

# Acceptance Test Cases :

: (表示螢幕顯示) (紅字代表使用者輸入)

1. : Enter ID or insert 'Q' to exit:

Q

: Quit

2. : Enter ID or insert 'Q' to exit:

1234

: Invalid ID

Enter ID or insert 'Q' to exit:

3. : Enter ID or insert 'Q' to exit:

955002056

: Welcome 許文馨

Insert command:

G) Show Grade

R) Show Rank

A) Show Average

W) Update Weight

E) Exit

>

- (1) E

: Exit

Enter ID or insert 'Q' to exit:

Q

: Quit

- (2) G

: 許文馨's grade:

lab1: 88

lab2: 92

lab3: 88

mid-term: 98

final exam: 91

total grade: 93

Insert command:

G) Show Grade

R) Show Rank

A) Show Average

W) Update Weight

E) Exit

>

E

: Exit

Enter ID or insert 'Q' to exit:

Q

: Quit

(3) R

: 許文馨's rank: 9

Insert command:

G) Show Grade

R) Show Rank

A) Show Average

W) Update Weight

E) Exit

>

E

: Exit

Enter ID or insert 'Q' to exit:

Q

: Quit

(4) W

: Old Weights:

lab1: 10.0%

lab2: 10.0%

lab3: 10.0%

mid-term: 30.0%

final exam: 40.0%

New Weights:

lab1: 20

lab2: 20

lab3: 20

mid-term: 20

final exam: 20

Check new Weights:

lab1: 20%

lab2: 20%

lab3: 20%

mid-term: 20%

final exam: 20%

Is it correct?(y/n)

y

: Insert command:

G) Show Grade

R) Show Rank

A) Show Average

W) Update Weight

E) Exit

>

E

: Exit

Enter ID or insert 'Q' to exit:

Q

: Quit

(5) W

: Old Weights:

lab1: 10.0%

lab2: 10.0%

lab3: 10.0%

mid-term: 30.0%

final exam: 40.0%

New Weights:

lab1: 20

lab2: 20

lab3: 20

mid-term: 20

final exam: 20

Check new Weights:

lab1: 20%

lab2: 20%

lab3: 20%

mid-term: 20%

final exam: 20%

Is it correct?(y/n)

n

: Insert command:

G) Show Grade

R) Show Rank

A) Show Average

W) Update Weight

E) Exit

>

E

: Exit

Enter ID or insert 'Q' to exit:

Q

: Quit

(6) A

: Average grade:89

Insert command:

G) Show Grade

R) Show Rank

A) Show Average

W) Update Weight

E) Exit

>

E

: Exit

Enter ID or insert 'Q' to exit:

Q

: Quit

(7) X (or other command except 'G', 'R', 'A', 'W', 'E')

: Invalid Command

Insert command:

G) Show Grade

R) Show Rank

A) Show Average

W) Update Weight

E) Exit

>

E

: Exit

Enter ID or insert 'Q' to exit:

Q

: Quit

# User's Manual 使用手冊

此成績系統將記錄每一個學生的該課成績，包含 lab1, lab2, lab3, mid-term, final exam，並依照權重(預設值為 0.1, 0.1, 0.1, 0.3, 0.4)算出總成績並建檔，當算出所有人的總成績後，也會自動排列所有人在當前權重下的名次。使用者可以先輸入 ID，並接著輸入相關指令來查詢該 ID 的成績或更改權重。系統可處理兩種異常 1)輸入錯誤 ID 2)輸入不正確指令

系統使用範例如下：

螢幕顯示：Enter ID or insert 'Q' to exit:

使用者輸入：955002056(許文馨的 ID)

螢幕顯示：Welcome 許文馨

Insert command:

- G) Show Grade
- R) Show Rank
- A) Show Average
- W) Update Weight
- E) Exit

>

使用者輸入：G

螢幕顯示：許文馨's grade:

lab1:	88
lab2:	92
lab3:	88
mid-term:	98
final exam:	91
total grade:	93

然後會自動回到 Insert command

螢幕顯示：Insert command:

- G) Show Grade
- R) Show Rank
- A) Show Average
- W) Update Weight
- E) Exit

>

使用找輸入： R

螢幕顯示：許文馨's rank: 17

使用者輸入： W

螢幕顯示： Old Weights:

lab1: 10.0%

lab2: 10.0%

lab3: 10.0%

mid-term: 30.0%

final exam: 40.0%

New Weights:

在此使用者連續輸入 5 個數字來決定分別從 lab1 到 final exam 的權重

使用者輸入： 20 20 20 20 20

螢幕顯示：

lab1: 20

lab2: 20

lab3: 20

mid-term: 20

final exam: 20

Check new Weights:

lab1: 20%

lab2: 20%

lab3: 20%

mid-term: 20%

final exam: 20%

輸入完後，系統將顯示新權重，並詢問是否正確並儲存(按 y 及儲存，n 及捨棄並回歸原本權重)

螢幕顯示：Is it correct?(y/n)

使用者輸入： n (捨棄，並回到 Insert command)

使用者輸入： A

螢幕顯示：Average grade:89

使用者輸入： E (離開選單)

螢幕顯示：Enter ID or insert 'Q' to exit:

使用者輸入：Q (離開系統)

螢幕顯示：Quit



# Architectural Design

## class Main

main()

## class UI(User interface)

與使用者互動之函式集合。

UI()

promptCommand()

promptID()

showFinishMsg()

showWelcomeMsg(Grade)

## class GradeSystem

有關整體成績系統相關，如讀取成績、調整權重…等。

GradeSystem()

find(ID)

showGrade(Grade)

showAverage(Grade)

showRank(Grade)

updateWeights(newWeights)

promptUpdateWeights()

updateRank()

## class Grade

個別學生之成績、加權總成績、排名等。

Grade(line)

getGrade(index)

updateTotalGrade(weights)

# Detailed Design

## Main.java

#####

本 Grade system 讓使用者(學生)取得其總成績(total grade)及排名(rank)。

Total grade：基於配分(weights)來算，而配分可以調整。

Rank：表示總成績在全班學生的分數排名

Input file: 全班學生的分數 例如

962001044 凌宗廷 87 86 98 88 87

962001051 李威廷 81 32 50 90 93

注意 data field names 如下:

ID name lab1 lab2 lab3 midTerm finalExam

#####

**public class** Main:

**public static void** main(String[] args):

創建UI及GradeSystem物件。

**while (true):**

使用者輸入ID或輸入Q離開

找出該學生，並顯示歡迎訊息

**while (true):**

使用者輸入命令

若不存在該命令，則讓使用者再輸入一次

根據命令，執行

G -> Show grade

R -> Show rank

A -> Show Average

W -> Update weights

E -> Exit

# UI.java

```
/*
 * class UI(user interface)
 *
 * UI( )
 * promptCommand( )
 * promptID( )
 * showFinishMsg( )
 * showWelcomeMsg(Grade)
 */
public class UI

    /* method promptCommand-----
    * @return 使用者輸入 command
    * @throws NoSuchCommandExceptions – 使用者輸出無法處理之
Command。
    * Time estimate : 0 (1)
    -----*/
    public Cmd promptCommand() throws
NoSuchCommandExceptions
        1. 印出 RPOMPT_COMMAND_MESSAGE
        2. 接收並轉成 Cmd，若發生錯誤則丟出 NoSuchCommandExceptions

    /* method promptID -----
    * @return 使用者輸入之 ID
    * Time estimate : 0 (1)
    -----*/
    public String promptID( )
        1. 印出 PROMPT_ID_MESSAGE
        2. 接收使用者輸入
```

```
/* method showFinishMsg -----  
* Time estimate : 0 (1)  
-----*/
```

```
public void showFinishMsg( )
```

印出結束訊息

```
/* method showWelcomeMsg -----  
* @param student 欲歡迎之學生  
* Time estimate : 0 (1)  
-----*/
```

```
public void showWelcomeMsg(Grade student)
```

印出歡迎訊息

## GradeSystem.java

```
/*  
* class GradeSystem  
*  
* GradeSystem( )  
* find(ID)  
* showGrade(Grade)  
* showAverage(Grade)  
* showRank(Grade)  
* updateWeights(newWeights)  
* promptUpdateWeights( )  
* updateRank( )  
*/
```

```
public class GradeSystem {
```

```
/* GradeSystem constructor -----
* @throws IOException – src/gradeinput.txt 檔案不存在或
毀損
```

```
* Time estimate :  $O(n \times \log(n))$ 
```

```
-----*/
```

```
public GradeSystem() throws IOException {
```

```
    1. 開啟檔案，每一行創建一個 Grade object 存入 students 列表中
    2. 以 updateWeights() method 設定 INITIAL_WEIGHTS，會順便設定
total grade 及 排名
```

```
/* method find -----
```

```
* @param ID 尋找的ID
```

```
* @return 對應的 Grade object
```

```
* Time estimate :  $O(n)$ 
```

```
-----*/
```

```
public Grade find(String ID) {
```

```
    1. Iterate 內部 students，若 ID 相同則回傳。
    2. 若迴圈結束，還未回傳，代表找不到該 ID 對應的 student，回傳
null。
```

```
/* method showGrade -----
```

```
* @param student 要印出成績之學生 Grade object
```

```
* @return void (印出其成績資料)
```

```
* Time estimate :  $O(1)$ 
```

```
-----*/
```

```
public void showGrade(Grade student) {
```

```
    印出 student 的名字、5個成績、最後是加權後總成績
```

```
/* method showAverage -----
```

```
* @param student 目標學生
```

```
* @return void (印出平均成績)
```

```
* Time estimate :  $O(1)$ 
```

```
-----*/
```

```
public void showAverage(Grade student) {
```

```
    計算平均成績，然後印出
```

```

/* method showRank -----
* @param student 目標學生
* Time estimate : 0 (1)
-----*/
public void showRank(Grade student) {
    印出已計算好之排名

/* method updateWeights -----
* @param newWeights 欲更新之新加權比重
* Time estimate : 0 (n × log(n))
-----*/
public void updateWeights(double[] newWeights) {
    1. 更新內部 weights
    2. 呼叫 updateTotalGrade，更新所有學生之 total grade
    3. 以新的 total grade 排序所有學生（由大到小）
    4. 更新所有學生之排名

/* method promptUpdateWeights -----
* Time estimate : 0 (n × log(n)) （同 updateWeights）
-----*/
public void promptUpdateWeights() {
    1. 印出原加權比重
    2. 使用者輸入新加權比重
    3. 印出接收之新加權比重，使用者確認後，執行 updateWeights

/* method updateRank -----
* Time estimate : 0 (n)
* Example: GradeSystem物件.updateRank() ;
-----*/
public void updateRank() {
    (預設 students 已依 total grade 由大至小之順序排列好。)
    從排名1開始，遞增地分配排名，若與前一學生相同分數，則相同排
    名。

```

# Grade.java

```
/*
 * Grade(line)
 * getGrade(index)
 * updateTotalGrade(weights)
 */
public class Grade {

    /* Grade constructor -----
     * @param gradeLine 格式為"ID name lab1 lab2 lab3 mid-
term finalExam"之字串
     * Time estimate : 0 (1)
     -----*/
    public Grade(String gradeLine) {
        1. 以空格區分出不同資料
        2. 依序設定到內部欄位

    /* method getGrade -----
     * @param index 指定是哪個成績
     * @return 其字串形式
     * Time estimate : 0 (1)
     -----*/
    public String getGrade(int index) {
        1. 檢查是否低於 60 分
        2. 是，則加註 *；反之，則轉為 String 後直接回傳。

    /* method updateTotalGrade -----
     * @param weights 加權權重
     * Time estimate : 0 (1)
     -----*/
    public void updateTotalGrade(double[] weights) {
        1. 算出該加權權重下之 total grade
        2. 小數點後，四捨五入。
```