



# บทที่ 7

## อาร์เรย์

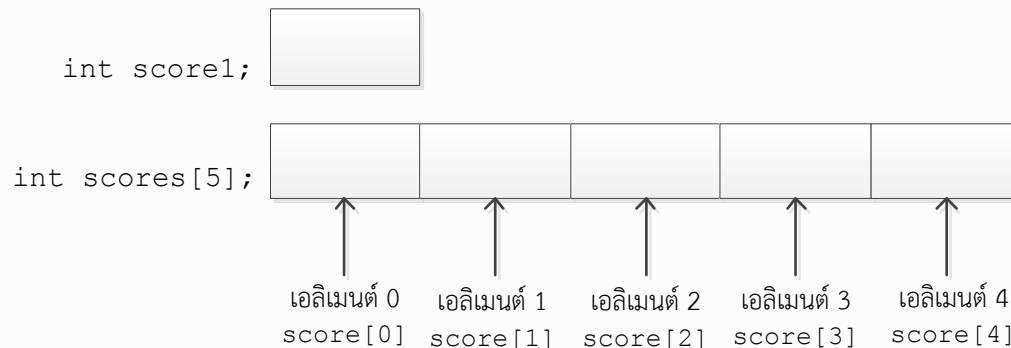
รายวิชา การโปรแกรมคอมพิวเตอร์

ดร.นิฏฐิตา เชิดชู

## 7.1 ตัวแปรอาร์เรย์



- ใช้เก็บข้อมูลเป็นกลุ่ม
- ข้อมูลจะต้องมี data type ชนิดเดียวกัน
- เช่น ต้องการเก็บคะแนนของ น.ศ. จำนวน 5 คน อาจเลือกใช้
  - score1, score2, score3, score4, score5
  - scores[5] -> จอง พ.ท. หน่วยความจำติดกันไว้ให้ 5 ห้อง





## 7.1.1 การประกาศตัวแปรอาร์เรย์

- Data type Name of array [size of array]
  - Data type – กำหนด data type ได้ชนิดเดียวเท่านั้น
  - Name of array – ตามหลักการกำหนดชื่อของตัวแปร
  - Size of array – จ.น. ข้อมูล หรือ เอลิเมนต์ (element) ที่ต้องการจอง พ.ท.
- การประกาศใช้งาน array

- บอก จ.น. ข้อมูล

การประกาศอาเรย์	จำนวนเอลิเมนต์	หน่วยความจำสำหรับชนิดข้อมูล 1 ตัว (ไบต์)	หน่วยความ ทั้งหมด (ไบต์)
int scores[5];	5	4	20
float average[100];	100	4	400
double income[12];	12	8	96
char name[10];	10	1	10

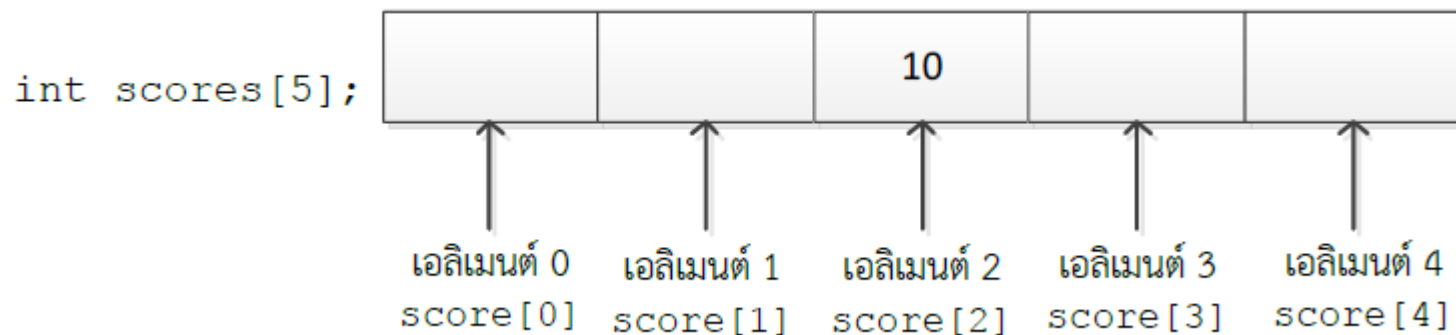
- ไม่บอก จ.น. ข้อมูล

```
int sales[] = {100, 204, 400, 500};
```



## 7.1.2 การเข้าถึงค่าในอาร์เรย์

- เข้าถึงในแต่ละเอลิเมนต์
- การอ้างอิงจะใช้ index ที่จะเริ่มเอลิเมนต์แรกคือ 0 เสมอ และเอลิเมนต์สุดท้าย คือ  $n-1$ ;  $n$  คือ ขนาดของอาร์เรย์
- การกำหนดค่าให้กับเอลิเมนต์ในอาร์เรย์
  - `scores[2] = 10;`



# การกำหนดค่าให้กับอาร์เรย์



นอกจากการกำหนดค่าเป็นตัวเลขให้กับเอลิเมนต์ในอาร์เรย์ เรายังสามารถกำหนดค่าในรูปแบบอื่นๆ ได้เช่นกัน

1) กำหนดค่าโดยใช้เอลิเมนต์ในอาร์เรย์อื่น เช่น

```
int scores[5];  
int numbers[3];  
scores[3] = 10;  
scores[4] = numbers[2];
```

2) นอกจากนั้นยังสามารถทำการกำหนดค่าของเอลิเมนต์ภายในอาร์เรย์ ไปพร้อมๆ กับการประกาศอาร์เรย์ก็ได้ เช่น

```
int amounts[4] = {3, 32, 17};
```

3) เลือกกำหนดค่าเอลิเมนต์เพียงบางเอลิเมนต์ก็ได้ เช่น

```
int index[10] = {1, 2, 3, 4};
```

ในกรณีเลือกกำหนดค่าเพียงบางเอลิเมนต์ในอาร์เรย์ จะไม่สามารถเลือกกำหนดเอลิเมนต์แบบไม่ต่อเนื่องได้ เช่น ในกรณีตัวอย่างต้องการกำหนดค่าเพียงแค่เอลิเมนต์ที่ 0-3 ก็ทำได้ แต่ไม่สามารถกำหนดเป็นช่วงๆ ได้ เช่น `int index[10] = {1, 2, 3, 4, , , 5, 7, 8};` เป็นต้น

# การอ่านค่าจากอาร์เรย์



1) การเลือกเอลิเมนต์ใดๆ ในอาร์เรย์เพื่อแสดงผล เช่น

```
cout<< scores[2];
```

2) การใช้วงเล็บเพื่อช่วยให้การแสดงผลหลายๆ เอลิเมนต์ในอาร์เรย์มีความสะดวก

มากยิ่งขึ้น ดังแสดงในตัวอย่างที่ 7.1

## ตัวอย่างที่ 7.1

### \*\* ข้อควรระวัง

- array[n] ไม่มี -> ค่าผิดพลาด

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     const int dataSize = 10;
7     int scores[dataSize] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
8     //display all elements in the array
9     for(int count = 0; count < dataSize; count++)
10     {
11         cout<<scores[count]<<endl;
12     }
13
14     return 0;
15 }
```

## ผลลัพธ์ของการรันโปรแกรม

```
10
20
30
40
50
60
70
80
90
100
```



# ตัวอย่างการเข้าถึงอาร์เรย์ที่เอลิเมนต์ n

## ตัวอย่างที่ 7.2

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     const int dataSize = 10;
7     int scores[dataSize] = {10,20,30,40,50,60,70,80,90,100};
8     float sum = 0;
9     float avg = 0;
10    //display all elements in the array
11    for(int count = 0; count <= dataSize; count++)
12    {
13        cout<< scores[count]<<endl;
14    }
15    return 0;
16 }
```

ผลลัพธ์ของการรันโปรแกรม

```
10
20
30
40
50
60
70
80
90
100
0
```



### ตัวอย่างที่ 7.3

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     const int SIZE = 3;
7     int array1[SIZE] = {1, 1, 1};
8     int array2[SIZE] = {2, 2, 2};
9
10    cout<< "The values stored in array 1 are: ";
11    for (int count = 0; count < SIZE; count++)
12    {
13        cout<<array1[count]<<"\t";
14    }
15
16    //Trying to store 5 numbers into array 2
17    for (int count = 0; count < 5; count++)
18    {
19        array2[count] = 3;
20    }
21
22    cout<<endl<<"The values stored in array 1 are: ";
23    for (int count = 0; count < SIZE; count++)
24    {
25        cout<<array1[count]<<"\t";
26    }
27    return 0;
28 }
```

#### ผลลัพธ์ของการรันโปรแกรม

```
The values stored in array 1 are: 1      1      1
The values stored in array 1 are: 3      3      1
```



## 7.2 การใช้อาร์เรย์เป็นอาร์กิวเมนต์ในฟังก์ชัน



- การส่งค่าอาร์เรย์ให้กับฟังก์ชัน
  - ส่งค่าให้ทีละเอลิเมนต์
  - ส่งอาร์เรย์ทั้งอาร์เรย์รวดเดียว

ตัวอย่างที่ 7.4

```
1 #include <iostream>
2 using namespace std;
3
4 void showValue(int);
5
6 int main()
7 {
8     int students[] = {43, 60, 33, 32, 70};
9     showValue(students[2]);
10    return 0;
11 }
12
13 void showValue(int number)
14 {
15     cout<< number <<endl;
16 }
```

ผลลัพธ์ของการรันโปรแกรม

33

ตัวอย่างที่ 7.5

```
1 #include <iostream>
2 using namespace std;
3
4 void showValue(int[], int);
5
6 int main()
7 {
8     const int SIZE = 5;
9     int students[] = {43, 60, 33, 32, 70};
10    showValue(students, SIZE);
11    return 0;
12 }
13
14 void showValue(int numbers[], int size)
15 {
16     for(int count = 0; count < size; count++)
17     {
18         cout<< numbers[count] <<endl;
19     }
20 }
```

ผลลัพธ์ของการรันโปรแกรม

43  
60  
33  
70



## 7.3 อาร์เรย์ของอักขระหรือสายอักขระ

- การประกาศตัวแปรชนิดสายอักขระ
  - ใช้เก็บชื่อที่มีตัวอักษรมากกว่า 1 ตัว
  - นิยมเก็บในรูปอาร์เรย์
  - ต้องเผื่อที่ไว้เก็บอักขระ null ('\0') ไว้ตำแหน่งท้ายสุด
  - เช่น อยากเก็บชื่อขนาด 10 อักขระในตัวแปร name
  - `char name[11] = Nitthita;` **สังเกต:** ต้องประกาศอาร์เรย์ขนาด 11 !!!

'N'	'I'	'T'	'T'	'H'	'I'	'T'	'A'	'\0'		
-----	-----	-----	-----	-----	-----	-----	-----	------	--	--



- การอ่านค่าสายอักขระจากแป้นพิมพ์
  - cin -> ใช้กับสายอักขระที่ไม่มีช่องว่าง (space)
  - cin.getline() -> ใช้กับสายอักขระที่มีช่องว่าง (#include <string>)

#### ตัวอย่างที่ 7.7

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main()
6 {
7     const int NAME_LEN = 21;
8     char name[NAME_LEN];
9     cout<< "Please enter your name: ";
10    cin>>name;
11    cout<< "Hello "<< name<< ", welcome to our C++ class.";
12    return 0;
13 }
```

ผลลัพธ์ของการรันโปรแกรม (ตัวหนาคือสิ่งที่ผู้ใช้งานป้อนให้กับโปรแกรม)

```
Please enter your name: Nitthita Chirdchoo
Hello Nitthita, welcome to our C++ class.
```



### ตัวอย่างที่ 7.8

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main()
6 {
7     const int NAME_LEN = 21;
8     char name[NAME_LEN];
9     cout<< "Please enter your name: ";
10    cin.getline(name, NAME_LEN);
11    cout<< "Hello "<< name<< ", welcome to our C++ class.";
12    return 0;
13 }
```

ผลลัพธ์ของการรันโปรแกรม (ตัวหนาคือสิ่งที่ผู้ใช้งานป้อนให้กับโปรแกรม)

```
Please enter your name: Nitthita Chirdchoo
Hello Nitthita Chirdchoo, welcome to our C++ class.
```

# ฟังก์ชันพื้นฐานเกี่ยวกับสายอักขระ



ฟังก์ชัน	วัตถุประสงค์การใช้งาน	รูปแบบการใช้งาน
strcpy()	คัดลอกสายอักขระในตัวแปรสายอักขระหนึ่งไปสู่อีกตัวแปรหนึ่ง	strcpy(str1, str2);
strcat()	ต่อสายอักขระ 2 สายอักขระเข้าด้วยกัน กลายเป็นสายอักขระเดียวโดยผลลัพธ์ของการต่อสายอักขระจะอยู่ที่สายอักขระที่เป็นอาร์กิวเมนต์แรก (ดูตัวอย่างการใช้งานประกอบ) ดังนั้นสายอักขระแรกจะต้องมีขนาดเพียงพอที่จะเก็บข้อมูลของสายอักขระหลังการต่อเสร็จสิ้นแล้ว	strcat(str1, str2);
strlen()	หาความยาว (จำนวนอักขระ) ของสายอักขระที่สนใจ	strlen(str1);
strcmp()	เปรียบเทียบว่าสายอักขระ 2 สายอักขระ โดยจะให้ผลลัพธ์ดังนี้  str1 == str2 ผลลัพธ์มีค่าเป็น 0 str1 > str2 ผลลัพธ์มีค่าเป็นบวก str1 < str2 ผลลัพธ์มีค่าเป็นลบ  โดยหลักการเปรียบเทียบจะอาศัยรูปแบบเดียวกับการ  จัดเรียงตัวอักษรในดิกชันนารี นั่นคือ  a < aa < aaa < ... < b < ba < bb <... < bz	strcmp(str1, str2);



## ตัวอย่างที่ 7.9

```
1 #include <iostream>
2 #include <string>
3 #include <cstring>
4 using namespace std;
5
6 int main()
7 {
8     const int SIZE = 20;
9     char name1[SIZE], name2[SIZE];
10    do
11    {
12        cout<< "Please provide two different names.";
13        cout<<endl;
14        cout<< "-----"<<endl;
15        cout<< "The name of the first person: ";
16        cin.getline(name1, SIZE);
17        cout<< endl<< "The name of the second person: ";
18        cin.getline(name2, SIZE);
19    }while(!strcmp(name1, name2));
20
21    cout<<endl<< "Thank you";
22
23    return 0;
24 }
```

### ผลลัพธ์ของการรันโปรแกรม (ตัวหนาคือสิ่งที่ผู้ใช้งานป้อนให้กับโปรแกรม)

```
Please provide two different names.
The name of the first person: John
The name of the second person: Jane
Thank you
```

```
Please provide two different names.
-----
The name of the first person: John
The name of the second person: JOHN
Thank you
```

Problem!!!



## ตัวอย่างที่ 7.10

```
1 #include <iostream>
2 #include <string>
3 #include <cstring>
4 using namespace std;
5
6 int main()
7 {
8     const int SIZE = 20;
9     char name1[SIZE], name2[SIZE];
10    do
11    {
12        cout<< "Please provide two different names.";
13        cout<<endl;
14        cout<< "-----"<<endl;
15        cout<< "The name of the first person: ";
16        cin.getline(name1, SIZE);
17        cout<< endl<< "The name of the second person: ";
18        cin.getline(name2, SIZE);
19
20        //change all characters to uppercase
21        for(int count = 0; count < SIZE; count++)
22        {
23            name1[count] = toupper(name1[count]);
24            name2[count] = toupper(name2[count]);
25        }
26        }while(!strcmp(name1, name2));
27
28        cout<<endl<< "Thank you";
29
30        return 0;
31    }
```

ผลลัพธ์ของการรันโปรแกรม (ตัวหนาคือสิ่งที่ผู้ใช้งานป้อนให้กับโปรแกรม)

Please provide two different names.

-----  
The name of the first person: John  
The name of the second person: JOHN  
Please provide two different names.

-----  
The name of the first person: John  
The name of the second person: Jane  
Thank you

