呂文楷

謝謝老師，我好喜歡這堂課。我覺得這是我大學以來最好的一堂課。

我覺得這堂課有以下幾個優點，以及我為什麼喜歡：

1. 內容本身：上完這堂課，我驚覺我以前寫的程式根本都是亂寫。有了老師教的這些基礎，以期這學期的訓練，我現在有辦法看程式碼了，也有 design pattern 初步的概念，對於如何把程式寫好也開始有一些方向了。
2. 人數少：這堂課的人數偏少，很慶幸我這學期能選到這堂課，但也因為人少，所以老師可以給大家比較多的回饋。
3. 小組討論：我覺得這是這堂課很厲害的地方，每次作業都可以在小組討論中，從不同的看法去設計程式架構，也理解到每個人對於 design pattern 的詮釋方向皆不同，每次討論都會有很多收穫。
4. 上課練習：老師上課後講完一個 pattern，就會讓我們去練習，我覺得這是一個很好的模式，剛學完馬上應用，記憶會變得更加深刻。
5. Trace code：這是我第一次花大量的時間去看 source code，我覺得這個時間花得很直得。我負責 eureka 的部分，也發現原來一個 service registry 其實會需要處理超多問題，我也很佩服寫出這些程式碼的人，他們真的好厲害，期許我自己有一天也能寫出這樣好看且有用的程式碼。
6. 老師的回饋：謝謝老師願意花大量的時間與精力去跟我們說我們每一組報告的問題，雖然我上課當下常常聽不懂，但是回去多想幾次後，常常有豁然開朗的感覺，還好有老師的提點，不然我會一直抱有錯誤的觀念。

以下是我花了一點時間才開始習慣的地方：

1. 小組討論：初期討論 design pattern我真的不知道我在幹嘛，直到後來我把 gof 關於 design pattern 的部分看完，才有辦法在討論中產生一些關於 design pattern 的論點。我必須很慚愧的說，作業的 bridge pattern、iterator pattern 和 decorator pattern 這三次我們其實用老師的 process 做不出來好的 refactor design，是去翻書有了一部分概念後，才有辦法套用 design pattern，我覺得我可能還要更多的練習才能熟悉的應用老師的 process 來產生 design pattern。
2. Trace code：學期剛開始有好幾個禮拜都不知道在看什麼，一個 pattern 都看不出來，我大概看了 5000 行後，（也可能是老師教了比較多後），才有辦法開始看出 pattern，這時往回看也才發現過去沒有看到的 pattern。我其實到最後一個禮拜，才發現 eureka 厲害的地方是在於他的分散式架構以及availability， 我之前有點抓錯重點了。
3. 上課時間：早八對我有點太早了，我有時候會不小心睡過頭，每次都很後悔，但後來請同學叫我起床後，就沒這個問題了。
4. 作業難度：作業的 spec 很單純，很多時候我們花了很多時間來想像這個題目的應用場景，但有時候想錯了，就會導致結果偏差，也就是可以完成 spec 上的東西，但實際解決的問題並不是題目要我們解決的，希望作業可以有更詳細的 spec，或是複雜一點，讓我們可以比較好去體會 design pattern 的問題。像是期中考的複雜程度，就讓我們發現 COR 的好處，尤其是當我們這組後來又重新寫了一次後。
5. DevOps：Devops 很重要，但我覺得這堂課可以不用報告 devops，把時間拿來練習更多 design pattern，因為網路上已經有很多資源，或是去實習工作後，就可以更實際的操作到這些流程與架構。
6. 希望有18週。