2022

# 程式語言與編譯器_HW2

**Programming Languages Track:**

**Practice on programming the same exercises in 5 different programming languages: Java, Python, R, ML, and Prolog**


Team’s member:

- 410821204 / 杜昉紜 / 資工三 --- R/Java/Prolog
- 410821203 / 朱婉云 / 資工三 --- Python/ML

# 目錄

# 1 | 問題描述

**Programming Languages Track:**

Practice on programming the same exercises in 5 different programming languages: Java, Python, R, ML, and Prolog. You may integrate the two programming exercises into one program for each programming language.

- **Programming Exercises**: The attached file HW2data.csv has the scores of a class: HW1, HW2, HW3, Midterm, and Final.

  A) The overall score of a student is calculated by the following formula:

**Score** = **HW1** * 0.1 + **HW2** * 0.1 + **HW3** * 0.1 + **Midterm** * 0.3 + **Final** * 0.4

Write a program in the above 5 different programming languages to calculate the overall score of each student.

B) The grade of a student is translated according to the following table:

| Score | Grade |
|---|---|
| 0-49 | E |
| 50-59 | D |
| 60-62 | C- |
| 63-66 | C |
| 67-69 | C+ |
| 70-72 | B- |
| 73-76 | B |
| 77-79 | B+ |
| 80-84 | A- |
| 85-89 | A |
| 90-100 | A+ |

Write a program in the above 5 different programming languages to translate the overall score of each student into a grade.

# 2 | 程式重點

## 2.1 | R

### 2.1.1 | 透過 read.table 讀入 csv 資料

```
read.table(file=mydata, header=TRUE, sep=",")
```

read.table 可以讀取大多數的 ASCII 資料，其中 file/ header/ sep 分別指的是以下意思：

| | |
|---|---|
| file | 檔案路徑相對於目前工作目錄 |
| header | 資料是否有包含欄位名稱 |
| sep | 資料的分隔符號 |

### 2.1.2 | 抓取 csv 的欄位資料

```
data <- read.table(file=mydata, header=TRUE, sep=",")
data$HW1        //$ 可以抓取其中一個欄位的所有資料
```

### 2.1.3 | 將資料建成 data.frame

```
data.frame(data$ID, score)
```

```
> total_info <- data.frame(data$ID, score)
> total_info
      data.ID score
1   410021001 73.55
2   410021002 64.10
3   410021003 76.95
4   410021004 55.40
5   410021005 63.00
6   410021006 70.00
7   410021007 67.15
8   410021008 57.70
9   410021009 62.55
10  410021010 80.35
```

資料框是非常常見的二維資料格式，由一系列的欄位（Column）和列（Row）所組成，常見的 Excel 試算表也是類似的資料表現形式，可使用 data.frame()來創建新的資料框

## 2.2| JAVA

### 2.2.1| BufferedReade 讀取檔案內容

BufferedReader 可以用來讀取鍵盤輸入和檔案內容

使用 BufferedReader 物件的 readLine()方法必須處理 IOException 例外（exception）

```
BufferedReader br = new BufferedReader(new FileReader(path));
            try
            {
                  while((line = br.readLine()) != null)
                  {
                        String[] values = line.split(",");
                        if(i != 0)
                        {
                        //System.out.println(values[0]);
                       // print each HW1 score
                        }
                        i++; // get the length
                  }
      //System.out.println("total_length is : " + (i-1));
            }
            catch (FileNotFoundException e)
            {
                  e.printStackTrace();
            }
            catch (IOException e)
            {
                  e.printStackTrace();
            }
```

### 2.2.2| split() 方法根據匹配給定的正則表達式來拆分字符串

.split( “,” )用”,” 分隔 str 字符串

```
String[] values = line.split(",");
```

## 2.3| PROLOG

### 2.3.1| And expression

```
and(Range_X, Range_Y) :- /* && */
      Range_X , Range_Y
.
```

```
and(Score >= 0, Score < 50)
```

### 2.3.2| 換行

nl 為換行

```
total_score(11) :-
      nl
.
```

### 2.3.3| 控制 (if - else)

if (score: 0~49) -> print (E)

```
and(Score >= 0, Score < 49.5) -> write('E'), nl;
```

## 2.4| PYTHON

### 2.4.1| 透過 DataFrame 讀入 csv 資料

```
df = pd.read_csv("HW2data.csv")
```

### 2.4.2| 在表格增加新的項目

```
df["Score"] = Score
df["Grade"] = Grade
```

### 2.4.3| 直接呈現整個表格

```
df
```

## 2.5| ML

### 2.5.1| 讀取輸入的資料

```
let student no id_ name hw1 hw2 hw3 mid fin =
```

### 2.5.2| 計算成績

因為成績是浮點數，所以乘法要寫成”*.”

```
let score = hw1 *. 0.1 +. hw2 *. 0.1 +. hw3 *. 0.1 +. mid *. 0.3 +.
fin *. 0.4 in
```

### 2.5.3| 輸入資料

因為成績是浮點數，所有成績都要加上小數點

```
student 1 410021001 "Alan" 90. 84.5 117. 60. 66.;;
```

### 2.5.4| 輸出資料

```
Printf.printf "%s / %f / %s\n" name score grade;;
```

# 3 | 程式碼

## 3.1 | R

```
mydata <- "C:/Users/user/Downloads/HW2data.csv"
data <- read.table(file=mydata, header=TRUE, sep=",")
score <- data$HW1*0.1 + data$HW2*0.1 + data$HW3*0.1 +
data$Midterm*0.3 + data$Final*0.4
total_info <- data.frame(data$ID, score)
total_info
all <- nrow(total_info) #計算資料行數有多少
for(i in 1:all){
print(total_info[i, "data.ID"])
if(total_info[i, "score"] >= 89.5){
print("A+")
}else if(total_info[i, "score"] >= 84.5){
print("A")
}else if(total_info[i, "score"] >= 79.5){
print("A-")
}else if(total_info[i, "score"] >= 76.5){
print("B+")
}else if(total_info[i, "score"] >= 72.5){
print("B")
}else if(total_info[i, "score"] >= 69.5){
print("B-")
}else if(total_info[i, "score"] >= 66.5){
print("C+")
}else if(total_info[i, "score"] >= 62.5){
print("C")
}else if(total_info[i, "score"] >= 59.5){
print("C-")
}else if(total_info[i, "score"] >= 49.5){
print("D")
}else {
print("E")
}
}
```

## 3.2 | JAVA

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class csvreader
{
	/*
	FileReader 用於讀取字符串
	BufferedReader 從字符輸入流中(FileReader)讀取文本，緩衝字符以提供高效讀取
	使用 BufferedReader 物件的 readLine() 方法必須處理 IOException 例外 (exception)
	*/
      public static void main(String[] args) throws IOException
      {
            String path = "C:\\HW2data.csv";
            String line;
            int i = 0; // not going to print first line also record the length.
            // print the csv data to check
            BufferedReader br = new BufferedReader(new FileReader(path));
            try
            {
                  while((line = br.readLine()) != null)
                  {
                        String[] values = line.split(",");
                        if(i != 0)
                        {
                              //System.out.println(values[0]);
                              // print each HW1 score
                        }
                        i++; // get the length
                  }
                  //System.out.println("total_length is : " + (i-1));
            }
            catch (IOException e)
            {
                  e.printStackTrace();
            }
```

```
            // while knowing the length, create a array to store
what we need
            int length = i;
            i = 0; // zero the i
            double[] Score = new double[length];
            int num = 0;
            BufferedReader br1 = new BufferedReader(new
FileReader(path));
            try
            {
                  while((line = br1.readLine()) != null)
                  {
                        String[] values = line.split(",");
                        if(i != 0)
                        {
                              System.out.println(values[1]);
                              Score[num] =
Double.parseDouble(values[3])*0.1 +
Double.parseDouble(values[4])*0.1 +
Double.parseDouble(values[5])*0.1 +
Double.parseDouble(values[6])*0.3 +
Double.parseDouble(values[7])*0.4;
                              System.out.println(Score[num]);
                              // next to caculate the final score
                              if(Score[num] >= 89.5)
                              {
                                    System.out.println("A+");
                              }
                              else if(Score[num] >= 84.5)
                              {
                                    System.out.println("A");
                              }
                              else if(Score[num] >= 79.5)
                              {
                                    System.out.println("A-");
                              }
                              else if(Score[num] >= 76.5)
                              {
                                    System.out.println("B+");
                              }
                              else if(Score[num] >= 72.5)
                              {
                                    System.out.println("B");
                              }
```

```
                                else if(Score[num] >= 69.5)
                                {
                                        System.out.println("B-");
                                }
                                else if(Score[num] >= 66.5)
                                {
                                        System.out.println("C+");
                                }
                                else if(Score[num] >= 62.5)
                                {
                                        System.out.println("C");
                                }
                                else if(Score[num] >= 59.5)
                                {
                                        System.out.println("C-");
                                }
                                else if(Score[num] >= 49.5)
                                {
                                        System.out.println("D");
                                }
                                else
                                {
                                        System.out.println("E");
                                }
                        }
                        i++;
                        num++;
                }
        }
        catch (IOException e)
        {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }
    }
}
```

## 3.3 | PROLOG

```
/*Before testing, we need to set execute path what is our pl's path , so
we need to
 write down this code:
 working_directory(CWD,'C:/Users/user/Desktop').

Note that if once we have modified the code and need to execute it
we need to scan the file again.*/
total_score(11) :-
       nl
.

total_score(N) :- /* all total_score will be printed */
       nl,
       stu_info(N),
       Next is (N + 1),
       total_score(Next)
.

and(Range_X, Range_Y) :- /* && */
       Range_X , Range_Y
.

stu_info(No)  :- /* student_info format */
    student(No, Id, Name, HW1, HW2, HW3, Mid, Final),
    write(Id),    /* print student ID */
       nl,           /* '\n' */
       write(Name),
       nl,
    stu_score(HW1, HW2, HW3, Mid, Final)
.

stu_score(HW1, HW2, HW3, Mid, Final):- /* calculate every student's score
*/
    Score is (HW1 * 0.1) + (HW2 * 0.1) + (HW3 * 0.1) + (Mid * 0.3) +
(Final * 0.4),
    write(Score), /* print total_score */
       nl,
    grade(Score)
.

grade(Score):-
       and(Score >= 0, Score < 49.5) -> write('E'),
       nl;
    and(Score >= 49.5, Score < 59.5) -> write('D'),
       nl;
    and(Score >= 59.5, Score < 62.5) -> write('C-'),
```

```
      nl;
    and(Score >= 62.5, Score < 66.5) -> write('C'),
      nl;
    and(Score >= 66.5, Score < 69.5) -> write('C+'),
      nl;
    and(Score >= 69.5, Score < 72.5) -> write('B-'),
      nl;
    and(Score >= 72.5, Score < 76.5) -> write('B'),
      nl;
    and(Score >= 76.5, Score < 79.5) -> write('B+'),
      nl;
    and(Score >= 79.5, Score < 84.5) -> write('A-'),
      nl;
    and(Score >= 84.5, Score < 89.5) -> write('A'),
      nl;
    and(Score >= 89.5, Score =< 100) -> write('A+'),
      nl
.
/*key in student_info*/
student(1, 410021001, 'Alan', 90, 84.5, 117, 60, 66).
student(2, 410021002, 'Bob', 85, 49, 80, 57, 64).
student(3, 410021003, 'Carrie', 90, 110.5, 117, 68, 62).
student(4, 410021004, 'David', 117, 85, 0, 44, 55).
student(5, 410021005, 'Ethan', 85, 56, 50, 57, 67).
student(6, 410021006, 'Frank', 90, 65, 65, 72, 66).
student(7, 410021007, 'Gary', 117, 110.5, 65, 69, 43).
student(8, 410021008, 'Helen', 117, 65, 50, 43, 54).
student(9, 410021009, 'Igor', 63, 59.5, 50, 51, 75).
student(10, 410021010, 'Jeff', 117, 110.5, 117, 53, 75).
```

## 3.4 | PYTHON

```
import numpy as np
import pandas as pd
#讀入沒有屬性列的 csv 檔
df = pd.read_csv("HW2data.csv")

Score = 0.1*df["HW1"] + 0.1*df["HW2"] + 0.1*df["HW3"] + 0.3*df["Midterm"]
+ 0.4*df["Final"]
df["Score"] = Score

Grade = Score
for i in range(len(Grade)):
    if(Score[i] >= 89.5):
        Grade[i] = "A+"
    elif(Score[i] >= 84.5):
        Grade[i] = "A"
    elif(Score[i] >= 79.5):
        Grade[i] = "A-"
    elif(Score[i] >= 76.5):
        Grade[i] = "B+"
    elif(Score[i] >= 72.5):
        Grade[i] = "B"
    elif(Score[i] >= 69.5):
        Grade[i] = "B-"
    elif(Score[i] >= 66.5):
        Grade[i] = "C+"
    elif(Score[i] >= 62.5):
        Grade[i] = "C"
    elif(Score[i] >= 59.5):
        Grade[i] = "C-"
    elif(Score[i] >= 49.5):
        Grade[i] = "D"
    else:
        Grade[i] = "E"df["Grade"] = Grade
df
```

## 3.5 | ML

```
let student no id_ name hw1 hw2 hw3 mid fin =
    let score = hw1 *. 0.1 +. hw2 *. 0.1 +. hw3 *. 0.1 +. mid *. 0.3 +.
fin *. 0.4 in
    let grade =
        if score >= 89.5 then "A+"
        else if score >= 84.5 then "A"
        else if score >= 79.5 then "A-"
        else if score >= 76.5 then "B+"
        else if score >= 72.5 then "B"
        else if score >= 69.5 then "B-"
        else if score >= 66.5 then "C+"
        else if score >= 62.5 then "C"
        else if score >= 59.5 then "C-"
        else if score >= 49.5 then "D"
        else "E" in
    Printf.printf "%s / %f / %s\n" name score grade;;

student 1 410021001 "Alan" 90. 84.5 117. 60. 66.;;
student 2 410021002 "Bob" 85. 49. 80. 57. 64.;;
student 3 410021003 "Carrie" 90. 110.5 117. 68. 62.;;
student 4 410021004 "David" 117. 85. 0. 44. 55.;;
student 5 410021005 "Ethan" 85. 56. 50. 57. 67.;;
student 6 410021006 "Frank" 90. 65. 65. 72. 66.;;
student 7 410021007 "Gary" 117. 110.5 65. 69. 43.;;
student 8 410021008 "Helen" 117. 65. 50. 43. 54.;;
student 9 410021009 "Igor" 63. 59.5 50. 51. 75.;;
student 10 410021010 "Jeff" 117. 110.5 117. 53. 75.;;
```

# 4 | 運行結果

## 4.1 | 全部學生成績

| | |
|---|---|
| 410021001 | B |
| 410021002 | C |
| 410021003 | B+ |
| 410021004 | D |
| 410021005 | C |
| 410021006 | B- |
| 410021007 | C+ |
| 410021008 | D |
| 410021009 | C |
| 410021010 | A- |

# 4.2 | R

C:\Users\user\Desktop\HW2.R - R 編輯器

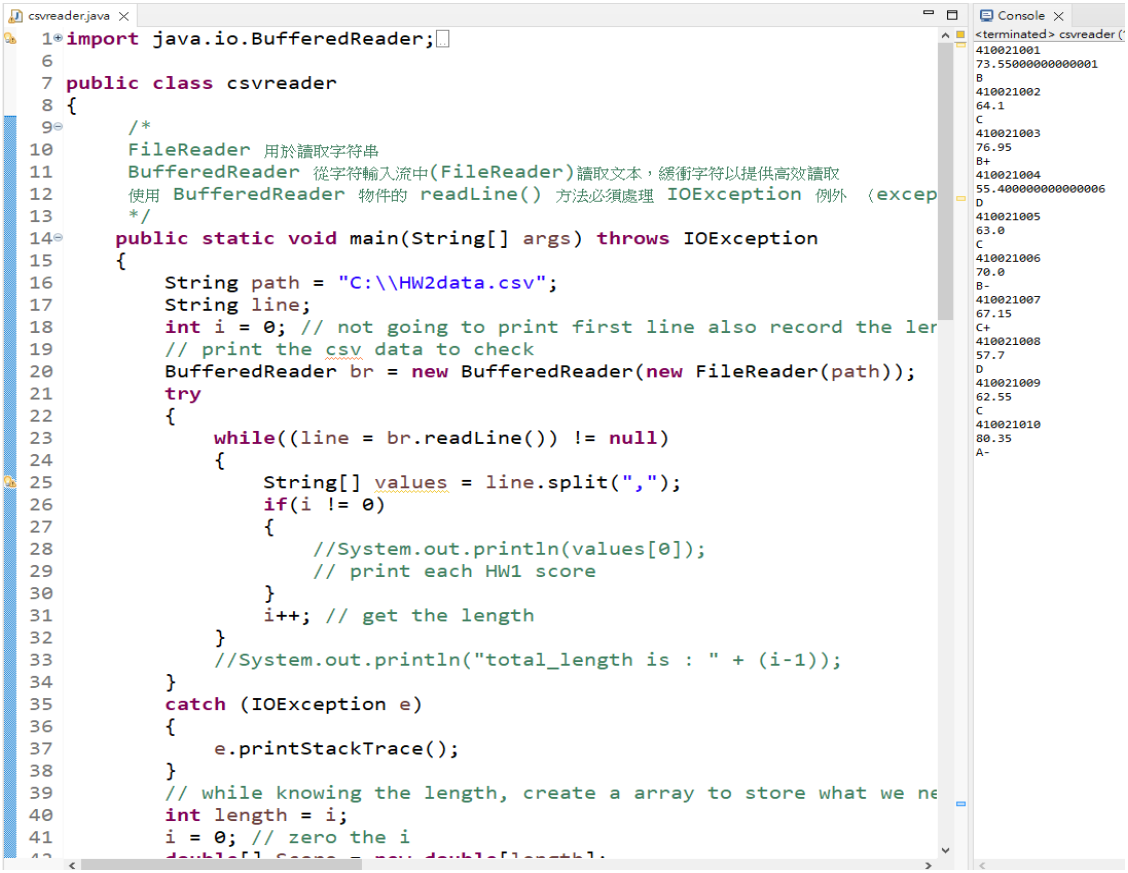
```
mydata <- "C:/Users/user/Downloads/HW2data.csv"
data <- read.table(file=mydata, header=TRUE, sep=",")
score <- data$HW1*0.1 + data$HW2*0.1 + data$HW3*0.1 + data$Midterm*0.3 + data$Final*0.4
total_info <- data.frame(data$ID, score)
total_info
all <- nrow(total_info)#計算資料行數有多少
for(i in 1:all){
print(total_info[i, "data.ID"])
print(total_info[i, "score"])
if(total_info[i, "score"] >= 89.5){
print("A+")
}else if(total_info[i, "score"] >= 84.5){
print("A")
}else if(total_info[i, "score"] >= 79.5){
print("A-")
}else if(total_info[i, "score"] >= 76.5){
print("B+")
}else if(total_info[i, "score"] >= 72.5){
print("B")
}else if(total_info[i, "score"] >= 69.5){
print("B-")
}else if(total_info[i, "score"] >= 66.5){
print("C+")
}else if(total_info[i, "score"] >= 62.5){
print("C")
}else if(total_info[i, "score"] >= 59.5){
print("C-")
}else if(total_info[i, "score"] >= 49.5){
print("D")
}else {
print("E")
}
}
```

R Console

```
+ }
[1] 410021001
[1] 73.55
[1] "B"
[1] 410021002
[1] 64.1
[1] "C"
[1] 410021003
[1] 76.95
[1] "B+"
[1] 410021004
[1] 55.4
[1] "D"
[1] 410021005
[1] 63
[1] "C"
[1] 410021006
[1] 70
[1] "B-"
[1] 410021007
[1] 67.15
[1] "C+"
[1] 410021008
[1] 57.7
[1] "D"
[1] 410021009
[1] 62.55
[1] "C"
[1] 410021010
[1] 80.35
[1] "A-"
>
```


```
[1] 410021001
[1] 73.55
[1] "B"
[1] 410021002
[1] 64.1
[1] "C"
[1] 410021003
[1] 76.95
[1] "B+"
[1] 410021004
[1] 55.4
[1] "D"
[1] 410021005
[1] 63
[1] "C"
[1] 410021006
[1] 70
[1] "B-"
[1] 410021007
[1] 67.15
[1] "C+"
[1] 410021008
[1] 57.7
[1] "D"
[1] 410021009
[1] 62.55
[1] "C"
[1] 410021010
[1] 80.35
[1] "A-"
>
```

# 4.3 | JAVA

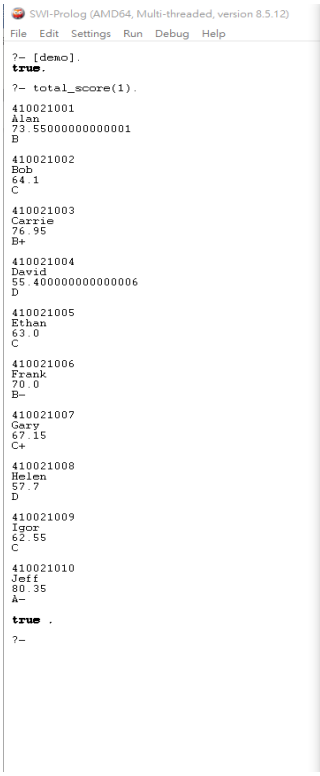


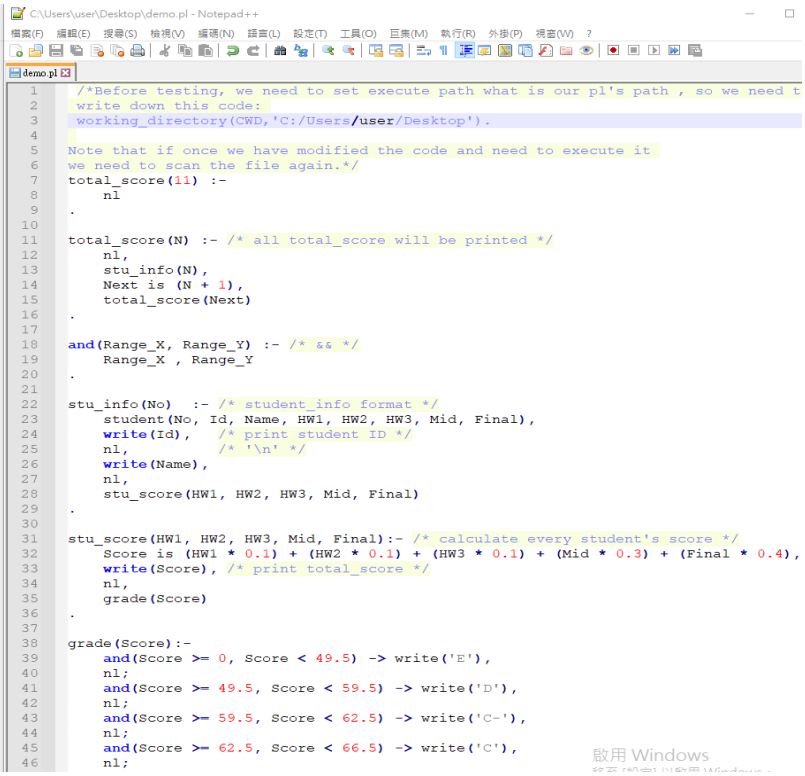
```java
import java.io.BufferedReader;

public class csvreader
{
    /*
    FileReader 用於讀取字符串
    BufferedReader 從字符輸入流中(FileReader)讀取文本，緩衝字符以提供高效讀取
    使用 BufferedReader 物件的 readLine() 方法必須處理 IOException 例外 (excep
    */
   public static void main(String[] args) throws IOException
   {
       String path = "C:\\HW2data.csv";
       String line;
       int i = 0; // not going to print first line also record the ler
       // print the csv data to check
       BufferedReader br = new BufferedReader(new FileReader(path));
       try
       {
           while((line = br.readLine()) != null)
           {
               String[] values = line.split(",");
               if(i != 0)
               {
                   //System.out.println(values[0]);
                   // print each HW1 score
               }
               i++; // get the length
           }
           //System.out.println("total_length is : " + (i-1));
       }
       catch (IOException e)
       {
           e.printStackTrace();
       }
       // while knowing the length, create a array to store what we ne
       int length = i;
       i = 0; // zero the i
```

```
Console
<terminated> csvreader (1)
410021001
73.55000000000001
B
410021002
64.1
C
410021003
76.95
B+
410021004
55.400000000000006
D
410021005
63.0
C
410021006
70.0
B-
410021007
67.15
C+
410021008
57.7
D
410021009
62.55
C
410021010
80.35
A-
```

# 4.4 | PROLOG




SWI-Prolog (AMD64, Multi-threaded, version 8.5.12)
File Edit Settings Run Debug Help
?- [demo].
true.
?- total_score(1).
410021001
Alan
73.55000000000001
B
410021002
Bob
64.1
C
410021003
Carrie
76.95
B+
410021004
David
55.400000000000006
D
410021005
Ethan
63.0
C
410021006
Frank
70.0
B-
410021007
Gary
67.15
C+
410021008
Helen
57.7
D
410021009
Igor
62.55
C
410021010
Jeff
80.35
A-
true .
?-

# 4.5 | PYTHON

In [4]:
```
import numpy as np
import pandas as pd
#讀入沒有屬性列的csv檔
df = pd.read_csv("HW2data.csv")

Score = 0.1*df["HW1"] + 0.1*df["HW2"] + 0.1*df["HW3"] + 0.3*df["Midterm"] + 0.4*df["Final"]
df["Score"] = Score

Grade = Score
for i in range(len(Grade)):
    if(Score[i] >= 89.5):
        Grade[i] = "A+"
    elif(Score[i] >= 84.5):
        Grade[i] = "A"
    elif(Score[i] >= 79.5):
        Grade[i] = "A-"
    elif(Score[i] >= 76.5):
        Grade[i] = "B+"
    elif(Score[i] >= 72.5):
        Grade[i] = "B"
    elif(Score[i] >= 69.5):
        Grade[i] = "B-"
    elif(Score[i] >= 66.5):
        Grade[i] = "C+"
    elif(Score[i] >= 62.5):
        Grade[i] = "C"
    elif(Score[i] >= 59.5):
        Grade[i] = "C-"
    elif(Score[i] >= 49.5):
        Grade[i] = "D"
    else:
        Grade[i] = "E"
df["Grade"] = Grade
df
```

Out[4]:

| | NO | ID | Name | HW1 | HW2 | HW3 | Midterm | Final | Score | Grade |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 410021001 | Alan | 90 | 84.5 | 117 | 60 | 66 | 73.55 | B |
| 1 | 2 | 410021002 | Bob | 85 | 49.0 | 80 | 57 | 64 | 64.10 | C |
| 2 | 3 | 410021003 | Carrie | 90 | 110.5 | 117 | 68 | 62 | 76.95 | B+ |
| 3 | 4 | 410021004 | David | 117 | 85.0 | 0 | 44 | 55 | 55.40 | D |
| 4 | 5 | 410021005 | Ethan | 85 | 56.0 | 50 | 57 | 67 | 63.00 | C |
| 5 | 6 | 410021006 | Frank | 90 | 65.0 | 65 | 72 | 66 | 70.00 | B- |
| 6 | 7 | 410021007 | Gary | 117 | 110.5 | 65 | 69 | 43 | 67.15 | C+ |
| 7 | 8 | 410021008 | Helen | 117 | 65.0 | 50 | 43 | 54 | 57.70 | D |
| 8 | 9 | 410021009 | Igor | 63 | 59.5 | 50 | 51 | 75 | 62.55 | C |
| 9 | 10 | 410021010 | Jeff | 117 | 110.5 | 117 | 53 | 75 | 80.35 | A- |

# 4.6 | ML

```
8  ********************************************************************************)
9  let student no id_ name hw1 hw2 hw3 mid fin =
10     let score = hw1 *. 0.1 +. hw2 *. 0.1 +. hw3 *. 0.1 +. mid *. 0.3 +. fin *. 0.4 in
11     let grade =
12         if score >= 89.5 then "A+"
13         else if score >= 84.5 then "A"
14         else if score >= 79.5 then "A-"
15         else if score >= 76.5 then "B+"
16         else if score >= 72.5 then "B"
17         else if score >= 69.5 then "B-"
18         else if score >= 66.5 then "C+"
19         else if score >= 62.5 then "C"
20         else if score >= 59.5 then "C-"
21         else if score >= 49.5 then "D"
22         else "E" in
23     Printf.printf "%s / %f / %s\n" name score grade;;
24
25 student 1 410021001 "Alan" 90. 84.5 117. 60. 66.;;
26 student 2 410021002 "Bob" 85. 49. 80. 57. 64.;;
27 student 3 410021003 "Carrie" 90. 110.5 117. 68. 62.;;
28 student 4 410021004 "David" 117. 85. 0. 44. 55.;;
29 student 5 410021005 "Ethan" 85. 56. 50. 57. 67.;;
30 student 6 410021006 "Frank" 90. 65. 65. 72. 66.;;
31 student 7 410021007 "Gary" 117. 110.5 65. 69. 43.;;
32 student 8 410021008 "Helen" 117. 65. 50. 43. 54.;;
33 student 9 410021009 "Igor" 63. 59.5 50. 51. 75.;;
34 student 10 410021010 "Jeff" 117. 110.5 117. 53. 75.;;
```

input

```
Alan / 73.550000 / B
Bob / 64.100000 / C
Carrie / 76.950000 / B+
David / 55.400000 / D
Ethan / 63.000000 / C
Frank / 70.000000 / B-
Gary / 67.150000 / C+
Helen / 57.700000 / D
Igor / 62.550000 / C
Jeff / 80.350000 / A-
```

# 5 | 討論

## 5.1 | 410821204 - 杜昉紜

經過了這 5 個程式語言的練習，發現 java / R / python 都可以很輕易的將 csv 讀入，然而因為對 prolog / ML 不太熟悉，所以在找相關範例 csv 處理時，發現類似數據比前面 3 個語言還要少，讓人不禁感嘆，難怪會有那麼多程式語言陸續出來，關於好用性與容易解讀性，非屬前面三個語言不可。

## 5.2 | 410821203 - 朱婉云

我除了 python 以外的程式語言都是第一次碰到，而且我寫的 ML 有些地方比較特別，像是用 ;; 當作結尾、用 Printf.printf 輸出資料等。裡面的內容只有要讀入 csv 檔的地方比較麻煩而已，其他數學運算只要有學過最基礎的 C 語言，基本上都很好上手。