

Document your code

There are lots of queries in this [programming forum](#) like in the following [Documentation](#) thread that

"How can I make the documentation for my coding? I want to mean what is the structure of documentation that are acceptable to clients and how can someone other than me can understand my code without going into much details about the code?"

At that time I did not have much experience with documentation and so gave some standard guidelines like

1. All variables should be commented as why they are defined and what is there use.
2. Each function should be commented as what are the input parameters and why and what is the output and some logic in the loop's and if's.
3. See the Java Doc and see how they generate the Documentation from the codes and comments and that should be more than sufficient.

But after making some research and as the need arised for me to seriously document my code, I realised that there are some documentation systems available like the one at [Doxygen](#) for one of my favorite languages, C++.

Doxygen is a documentation system for C++, C, Java, Objective-C, IDL (Corba and Microsoft flavors) and to some extent for PHP, C#, and D.

Now to make the documentation of your code its very much essential that you already used a commenting style that Doxygen understands, otherwise the documentation turns out to be pretty much meaningless. So what can you do is follow some of the guidelines that Doxygen has laid down which is more than sufficient to generate the manual / Documentation for your code. Complete list can be found at [Doxygen Manual](#) but if you are not interested in reading all the details then you can just start with the minimal documentation guidelines given below and when you think you need more time on documentation you can go through the above detailed documentation.

Doxygen has defined 'Document Blocks' for documenting the blocks of code.

I am showing you the example block which I like the most and use more often than any other blocks defined by Doxygen.

Three slashes for comments instead of the standard two used in C++. One can use an exclamation Mark instead of the first slash, here I prefer all 3 slashes.

```
///  
/// ... text ...  
///
```

The above represents a comment block.

Now there are two ways to describe your class. One is a brief description and another is a detailed description.

One could use the \brief command with one of the above comment blocks, after the 3 slashes. This command ends at the end of a paragraph, so an empty line follows. One may give a detailed description after the empty line.

Now there are also many alternatives to the brief description command but again I prefer to mention the keyword \brief because a fellow programmer may not remember all the syntaxes of Doxygen, but its easy

to understand `\brief` as a brief description of the class since all syntaxes start with `\keyword` and this is what I prefer.

```
///  
/// \brief Brief description.  
/// Brief description continued.  
/// (Empty Line)  
/// Detailed description may start here.  
///
```

I always follow the convention that all the functions should be followed with a comment block of Documentation and all the member variables should have a comment after they are declared.

```
int var; ///  
Detailed description of the member
```

Note the "<" symbol after the `///`

It helps in identifying that the comment is for the line that is previous to or as "pointed to" by the "<" symbol. Without this symbol it means that the comment is for the following line.

```
int var; ///  
Detailed description after the member for var and not var1  
int var1; ///  
Detailed description after the member for var1
```

There are many other commands and to see the complete list of them, goto [Special commands](#). I will try and explain here the ones used most.

I prefer to use the following commands

`\brief`

- Brief description of the function or the class followed by the detailed description after an empty line

`\param`

- Parameter passed to the function with a name and description

`\remarks`

- Any special comments I would like to make a note of to others or to me in the future. Something like the function is not thread safe or anything that can cause the program to fail in some condition

`\return`

- What are the return values of the function on success and on failures.

`\b`

- Make the next word as bold

`\e`

- Make the next word as italics

`\c`

- Make the next word as courier

`\code \endcode`

- Formatting of the code will be done

`\n`

- Pretty much known to all that it forces a newline

`\file`

- To summarise the content of the file

The attached zip file contains a Microsoft Visual C++ Project with the following files

stack.h -> Stack Class.

stack.cpp -> implementation of the functions defined in stack.h

main.cpp -> Main function where the execution starts.

and Documentation generated by Doxygen with Doxygen configuration File.

Source: <http://www.go4expert.com/articles/document-your-code-t394/>