Sonoma State University
Computer Science Department
CS 370 – Fall 2022 – Watts

Exercise 8

(Due: Friday, 9 December 2022, by 11:59 pm.)

In this Exercise you will be working with Doxygen to generate a website containing documentation for small software system.

> Doxygen is the de facto standard tool for generating documentation from annotated C++ sources, but it also supports other popular programming languages such as C, Objective-C, C#, PHP, Java, Python, IDL (Corba, Microsoft, and UNO/OpenOffice flavors), Fortran, and to some extent D. Doxygen also supports the hardware description language VHDL.
>
> Doxygen can help you in three ways:
>
> 1. It can generate an on-line documentation browser (in HTML) and/or an off-line reference manual (in LaTeX) from a set of documented source files. There is also support for generating output in RTF (MS-Word), PostScript, hyperlinked PDF, compressed HTML, and Unix man pages. The documentation is extracted directly from the sources, which makes it much easier to keep the documentation consistent with the source code.
> 2. You can configure doxygen to extract the code structure from undocumented source files. This is very useful to quickly find your way in large source distributions. Doxygen can also visualize the relations between the various elements by means of include dependency graphs, inheritance diagrams, and collaboration diagrams, which are all generated automatically.
> 3. You can also use doxygen for creating normal documentation (as I did for the doxygen user manual and web-site).
>
> Doxygen is developed under Mac OS X and Linux, but is set-up to be highly portable. As a result, it runs on most other Unix flavors as well. Furthermore, executables for Windows are available.

1. Copy the folder “Exercise8” from the course pickup folder to your public_html folder. Rename the folder *submissionid*E8 and use the Linux comman:

```
chmod 755 submissionidE8
```

to enable viewing of your documentation website.

2. Enter the Linux command:

```
doxygen -g
```

to generate a Doxygen configuration file.

3. In the generated file called Doxyfile, modify the following configuration parameters:

```
PROJECT_NAME           = "CS 370 Exercise 8 – First Last"
```

Where First and Last are your first and last name

```
JAVADOC_BANNER         = YES
EXTRACT_PRIVATE        = YES
```

4. Enter the Linux command:

```
doxygen Doxyfile
```

to generate a Dyoxygen website.

5. Enter the following Linux commands to allow you to view the website:

```
chmod 755 html
chmod 644 html/*
chmod 755 html/search
chmod 644 html/search/*
```

6. Using a browser of your choice, view the generated website. The URLshould be:

[https://blue.cs.sonoma.edu/~*cslogin*/*submissionid*E8/html](https://blue.cs.sonoma.edu/~cslogin/submissionidE8/html)

where *cslogin* is your login id for the blue server and *submissionid* is your submission id for this class.
Explore this web site – you should see entries for some, but not all, of the files in this folder.

7. To improve this website, replace the current flowerbox documentation in the files with Doxygen's tags and preferred format:

1. Modify the top line of each flowerbox to have a space after the second *
2. Remove all of the *s at the end of the flowerbox lines
3. Replace all occurrences of Assignment with: @brief
4. Replace all occurrences of File with: @file
5. Replace all occurrences of Author with: @author
6. Add your name as the author
7. Replace all occurrences of Date with: @date
8. Replace all occurrences of Class with: @class
9. Replace all occurrences of Function with: @brief
10. Replace all occurrences of Description with: @remarks
11. Replace all occurrences of Parameters with: @param
12. Replace all occurrences of Return value with: @returns

The documentation at the beginning of ABC.h should now be

```
/** *************************************************************************
* @brief Exercise 8 - Doxygen documentation exercise
* @file ABC.h
* @author
* @date Fall 2022
*
* @remarks This file contains the description of the ABC class.
****************************************************************************/
```

8. Repeat steps 4, 5, and 6 to review your modified website. It should now include all of your files and documentation for the ABC class.

9. Modify the file ABC.h to use Doxygen style comments for each of the variables. For example, the comment for private variable 'a' should be changed from:

```
// integer variable a contains the number of integers in c
```

to:

```
    /**
    * @var int a
    * @brief integer variable a contains the number of integers in c
    **/
```

10. Again, repeat steps 4, 5, and 6 to review your modified website.

11. If you have been observing the warnings generated by Doxygen, the only ones that should remain refer to the define macros in ABC.cpp. Add this documentation to the file before the defines.

```
/** *************************************************************************
* @def DEBUG false
*        Sets the DEBUG flag
* @def debug if (DEBUG) cout
*        Writes to standard output if DEBUG is true
****************************************************************************/
```

12. Again, repeat steps 4, 5, and 6 to review your modified website. While this is what is described on the page: https://doxygen.nl/manual/commands.html#cmddef, it will likely not get rid of these warnings. Extra credit will be given to anyone who can resolve problem!

13. Create a makfile to compile UsesABC.cpp.

14. Modify UsesABC.cpp so that it tests all of the methods in the class ABC.

Submission

To turn in: A text file called *submissionid*E8.txt. This file should contain the URL for the website containing your generated documentation, a copy of your makefile, and a copy of your modified UsesABC.cpp file.
Date Due: Friday, 9 December 2022, by 11:59 pm.)