# Outlab 6 - Doxygen & Sphinx

Due: 12.00, 28 September

This Outlab is based on Doxygen and Sphinx. Check the resources to learn more about them before starting to work. Outlab resource files can be found here.

**Follow the naming convention of the submission directory. The structure is given at the end for ease of reference. Q2 will be auto-graded. Q1 PDF/HTML will be manually checked.**

## Utilities

You can use the scripts to clean the build files and start fresh for any part. This is important if you want to avoid repeating the configuration steps of Sphinx for any question.

## Q1. DSA Library (Doxygen & C++) (10)

**(a) Separating declaration and implementation (7.5)**

This is an extension of the first question of inlab. Move all the declarations (classes and functions) to a separate header file (DSA.h) and add the Doxygen comments to that file. Modify the DSA.cpp accordingly so that it only contains the implementations of all the constructors and member functions and includes only DSA.h. The Doxygen page generated (using DSA.h) should be nearly identical to the inlab page. Also, generate and add the PDF similarly to inlab Q1.

**(b) Compiling (2.5)**

Ensure tester.cpp compiles and executes correctly.

# Q2. DSA Library (Sphinx & Python) (35 + 10B)

**IMPORTANT FOR AUTOGRADING: Follow the commenting format given in this link for Q2.**

Most of the parts remain the same as inlab/outlab Q1, which is why the marks for the comments are less. The configuration remains the same as inlab Q2(a), but the project name will be outlabq2. You need to create documentation for a data structures library (file DSA.py has been provided in outlab06res/q2) by adding detailed comments to all classes and functions. The tester.py file is only for your reference on the usage of the library. You might need to copy the corrected heap implementation from your inlab Q3 to the bottom of DSA.py.

You need to write **doctest comments** for your code as well. This means you will need to add sample code (along with the output) in a shell format within the comments. Refer to this to know more about doctest and running it. An example for doctest. The auto-grading part will be based on whether your comments can execute correctly using doctest. Tips: add 'sphinx.ext.doctest' to extensions in conf.py and use the command 'make doctest' to run tests. The doctest is like an iPython shell.

The steps (and rubrics) are given below:

**(a) Contents (2)**

The contents page should show q2 and the DSA module inside it.

**(b) Class Descriptions (2)**

Add class descriptions for each of the data structures.
The description should include the list/names of utility/member functions of that data structure.

**(c) Function descriptions (31 + 5B)**

1. Constructors and Converters (1.5 + 5)
   Add documentation for all types of constructors and converters (__str__) of each data structure class.
   It should contain a brief description, input and output parameters (if any), and a list of available functions.
   Doctest for constructors/converters: 1 * 5 (1 per data structure)
   You may have to find out how to make the constructor docstring visible on the documentation.
2. Member Functions (3.5 + 20)

Add documentation to all the member functions (insert, find, delete, print, reverse, etc.).
It should contain a brief description, input parameters, output parameters, and return details (if any).
Doctests for function calls: (4 + 1B) * 5 (at least 4 per data structure, 5th bonus)

3. Spacing (1)
   Make the functions/arguments/parameters appear in a listed form without leaving blank spaces/lines in the comments (find the way to do this).

**(d) Bonus (5B)**

1. Theme (2)
   Combine sphinx with readthedocs.
2. Modules (3)
   Copy the classes into separate modules/files, and modify the documentation such that the contents page reflects the classes.

## Submission Instructions

As usual, create a folder named <roll-number>_outlab6, and compress it with the command (run it from the directory outside the folder):

**tar - czvf <roll-number>_outlab6.tar.gz <roll-number>_outlab6/**

The file structure should be as below (you can verify using the tree package):

```
<roll-number>_outlab6
    ├── q1
    │   ├── DSA.cpp
    │   ├── DSA.h
    │   ├── q1.pdf
    │   └── tester.cpp
    ├── q2
    │   ├── DSA.py
    │   ├── make.bat
    │   ├── Makefile
    │   ├── source
    │   │   ├── conf.py
    │   │   ├── DSA.rst
    │   │   ├── index.rst
    │   │   ├── modules.rst
    │   │   └── <any other rst files you might have used for doctest>
    │   └── tester.py
    └── references.txt
```

**Note: Mention all the links you use (apart from the ones in the resources) in references.txt.**

**May the Docs be with you!**