any additional comments you want to make can go here.  Did you like the
project?  Was it too hard, too easy?  My TA smells bad.  Well, you get
the idea.

This initial documentation here should be removed.
Or else you loose points.

*/
```

Documentation Rules.

- Each file/function should have a header block.
- Use descriptive and meaningful names for variables, constants, and functions.
- Don't just re-express the algorithm in English; tell us **why** you're doing something.

Right: For each name in the array, extract the lastname.

Wrong: Set i to 0. Loop from 0 to 10. Call strchr() on a[i], looking for the first ' ' character. Return the pointer the character immediately following the ' '.

Documentation Rules

- For each project, create a directory structure like this:

```
prj_?\
    Doxyfile
    Makefile
    README
    bin\
    data\
    doc\
    include\
    src\
```

Created using doxygen –g
Modify it after creation.

Sample project: http://www.compgeom.com/~piyush/teach/cop3330/homeworks/hw1/prj_1.tar.gz