

## Homework 2 出缺席管理系統

Deadline: 2021-10-25 23:59:59

### 前言

由於疫情因素，導致許多學生須透過線上教學系統學習，但仍須記錄學生的出缺席狀況。然而考慮到學校的網站有時會出現流量塞車的情況，因此希望可以額外設計一個簡單的出缺席管理系統

這個作業希望同學以 Java 程式語言幫忙建立一個出缺席管理系統，讓使用者能夠輸入指令來記錄、編輯以及列出學生出缺席情況

### 作業需求

#### ■ Functional Requirements

- 使用者可以藉由輸入指令來進行記錄、列出學生出缺席另外要能夠計算出席、缺席、遲到和早退的比例
- 希望可以記錄至多八個學生且每個學生至多十週的出缺席狀況，每一週紀錄一次
- 另外程式必須注意一旦學生該週被標記為缺席，則不可以同時有出席、遲到以及早退的紀錄。也就是不可以該學生在缺席的情況下還有遲到或早退的紀錄

### 基本指令集

指令格式	指令功能描述	備註
<b>edit 人 week -p/-a/-l/-e 1/0</b>  範例 1： <b>edit 小明 1 -p 1</b> 範例 2： <b>edit 小明 1 -p 0</b>	這個指令是登記某一個人某一禮拜的出缺席狀況。  以範例 1 來說，就是登記小明在 Week 1 為出席。 範例 2 則是刪除。	1. 週數的登記為 1 到 10。 -p：出席 -a：缺席 -l：遲到 -e：早退
<b>search week 人</b> 範例： <b>search 1 小明</b>	要能列出被查詢人在該週的出席情況顯示情況。  以範例來說，就是尋找小明在第一個禮拜的出席、遲到與早退狀況。	
<b>--help</b>	列出所有指令集，並說明各個指令的功能。	
<b>search 人 -p/-a/-l/-e rate</b>  範例： <b>search 小明 -p rate</b>	列出指定對象的總出席率/總缺席率/總遲到率/總早退率。  以範例來說，就是查找小明	-p：出席率 -a：缺席率 -l：遲到率 -e：早退率

	的總出席率。	
show all	顯示所有學生在所有週次的出缺席記錄。	
add new week	新增一週的紀錄。	1.初始可登記週數為 8，執行完 add new week 後，可登記週數增加至 9 2.新增的週次必須放在最後面且不得影響到前面的數值，包含出缺席率
del last week	刪除最後一週的紀錄，包含未登記的週次。	
show weeks	顯示目前的紀錄週次，包含尚未登記的週次，也就是目前可以登記的週次的最大上限值。	
exit	退出系統，結束指令輸入。	

[備註]：

1. 所有指令均是**小寫或阿拉伯數字**，指令和指令中間以半形符號的**空白鍵區隔**。
2. 在批改作業時，助教會測試程式各個指令是否能正常執行來給予分數，但指令間是具有相依性的，例如需要有 **edit** 功能，才能有 data 對 search 功能進行測試，需要有 **add new week** 的功能，才能測試 del 跟 show 功能，因此同學在撰寫作業時，對於這兩個功能要特別留意是否能正常執行，否則會影響到其他功能的給分。

#### 小建議:

1. 進行本作業時，建議同學先思考要如何運用 Array 儲存資料，如何從 Array 中取用資料；哪些資料要放一起，哪些要分開放，更動資料時哪些資料要一起更動，這個部分可以運用 object oriented 的概念，幫助同學更好的管理有關聯性的資料。
2. 指令的輸入可以運用 String 的 Method: split() 實作，請參考 Additional Doc

#### ■ Non-functional Requirement

- 提供完善、人性化、易閱讀的使用者介面
- 當使用者輸入指令後，需有相對應的回應，提供良好的使用者體驗
- 由於作業評分需要，**不可**擅自增加額外的停頓點或延遲效果

#### ■ Execution Flow (Use Case)

1. 執行程式後，系統顯示歡迎訊息，同時提示使用者可以輸入的指令集。
2. 使用者輸入欲使用的指令。
3. 在 2. 中使用者若使用 edit 功能，系統需顯示使用者是否 edit 成功，若成功則需要顯示 edit 後的結果，失敗則需顯示失敗原因。
4. 在 2. 中使用者若使用 search week 人，系統需顯示使用者是否 search 成功，若成功則需要顯示 search 結果，顯示被 search 對象是否出席，如果有遲到或早退的情況，也需顯示出來，失敗則需顯示失敗原因。

5. 在 2. 中使用者若使用 search 人 -p/-a/-l/-e rate，需顯示使用者是否 search 成功，若成功則需顯示 search 結果，失敗則需顯示失敗原因。
6. 在 2. 中使用者若使用 --help，系統必須列出指令集並說明各指令功能。
7. 在 2. 中使用者若使用 show all，系統必須要顯示相對應的結果。
8. 在 2. 中使用者若使用 add new week，系統必須要 add 是否成功，若成功要顯示當前有多少週的紀錄，其包含未登記的週次，若失敗要顯示失敗原因。
9. 在 2. 中使用者若使用 del last week，系統必須要 del 是否成功，若成功要顯示當前有多少週的紀錄，其包含未登記的週次，若失敗要顯示失敗原因。
10. 在 2. 中使用者若使用 show weeks，系統必須要顯示相對應的結果。
11. 在 2. 中使用者若使用 exit，退出系統。
12. 詢問使用者是否需再次輸入指令(1 or 0)，若為 1 則回到步驟 2，若為 0 則退出系統。

## 額外加分項目

- 在不破壞執行流程的限制下，可為此系統加入新功能或新特色以獲得額外加分，但需在 [readme.txt](#) 中明確說明，包含如何觸發或使用等等。
- User Interface 排列整齊且精美者，將能獲得額外加分。

## 作業繳交方式

請參考計概網站上的「[作業繳交說明與規範](#)」，透過 HW2 欄位的[上傳作業](#)繳交檔案。作業相關檔案分為以下四部分，括號中的數字為該部分的配分比：

1. **程式原始碼檔案 (70%)**：以 JAVA 根據作業要求撰寫而成的檔案，主程式檔名為「**hw2.java**」，程式需撰寫註解幫助別人看懂你的程式碼，若同學的程式執行時需要用到其他檔案，如「ConsoleIn.java」，也需作為程式檔案的一部分一起繳交上來。
2. **文件檔案(27%)**：檔名為「document.doc」、「document.docx」或「document.pdf」，文件需包含以下內容：
  - (1) 需求描述：描述使用者在使用此程式時可能會有何需求？你設計程式時如何考慮這些需求？程式中有哪些地方特別吸引使用者？
  - (2) 程式流程：說明程式進行流程，建議使用利用流程圖或是其他有助說明的圖示來幫助說明程式整體上如何運作
  - (3) 功能/邏輯說明：說明自己的程式裡，自己針對什麼樣的「功能」特地設計了什麼樣特別的「邏輯」來達成？建議用畫圖的方式來說明
  - (4) 使用說明：以教導使用者的角度，說明程式要如何執行使用或是需特別注意的點...等等。（不可只寫「看了就會使用」之類的話。）
  - (5) 其他：任何有助於別人了解或使用你的程式的說明。切勿流於閒聊
3. **readme.txt (3%)**：其中必須描述以下內容
  - (1) 作者：系級、修課班別、學號、姓名

- (2) 程式檔案名稱：用來完成作業需求的程式檔案名稱
- (3) 使用說明：以教導使用者的角度，說明程式要如何執行使用，操作注意事項、附加功能以及其觸發方式...等等。

4. TimeLog.doc：記錄此次作業撰寫時間。(「**確實**」填寫該表的同學可獲加 5 分)

### 特別注意事項

- 1. 繳交作業請使用計概網站上傳，若有相關問題請參考計概網站上的 [作業繳交說明與規範](#)。
- 2. 請務必確保檔案正確上傳，若以錯誤按鍵進行上傳，進而導致作業檔案放錯資料夾、檔名大小寫或拼寫有誤，將可能造成作業無法正確批改。因此，出現此情形將酌情扣分
- 3. 作業請勿抄襲，所有作業皆會經過程式比對判斷是否抄襲，若發現一律以零分計算