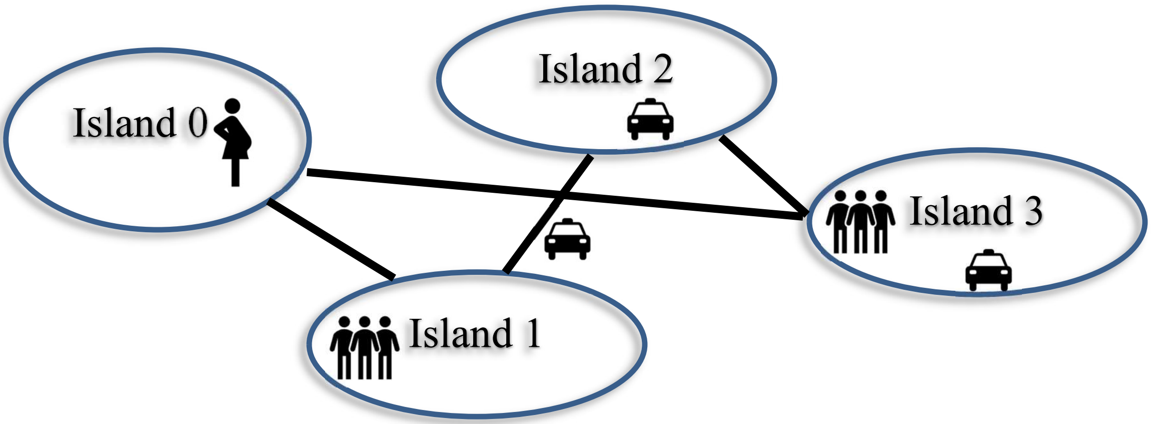
CSCM98 Coursework
Taxi Driver Coursework

Deadline: 20th of November, 11am

Coursework Submission:

To work on this project, you will have to download the visual studio solution available on the course website. For the final submission, you will need to send the *main.cpp* (and only this) file by email to b.mora@swan... All code comes in a single file so that testing can be done uniformly. You will receive an automatic email that you must both reply to and attach your submission to it. Please follow the submission guidelines carefully as any deviation from them will result in a loss of marks. Note that the work submitted must be your own work and you should not help or receive help from others. If you use extra material from somewhere else, please reference it through comments in the code.



Description

A number of people live on a set of islands connected by bridges. These people want to book a taxi trip from an island to another, and the run terminates when all trips have been performed. There are a set of conditions that need to be respected:

- There are *P* people per island initially.
- There are *N* islands with *B* bridges, with *B*>*N*. Note that to ensure a path exists between all islands, a path will always exist between island *i* and island *i*+1.
- Bridges have two single-lane tracks, and therefore only two taxis at a time can go on a bridge, whatever the direction is.
- Taxis pick customers when they are on an island if they have space available. They are restricted to 4 customers max.
- *T* Taxis are available and are initially located randomly. Taxis are implemented with threads.
- Taxis pick the next bridge to visit randomly.

Part of the code is already available to you and comes with definitions and an initial implementation of the problem described here above. You will have to complete the following tasks to make it work with the rules but try not to change the parts already implemented for you and try to understand the code first.

Completing the implementation of the semaphore class (35 marks)

You will use semaphores wherever you can to implement concurrency. As such, the constructor, *P*() and *V*() functions of the semaphore need to be completed.

Accessing resources in mutual exclusion (30 marks)

There are various resources in this problem like islands and bridges that would need to be accessed in mutual exclusion, possibly using semaphores. Complete the program and put the right mutual exclusion tools in place so that the rules established above are safeguarded. Functions to be completed include `Island::DropOnePassenger`, `Island::GetOnePassenger`, `Taxi::DropPassengers`, `Taxi::CrossBridge`.

Increasing the throughput per lane (20 marks)

The mayor of these islands, being annoyed of the regular traffic jams between the islands, has asked your experienced-with-traffic-jam lecturer to come up with an idea on how to improve the traffic. Ben quickly calculated that each track of each bridge could allow up to 4 cars going in the same direction at a given time. Implement this solution to discover if your lecturer was telling fairy tales to the mayor to get unfairly promoted. You will want to implement a second *CrossBridge* function for this on top of the first one.

Termination Criterion (10 marks)

Indicate as a comment in the code while the thread termination criterion algorithm does not need to be placed inside a critical section.

Code running and Submission Guidelines (5 marks)

5 marks will be given if the submission follows guidelines cited above and does not involve extra work (e.g. correcting code).

Notes:

- Any partial solution may return only partial marks. The exact marks will depend on the quality of the answers and can only be evaluated upon reception of the coursework.
- This is a **strictly personal piece of work**. Any external help must be referenced as comments in the code.
- Add your student ID and Name to the code.
- You will receive an email detailing how you should submit your work soon.
- Do not change any part of the code that does not require some changes. This may impact automatised tests.