

# Week4

## Pset1

這一題中，我們在嘗試 Passing Multi- Dimensional Array 時其實所打的那一串

```
void printArray(int a[][2], int len)
```

其實是在進行一個 `int [5][2] -> int (*)[2]` 的動作，

也就是說把 5 退化成指標，

那為什麼在本題中嘗試進行把 `int [2][5] -> int (*)[2]` 會出錯呢？

這其實就是在 Compile 過程中遇到型別錯誤，

換個方式來說就是 C++ 為了怕我們亂搞才設下限制跳出錯誤。

然後如果再換個方式來解釋，

那這其實與記憶體偏移量也有關，

但這就扯遠了不細說。

## Pset2

1. 因為 sum 為 Global Variable，所以並不會因為 `getAvg` 結束後就主動把 `getAvg` “recycle” 或是歸零，所以既然相同的數字相加兩次自然會變成兩倍。然後我們可以修改成如下來解決:

```
1  #include <iostream>
2  using namespace std;
3
4  const int cnt = 5;
5
6
7  void setScore(int score[], int num);
8  float getAvg(const int score[], int num);
9
10 int main() {
11     int score[cnt] = {0};
12     setScore(score, cnt);
13     cout << getAvg(score, cnt) << "\n";
14     cout << getAvg(score, cnt) << "\n";
15     return 0;
```

```
16 }
17 void setScore(int score[], int num) {
18     for (int i = 0; i < num; i++) cin >> score[i];
19 }
20 float getAvg(const int score[], int num) {
21     int sum = 0;
22     for (int i = 0; i < num; i++) sum += score[i];
23     return static_cast<float>(sum) / num;
24 }
```

2. Shown as below:

```
1  #include <iostream>
2  using namespace std;
3
4  const int cnt = 5;
5
6
7  void setScore(int score[], int num);
8  float getAvg(const int score[], int num);
9
10 int main() {
11     int score[cnt] = {0};
12     setScore(score, cnt);
13     cout << getAvg(score, cnt) << "\n";
14     cout << getAvg(score, cnt) << "\n";
15     return 0;
16 }
17 void setScore(int score[], int num) {
18     for (int i = 0; i < num; i++) cin >> score[i];
19 }
20 float getAvg(const int score[], int num) {
21     int sum = 0;
22     for (int i = 0; i < num; i++) sum += score[i];
23     return static_cast<float>(sum) / num;
24 }
```

3. 因為事實上 cnt 是用於當作 array length，而 declare a constant array length 能避免我們 go outside

an array，所以在課堂上說是值得推廣的好 Coding Style