# INFO 72220 Assignment 4

Due Date: November 13, 2020

Submission Format:

* 1 C program with Makefile

## Multithread Programming (50 marks)

In this assignment, you will use the POSIX thread library to create a simple program using master-slave thread programming model that deploys the famous “Observer Design Pattern”.

**Observer Design Pattern**

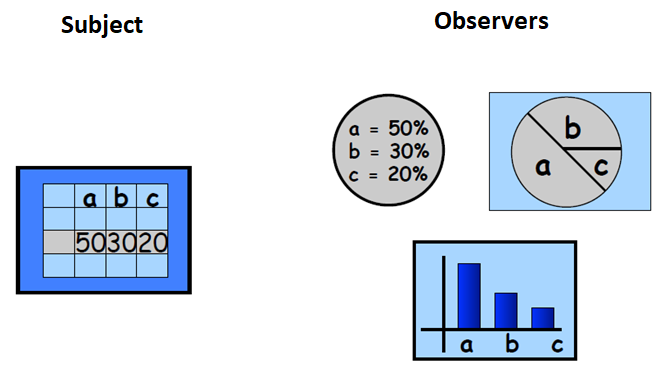
Subject

* The data source, holding the key information that is of multiple observers’ interests
* It holds all the observers that are listening to its data
* It is usually responsible for notifying the observers about the update, but in some simple implementations it will let the Observers check the data themselves (which applies to our case)

Observer

* The data consumer (often UI) that updates its displayed contents when Subject’s data changes
* It can also choose to filter the update and only delivers the required information to the user

In almost all configurations of the observer pattern, Subject and Observers are modularized. In our case, they should all be residing on different threads.



**Requirements**

1. We will deploy the master-slave thread programming model. The main thread will spawn a Subject thread, which generates random integers between 1 and 1,000 continuously, and four Observer threads (as an Observer thread pool), each of them monitoring the Subject’s data.
2. Only when the integer is divisible by 5 would the Subject thread update its “output integer”, which is the piece of data that the Observers are interested in consuming. This output integer therefore should be a global variable defined in the main thread. To avoid race condition (observers reading the data while the Subject thread tries to modify the data), you must deploy the suitable lock to protect this piece of global variable.
3. Each Observer thread will be responsible for monitoring the change of the Subject’s data, and update their outputs this way:
   1. Observer 1 – Only output the number when the data is divisible by 3
   2. Observer 2 – Only output the number when the data is divisible by 5
   3. Observer 3 – Only output the number when the data is divisible by 7
   4. Observer 4 – Only output the number when the data is divisible by 25
4. Each Observer thread will terminate after outputting 3 valid outputs
5. The main thread terminates the Subject thread and itself when there are no longer any observers alive. **Hint:** Use the pthread condition variable to achieve so.

**Marking Scheme (50 marks in total)**

1. Main Thread implementation
   1. Correct implementation of the Observer thread spawning (5 marks)
   2. Correct implementation of the pthread condition variable to track the states of the Observer threads and shut down itself and Subject thread correctly (5 marks)
   3. Correct implementation of the pthread\_mutex for data protection (5 marks)
   4. Correct implementation of the global variable(s) for inter-thread synchronization (3 marks)
2. Correct subject thread Implementation for random number generation and data output (5 marks)
3. Correct implementation for Observer 1 (5 marks)
4. Correct implementation for Observer 2 (5 marks)
5. Correct implementation for Observer 3 (5 marks)
6. Correct implementation for Observer 4 (5 marks)
7. Coding Style and Comments (5 marks)
8. Makefile (2 marks)