# COMP3230A Assignment 3

Gao Chang 3035141748

Reason For Working

1. Concurrency control and Coordination:

We use a *philosopher* struct to store philosopher information and an integer array to store position of fork holders. We use semaphores to control each fork, the philosopher status and fork holding status. These prevent concurrent changes and serve as safe coordinate tools.

2. Deadlock and Starvation Prevention:

We use the strategy below:

> **if** position of philosophers is odd **then**
>> pick the left fork
>> pick the right fork
>
> **else**
>> pick the right fork
>> pick the left fork
>
> **end if**

This breaks the *Circular Wait Condition*. Every odd-even pair will only compete for one fork first, and there will be no case when a loop of philosophers holding fork on the same side first and waiting for the other side.

Also, if a philosopher is waiting, since there is no deadlock, it will wait for the left and right philosopher to release their forks and will not be starved.