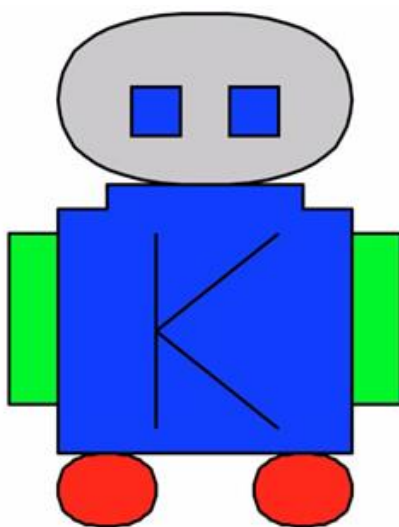


Assignment1

Based on the Assignment 1 of CS106A at Stanford University



作業檔案下載

這份作業將訓練各位同學寫程式的基本技能：寫出容易閱讀的格式 (**Style**)、清楚的使用說明 (**Comment**)、正確的空格 (**Indentation**)、以及寫程式最有趣的 - **Debug**. 本次作業全部都可以用上課教的概念完成（估計需要時間為 10 小時）

繳交期限 - 2020/10/8(四) 23:59

請勿使用未在上課討論過的 **Python** 指令

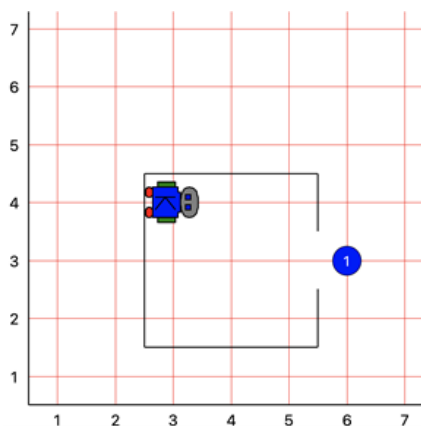
如果作業卡關歡迎各位到社團提問，也非常鼓勵同學們互相討論作業之**概念**，但請勿把 **code** 給任何人看（也不要將程式碼貼在社團裡）分享妳/你的 **code** 會剝奪其他學生獨立思考的機會，也會因此讓其他學生的程式碼與你/妳的極度相似，讓防抄襲軟體認定有抄襲嫌疑

如果真的有 **code** 方面的問題，可以在計概課堂下課詢問或是 利用 **TA 時間** 詢問 Jerry 或助教。只有他們可以看您的 **code**

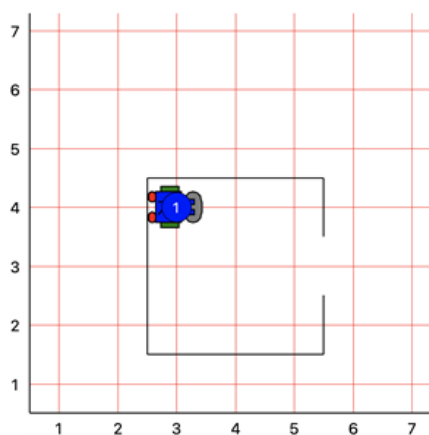
請大家 按照順序 完成本次作業。換句話說，請先完成 **Problem 1, Problem 2**, 再完成 **Problem 3, Problem 4**. 本次作業需要大量的時間思考，請大家記得 **Decomposition** 的重要，並將 **Algorithm** 寫在紙上，再按照語法填進程式裡.

每寫一個 **function** 就馬上測試一次！千萬不要等程式全部建造完畢才測試

Pre-condition



Post-condition



Problem 1 - CollectNewsPaperKarel

開始撰寫每一份程式碼之前，請先自己寫一段屬於自己的程式使用說明 - **Comments**.

將 **def main()** 裡面的 **TODO:** 刪去，再重新用自己的想法撰寫這個程式的概要（請用英文）。之後的每一題也請同學記得做這件事

Karel 一開始在家的西北方。請同學操控 **Karel** 去家門外拿報紙（也就是我們的 **Beeper**，位置永遠在 **Street 3, Avenue 6**）並讓 **Karel** 回到原位、面向東方、放下報紙。

三件事情 請同學注意：

1) 請寫兩個 **Functions** 練習 “*decomposition*”

* **Function (1)** - 移動到報紙

* **Function (2)** - 將報紙帶回家閱讀

2) 每一個自己創建的 **Functions** 也都要寫上 **Comments**

3) 沒錯，這題就是這麼簡單！千萬不要 **overthinking**

Problem 2 - StoneMasonKarel

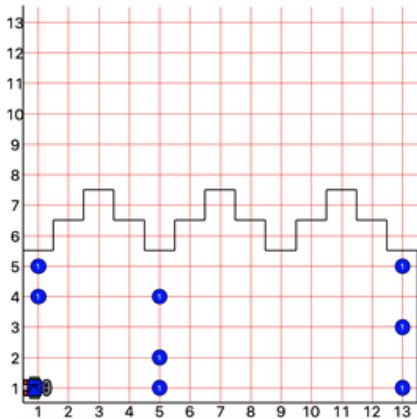
台灣政府雇用了我們的 **Karel** 來維修因 921 大地震而坍塌的拱門，也提供 **Karel** 無限的 **Beepers** 原料來完成這項艱鉅的任務。你們的任務就是操控 **Karel** 並以 **Beepers** 將每一個拱門兩旁的柱子填滿，讓結構變得非常牢固！

如左圖所示，當你們完成這份作業，每一個拱門兩旁的柱子都會被 **Beepers** 填滿，且 **Karel** 最後所在的位置會在 **Street 1**，面對東邊。

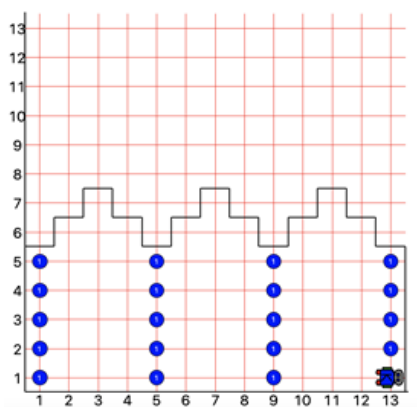
每一個拱門的大小都一樣、每一根柱子與柱子之間的距離永遠都是差三格。然而，我們會用不同的 **Worlds** 測試你們的程式。舉例來說，某一個 **World** 有 5 個拱門、6 根柱子，每一根柱子根柱子中間的距離都差 3 格（如下圖所示）。最後，五點重點提醒：

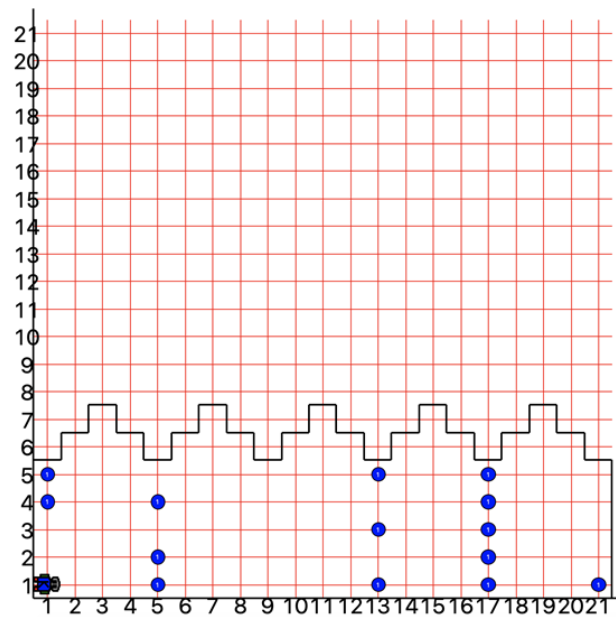
- * **Karel** 起始位置永遠是 (1, 1) 面向東方
- * 起始位置永遠有一根柱子（柱子左邊是牆壁）
- * 結束位置也都會有一根柱子（柱子右邊是牆壁）
- * 柱子缺的 **Beeper** 是隨機的，您的程式要能填滿任意柱子（缺一個、缺五個、...）
- * 請勿在「已經有 **Beeper**」的地方放上 **Beeper**
- * **Karel** 不知道有幾根柱子要修理，但它知道最後一根柱子的右邊是牆壁

Pre-condition



Post-condition

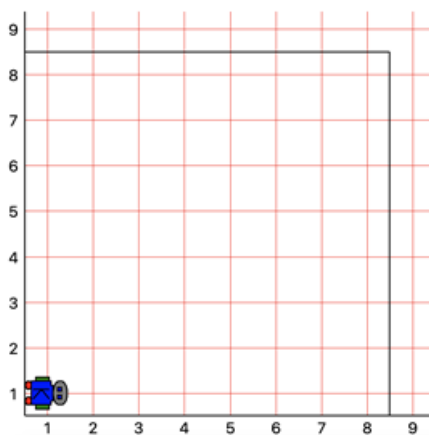




Problem 3 - CheckerboardKarel

Karel 是一名公共藝術家，它想使用 **Beeper** 為它的世界上色。因此他想到一個點子：用 **Beeper** 把世界彩繪成棋盤樣式！

Pre-condition

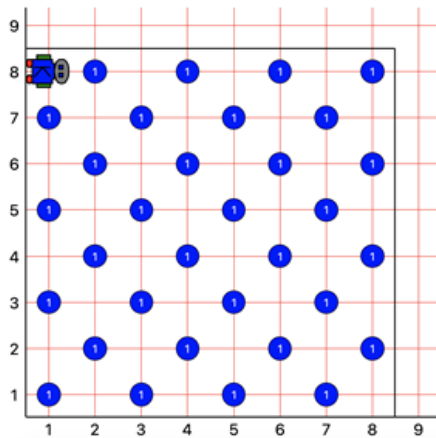


你/妳的任務是操控 **Karel** 把任何長方形、正方形的世界用 **Beeper** 間格排列 - 一個有下一個沒有 - 如圖所示

Karel 一開始所在的位置為 (1, 1) - **Street 1, Avenue 1** - 面向東邊 (**facing east**)

將世界彩繪成棋盤狀的第一步是在 (1, 1) 放上 **Beeper** ! 之後的工作，若以 8*8 的世界為例，**Karel** 會在 **Street 1** 的 (1, 3) 與 (1, 5) 與 (1, 7) 放上 **Beeper**、**Street 2** 的 (2, 2) 與 (2, 4) 與 (2, 6) 與 (2, 8) 也都會被放上 **Beeper**. 這份作業並不會要求 **Karel** 最後停在哪一格也 不會要求 **Karel** 最後要面對哪個方向.

Post-condition



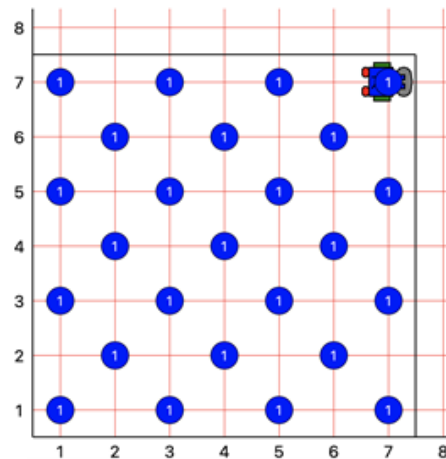
以下四點建議提供給各位同學：

* 請先嘗試讓 **Beepers** 在 **world 8*1** 間隔排列
(把 **code** 寫成一個 **function**, 如 **fill_one_line()**)

* 嘗試 **world 8*8**

* 嘗試 **world 7*7** (如左圖所示)

7*7 Post-condition

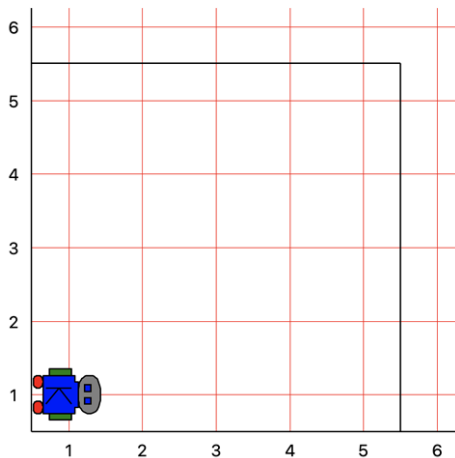


* 嘗試 **world 1*8**

這題困難之處並不在於程式本身，而是在想法。就算一位軟體工程師寫這份作業，她/他一定也要花上至少一小時。因此，如果同學卡關，請不要感到失落！試著把想法寫在紙上，想想看行不行得通，之後再把 **Algorithm** 填到 **PyCharm**

Problem 4 - MidpointFindingKarel

Pre-condition



最後一題，是這份作業看起來最簡單卻也非常有挑戰性的。我們將使用上課學過的指令操作 **Karel** 去找到 **Street 1** 的中點在哪裡

Karel 一開始位於 (1, 1) 並面向東邊。假設我們的世界是 5*5，那妳/你的程式最後要讓 **Karel** 停在 (1, 3)（面向哪邊都可以），並放置一個 **Beeper**（如左圖所示）

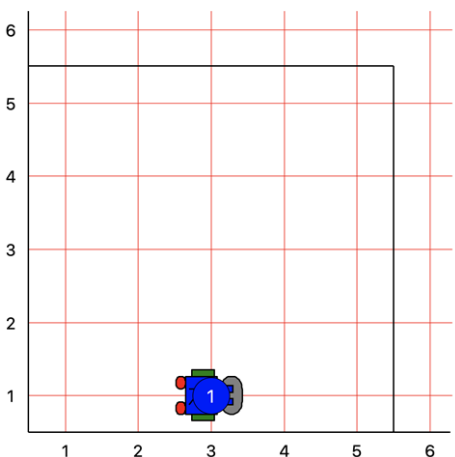
我們的世界如果是偶數（10*10），最後不管讓 **Karel** 以及 **Beeper** 停在 (1, 5) 或是 (1, 6) 都可以

下列四點重點提醒：

- * **Karel** 擁有無限多的 **Beepers**
- * 我們的世界一定是正方形且沒有任何阻礙
- * 請勿使用 **Python** 變數

這題有很多不同的解，希望同學們能發揮創意！先把任何可行的演算法寫在紙上，再一步一步把程式建造起來

Post-condition



評分標準

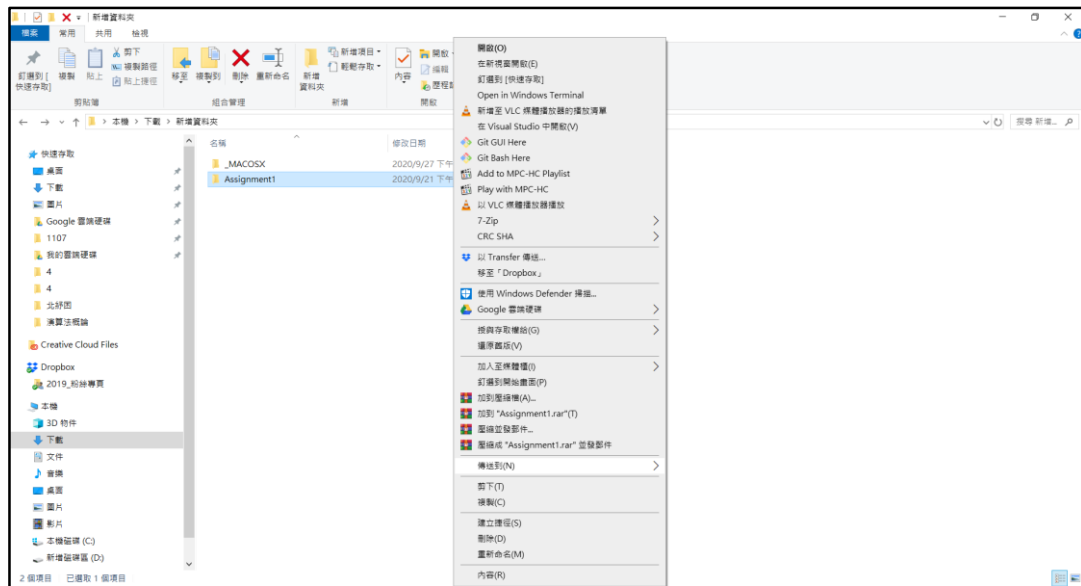
Functionality - 程式是否有通過我們不同世界的要求？程式必須沒有 **bug**、能順利完成指定的任務、並確保程式沒有卡在任何的無限環圈（**Infinite loop**）之中。

Style - 如同我們在課堂上所說，好的程式要有好的使用說明，也要讓人一目瞭然，這樣全世界的人才能使用各位的 **code** 去建造更多更有趣的程式。因此請大家寫**精簡扼要**的說明、**function** 敘述、單行註解（簡單觀念不用說明）、**function** 名稱的命名、還有最重要的 - **Scope** 是否有對齊（**Indentation**）。

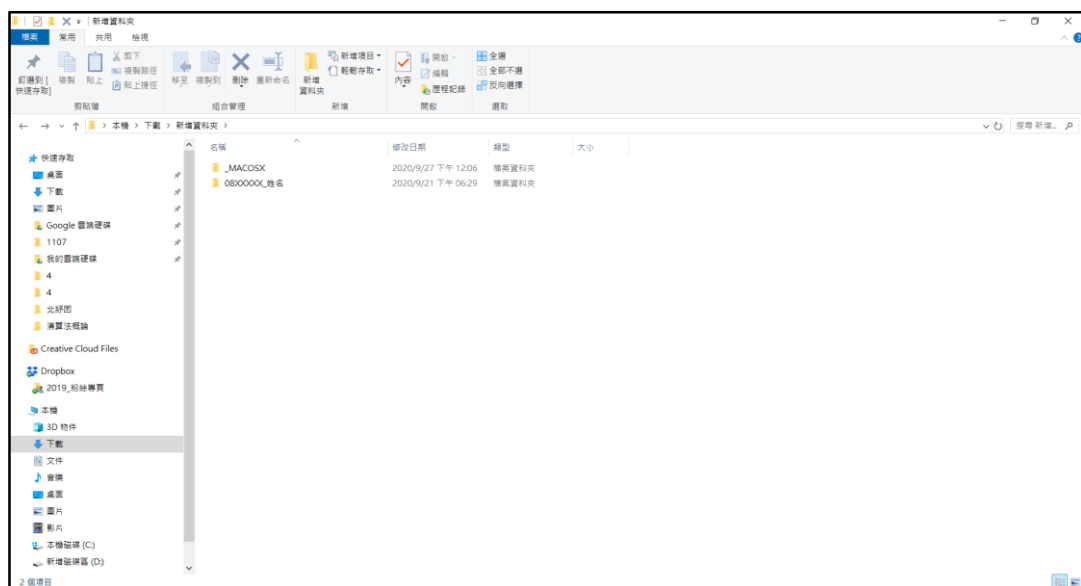
作業繳交

恭喜各位完成 **Assignment 1** 大家應該要對自己的成就感到驕傲，因為這份作業跟史丹佛大學的學生作業非常相似，代表各位成為世界最強的一群了！

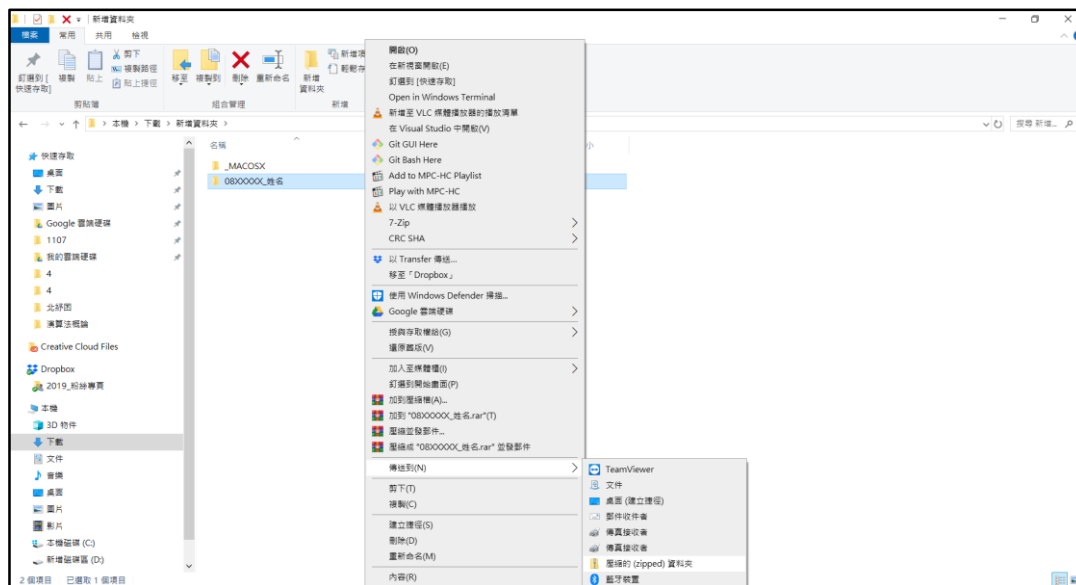
請同學於 **10/8(四) 23:59** 前依照下圖將作業上傳



找到作業資料夾，按右鍵，選擇重新命名

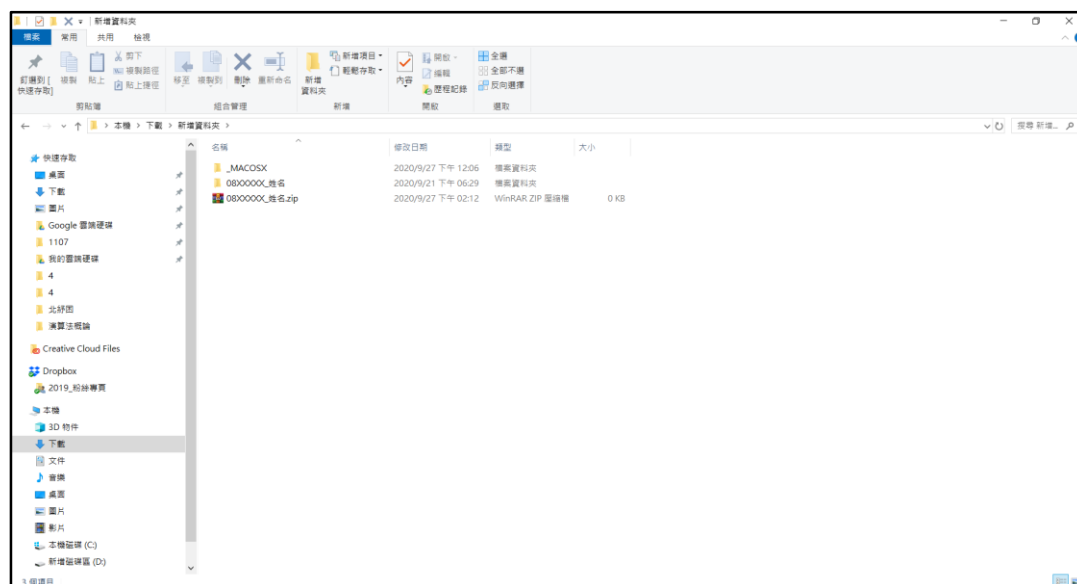


請命名成「學號_中文姓名」的格式



Windows 請點選「傳送到」->「壓縮的(zipped)資料夾」

Mac 請點選 **Compress "學號_中文姓名"**



成品如上

國立交通大學 數位教學平台

課程資訊

- 課程綱要
- 成員
- 公告列表
- 我的郵件

內容

- 大綱列表
- 同步教室 (QC3)
- 教材列表
- 作業列表**
- 討論區列表
- 試卷列表

評量

- 成績

工具

- 返回我的角色

【109上】1308計算機概論 Introduction to Computer Science

進行中作業

名稱	開始時間	結束時間	繳交狀態	功能
Assignment 1-Test	2020/09/21 00:00	2020/09/28 00:00	0 個已繳 55 個未繳	[檢視]

已繳作業

逾期未繳作業

預告作業

點選「檢視」

國立交通大學 數位教學平台

課程資訊

- 課程綱要
- 成員
- 公告列表
- 我的郵件

內容

- 大綱列表
- 同步教室 (QC3)
- 教材列表
- 作業列表
- 討論區列表
- 試卷列表

評量

- 成績

工具

- 返回我的角色

【109上】1308計算機概論 Introduction to Computer Science

Assignment 1-Test

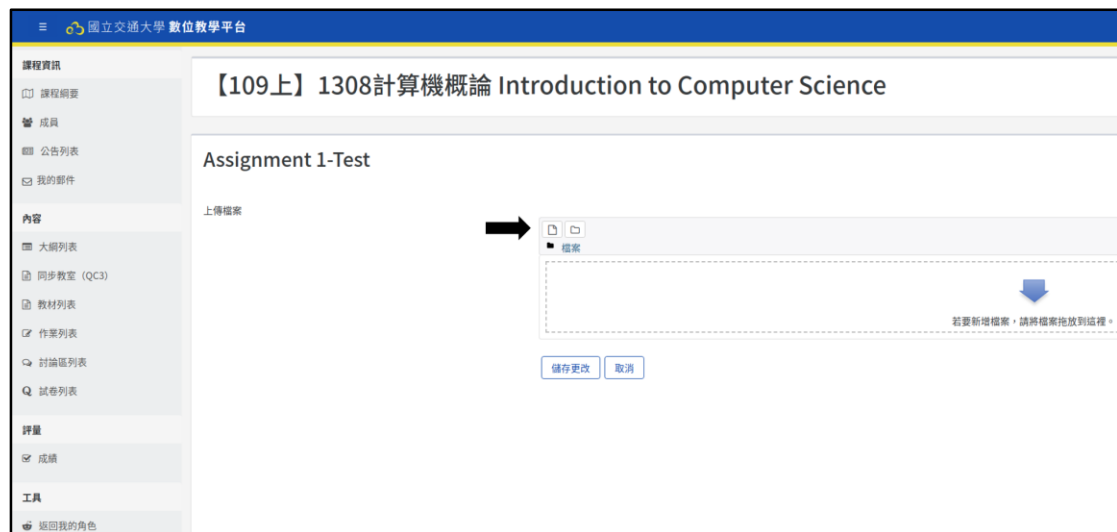
繳交狀態

作業繳交次數	這是第1次繳交
繳交狀態	沒有繳交作業
評分狀態	尚未評分
截止日期	2020年 09月 28日(Mon) 00:00
剩餘時間	6 日 5 小時
最後修改	-
作業加備註	評論 (0)

繳交作業

更改你所繳交的作業

按下「繳交作業」



按下「檔案圖示」

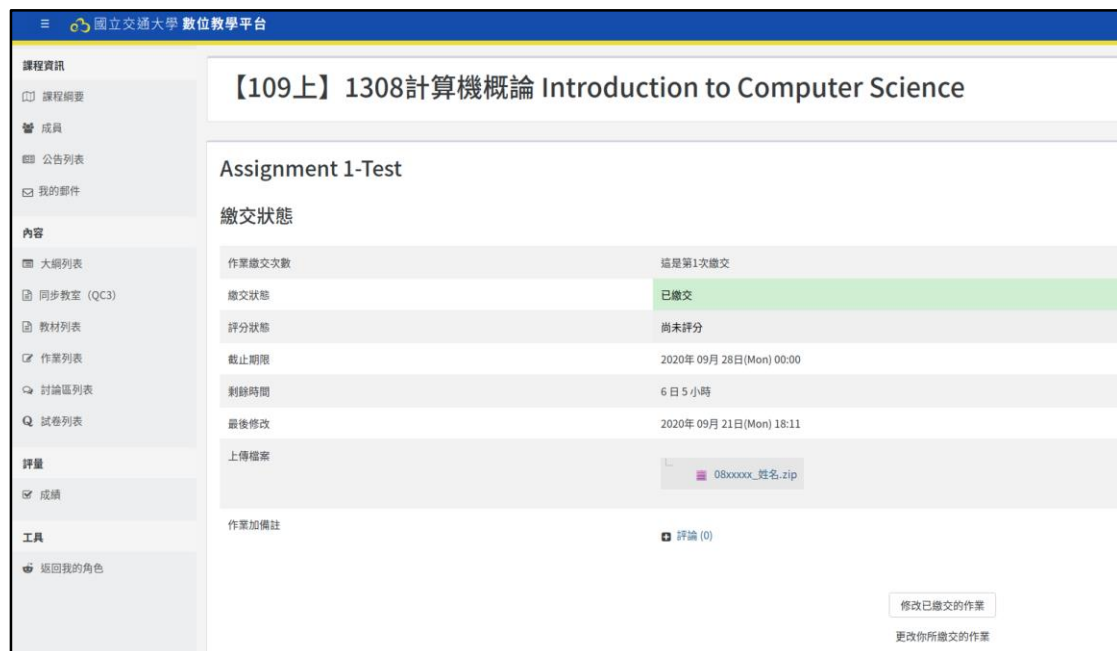


選擇先前壓縮好的作業檔案

按下「上傳這一檔案」



按下「儲存更改」



出現「已繳交」確定成功

stanCode

stanCode - 標準程式教育機構

Should you have any questions please feel free to contact us.