มหาวิทยาลัยราชภัฏนครปฐม

บทที่ 2
สตริงและโครงสร้างข้อมูลแบบเรคคอร์ด

ผู้บรรยาย : ผศ.ดร.ณัฐชามญฑ์ ศรีจำเริญรัตนา
สาขาวิชาคอมพิวเตอร์ธุรกิจ คณะวิทยาการจัดการ
มหาวิทยาลัยราชภัฏนครปฐม

สตริงและโครงสร้างข้อมูลแบบเรคคอร์ด

ปกติข้อความในภาษาซีจะถูกกำหนดในรูปแบบตารางอาร์เรย์ ทำให้สามารถนำเสนอตัวอักษรในรูปแบบข้อความ ค่าคงที่ได้ และยังสามารถพัฒนาให้สามารถนำตัวอักษรเหล่านั้นมารวบรวมให้เป็นรูปประโยคหรือข้อความที่มีความหมายได้

ดังนั้นจึงจำเป็นต้องเข้าใจรูปแบบการทำงานหรือฟังก์ชั่นต่าง ๆ ตามความเหมาะสมของโปรแกรมที่ต้องการพัฒนา

ผู้เรียนสามารถใช้ตัวแปรชนิดสตริงได้หลายรูปแบบ แต่ในโครงสร้างและขั้นตอนวิธีโดยใช้ภาษาซีนั้นมีฟังก์ชั่นอำนวยความสะดวกหลายตัว และผู้เรียนต้องเข้าใจการจัดการหน่วยความจำเพื่อประสิทธิภาพสูงสุดด้วย

2.1 ข้อมูลชนิดสตริง

- ข้อมูลชนิดสตริง (strings) เป็นอาร์เรย์หนึ่งมิติของอักขระที่นำมาต่อกันและอยู่ภายใต้เครื่องหมาย “ ” ที่ต้องมีเครื่องหมายบอกถึงการสิ้นสุดของข้อความด้วย “null character”
  หรืออาจจะต้องใส่เครื่องหมาย “\0” เพื่อแสดงถึงการสิ้นสุดชุดข้อมูลสตริงนั้น ๆ

- เนื่องจากเครื่องหมาย “\0” เป็นเครื่องหมายสิ้นสุดชุดข้อมูลเมื่อมีการสั่งพิมพ์ค่าจะไม่ทำการแสดงค่าออกมา

ตัวอย่าง การประกาศตัวแปรชนิดสตริง

```
char identifier[size]
```

โดยที่

identifier คือ ชื่อตัวแปรหรือออบเจ็กต์อาร์เรย์ char

size คือ ความยาวสตริงที่สามารถเก็บได้

2.1 ข้อมูลขนิดสตริง

ตัวอย่างที่ 2.1 การรับข้อมูลแบบอาเรย์

```
01  #include<iostream>
02  using namespace std;
03  int main()
04  {
05      char mystring[20];
06      mystring[0] = 'H';
07      mystring[1] = 'E';
08      mystring[2] = 'L';
09      mystring[3] = 'L';
10      mystring[4] = 'O';
11      mystring[5] = '\n';
12      mystring[6] = '\0';
13      cout<<mystring;
14      return 0;
15  }
```

❑ ผลการทำงาน

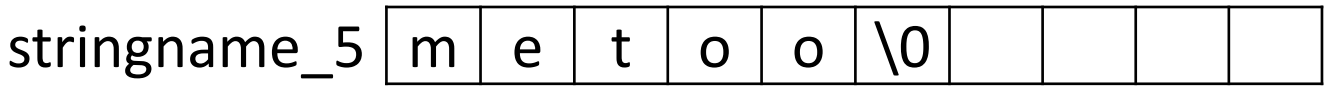
HELLO

2.2 การเก็บข้อมูลสตริงในหน่วยความจำ

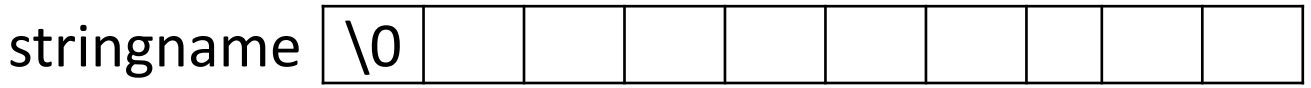
- ในกระบวนการแสดงข้อความ คอมไพล์เลอร์จะทำการเติมเครื่องหมาย “\0” ที่ตำแหน่งสุดท้ายของสตริงที่ประกาศเป็นอาร์เรย์
- ในการศึกษารูปแบบข้อมูลสตริงที่เก็บในตัวแปรชนิดอาร์เรย์พื้นที่หน่วยความจำที่จองสำหรับอาร์เรย์หลังตำแหน่งที่บอกจุดสิ้นสุดของข้อความจะไม่เป็นส่วนหนึ่งของข้อความนั้น ๆ

2.2 การเก็บข้อมูลสตริงในหน่วยความจำ(ต่อ)

ตัวอย่าง การเติมเครื่องหมาย “\0”



ตัวอย่าง การกำหนดข้อมูลสตริงที่กำหนดค่าให้กับตัวแปร stringname เป็น “null string”



- การตรวจสอบความยาวของตัวแปร stringname จะมีความยาวเป็น 0

2.3 การรับค่าสตริง

การทำงานของโปรแกรมเพื่อใช้ในการศึกษาขั้นตอนวิธี ประกอบด้วย 2 ส่วน คือ

- ส่วนของการแสดงผลการทำงานออกทางหน้าจอ
- ส่วนการรับข้อมูลจากผู้ใช้งานทางคีย์บอร์ดเพื่อใช้ในการประมวลผลของโปรแกรม

ในภาษาซีมีการกำหนดคำสั่งและฟังก์ชันมาตรฐานเพื่ออำนวยความสะดวกกับผู้พัฒนา

2.3 การรับค่าสตริง(ต่อ)

ตัวอย่างที่ 2.3 โปรแกรมรับข้อมูลจากทางคีย์บอร์ด

```
01  #include<iostream>
02  using namespace std;
03  int main ()
04  {
05      char str[10];
06      cout<<"Enter a string : " ;
07      cin>>str;
08      cout<<"Here is your string : "<<str;
09      return 0;
10  }
```

❑ **ผลการทำงาน**

```
Enter a string : hello
Here is your string : hello
```

2.3 การรับค่าสตริง(ต่อ)

ปัญหาเกิดจากฟังก์ชั่น cin

การทำงานในบางครั้งจะพบปัญหาในการรับข้อมูลทางคีย์บอร์ด เช่น “hello world” จะพบข้อความแสดงดังนี้

```
Enter a string : hello world
Here is your string : hello
```

เมื่อโปรแกรมแสดงผลจะแสดงข้อความ “hello” เท่านั้น ปัญหาการหยุดแสดงข้อความเมื่อพบเครื่องหมายเว้นวรรค แท็บ และคำสั่งขึ้นบรรทัดใหม่ จากฟังก์ชั่น cin สามารถแก้ปัญหานี้ได้โดยใช้ ฟังก์ชั่น gets()

2.3 การรับค่าสตริง(ต่อ)

ตัวอย่างที่ 2.4 ฟังก์ชั่น gets() เป็นการรับค่าทางคีย์บอร์ด เมื่อกด enter

```
01  #include<iostream>
02  using namespace std;
03  int main ()
04  {
05      char str[50];
06      cout<<"Enter a string : " ;
07      gets(str);
08      cout<<"Here is your string : "<<str;
09      return 0;
10  }
```

❑ ผลการทำงาน

```
Enter a string : hello world
Here is your string : hello world
```

2.4 การจัดการกับสตริง

ฟังก์ชั่นเกี่ยวกับการจัดการสตริงมีหลากหลาย ได้แก่

ฟังก์ชั่น strcpy(s1,s2); คัดลอกข้อความจากตัวแปร s2 มาเก็บที่ตัวแปร s1

ฟังก์ชั่น strcat(s1,s2); ทำการเชื่อมข้อความตัวแปร s1 และตัวแปร s2 เข้าหากัน

ฟังก์ชั่น strlen(s1); นับความยาวของสายอักขระของตัวแปร s1

2.4 การจัดการกับสตริง(ต่อ)

ตัวอย่างที่ 2.5 การใช้งานฟังก์ชั่นในการทำงานกับสตริง ดังต่อไปนี้

```
01  #include <string.h>
02  #include <iostream>
03  using namespace std;
04  int main ()
05  {
06      char str1[12] = "Hello";
07      char str2[12] = "World";
08      char str3[12];
09      int len ;
10      strcpy(str3, str1);
11      cout<<"\n strcpy( str3, str1) : "<< str3 ;
12      strcat( str1, str2);
13      cout<<"\n strcat( str1, str2): "<< str1 ;
14      len = strlen(str1);
15      cout<<"\n strlen(str1) : "<< len;
16      return 0;
17  }
```

❑ ผลการทำงานของโปรแกรม

```
strcpy(str3, str1) : Hello
strcat( str1, str2) : HelloWorld
strlen(str1) : 10
```

2.5 การใช้ฟังก์ชั่นเกี่ยวกับสตริง

ฟังก์ชั่นนับจำนวนตัวอักษรในประโยคที่ต้องการตรวจสอบ รวมถึงช่องว่าง (white space)

แต่ไม่รวมเครื่องหมาย “null terminator” ที่ใช้กำหนดจุดสิ้นสุดในประโยค

2.5 การใช้ฟังก์ชั่นเกี่ยวกับสตริง

ตัวอย่างที่ 2.6 การกำหนดค่าและการตรวจสอบความยาวของสตริง

```
01  #include <iostream>
02  #include <string.h>
03  using namespace std;
04  int main ()
05  {
06    char str[20] = "Display String\0";
07    char str_null[10]="\0";
08    cout<<"value of str variable is ="<< str <<"\n" ;
09    cout<<"Length of string is ="<<strlen(str) <<"\n" ;
10    cout<<"Checking size of string ="<<strlen(str_null) <<"\n" ;
11    return 0;
12  }
```

❑ ผลการทำงานของโปรแกรม

```
value of str variable is =Display string
Length of string is =14
Checking size of string =0
```

2.5 การใช้ฟังก์ชั่นเกี่ยวกับสตริง

1) **ฟังก์ชั่น strlen()** เป็นฟังก์ชั่นสำหรับการหาความยาวของสตริงในชุดข้อความตัวอักษร

- **ตัวอย่างที่ 2.7** การใช้ฟังก์ชั่น strlen()

```
01  #include <iostream>
02  #include <string.h>
03  using namespace std;
04  int main()
05  {
06     char str[100] ;
07     int i;
08     cout<<"Enter a string: " ;
09     cin>>str;
10     for(i=0; str[i]!='\0'; ++i);
11     cout<<"1. from for function ="<<i;
12     cout<<endl;
13     cout<<"2. from strlen function ="<<strlen(str);
14  }
```

❑ ผลการทำงานของโปรแกรม

```
Enter a string: student
1. from for function =7
2. from strlen function =7
```

2.5 การใช้ฟังก์ชั่นเกี่ยวกับสตริง

2) ฟังก์ชั่น strrev(s1) ใช้ในการกลับข้อความแสดงออกทางหน้าจอ

ตัวอย่างที่ 2.8 การใช้ฟังก์ชั่น strrev เพื่อกลับข้อความ

```
01  #include <iostream>
02  #include <string.h>
03  using namespace std;
04  int main()
05  {
06    char name[20] = "Hello";
07    cout<<"String before strrev( ) : "<<name;
08    cout<<"\n String after strrev( )  : "<<strrev(name);
09    return 0;
10  }
```

❑ ผลการทำงานของโปรแกรม

```
String before strrev( ) : Hello
String after strrev( )  : olleH
```

2.5 การใช้ฟังก์ชั่นเกี่ยวกับสตริง

3) ฟังก์ชั่น strlwr(s1) เปลี่ยนข้อความเป็นตัวพิมพ์เล็กทั้งหมด

ตัวอย่างที่ 2.9 การใช้ฟังก์ชั่น strlwr

```
01  #include <iostream>
02  #include <string.h>
03  using namespace std;
04  int main()
05  {
06    char name[20] = "HELLO";
07    cout<<"String before strlwr( ) : "<<name;
08    cout<<"\n String after strlwr( )  : "<< strlwr(name);
09    return 0;
10  }
```

❑ ผลการทำงานของโปรแกรม

```
String before strlwr( ) : HELLO
String after strlwr( )  : hello
```

2.5 การใช้ฟังก์ชั่นเกี่ยวกับสตริง

4) ฟังก์ชั่น strupr(s1) เปลี่ยนข้อความที่ต้องการให้เป็นตัวอักษรพิมพ์ใหญ่ทั้งหมด

ตัวอย่างที่ 2.10 การใช้ฟังก์ชั่น strupr

```
01  #include <iostream>
02  #include <string.h>
03  using namespace std;
04  int main()
05  {
06     char name[20] = "hello";
07     cout<<"String before strupr ( ) : "<<name;
08     cout<<"\n String after strupr ( )  : "<<strupr(name);
09     return 0;
10  }
```

❑ ผลการทำงานของโปรแกรม

```
String before strupr ( ) : hello
String after strupr ( )  : HELLO
```

2.5 การใช้ฟังก์ชั่นเกี่ยวกับสตริง

5) ฟังก์ชั่น strset( ) เป็นฟังก์ชั่นกำหนดการแสดงผลตามแบบของตัวอักษรที่กำหนด และ ตามจำนวนข้อความต้นแบบที่กำหนด

ตัวอย่างที่ 2.11 การใช้ฟังก์ชั่น strset

```
01  #include <iostream>
02  #include <string.h>
03  using namespace std;
04  int main()
05  {
06     char str[20] = "Test String";
07     cout<<"Original string is : "<<str;
08     strset(str,'#') ;
09     cout<<endl<<"After string set: "<<str ;
10     return 0;
11  }
```

❑ ผลการทำงานของโปรแกรม

```
Original string is : Test String
After string set: ###########
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

เรคคอร์ด (record) หรือระเบียนเป็นโครงสร้างข้อมูลที่กำหนดคุณสมบัติก่อนการใช้งาน โครงสร้างแบบเรคคอร์ดจะมีความคล้ายคลึงกับข้อมูลแบบอาร์เรย์

ระเบียนข้อมูลจะทำการจัดเก็บข้อมูล 2 ประเภทใหญ่ ๆ คือ

- ระเบียนเชิงตรรกะ ที่เก็บข้อมูลที่มีความสัมพันธ์กันเชิงตรรกะซึ่งกันและกันโดยไม่คำนึงว่าจะจัดเก็บไว้ที่ใดในอุปกรณ์บันทึกข้อมูล
- ระเบียนเชิงกายภาพ เป็นระเบียนในทัศนะของคอมพิวเตอร์ โดยพิจารณาจากการอ่านบันทึกข้อมูลลงสื่อบันทึกแต่ละครั้ง ซึ่งการอ่านหรือบันทึกแต่ละครั้งถือว่าเป็น 1 หน่วยของสื่อบันทึกข้อมูล

2.6 โครงสร้างข้อมูลเรคคอร์ด

การประกาศโครงสร้างข้อมูลแบบเรคคอร์ด

```
Record_Name(field_1, field_2, field_3,…,field_N)
```

โดย Record_Name : ชื่อของข้อมูลเรคคอร์ด

field_1, field_2, field_3,…,field_N : สมาชิกของเรคคอร์ด

2.6 โครงสร้างข้อมูลเรคคอร์ด

ตัวอย่าง เรคคอร์ดชื่อ customer ประกอบด้วยฟิลด์ ดังต่อไปนี้ Id, Name, Address, Tel ซึ่งกำหนดให้แต่ละฟิลด์ จัดเก็บข้อมูลที่แตกต่างชนิดกัน ดังนี้

ชื่อเรคคอร์ด	ชื่อฟิลด์	รายละเอียด	ชนิดข้อมูล	ขนาดข้อมูล	ค่าข้อมูล
customer	id	เก็บรหัสลูกค้า	Integer	8	Q9346978
	name	เก็บชื่อลูกค้า	string	20	YARA
	address	เก็บที่อยู่ลูกค้า	string	50	อ.เมือง จ.นครปฐม
	tel	เบอร์โทร	string	10	01-23456789

2.6 โครงสร้างข้อมูลเรคคอร์ด

ตัวอย่างที่ 2.12 รูปแบบการประกาศโครงสร้างข้อมูล

```
structure ชื่อโครงสร้างข้อมูล
{
   ชนิดข้อมูล   ชื่อฟิลด์ตัวแปร_1;
   ชนิดข้อมูล   ชื่อฟิลด์ตัวแปร_2;
   .....
   ชนิดข้อมูล   ชื่อฟิลด์ตัวแปร_n;
} ชื่อตัวแปรโครงสร้างแบบสตรักเจอร์;
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

2.6.1 ตำแหน่งการประกาศตัวแปรแบบเรคคอร์ด

การประกาศชื่อตัวแปรและชนิดของตัวแปรเพื่อเตรียมสำหรับการใช้งานตัวแปรแบบเรคคอร์ดมีการประกาศอยู่ 2 ตำแหน่งคือ

- 1. ประกาศในส่วนหลัง header_file
- 2. ประกาศภายในส่วน main()

```
#include<header_file>
        ประกาศตัวแปรแบบเรคคอร์ด
void main()
{
        ประกาศตัวแปรแบบเรคคอร์ด
        statement;
}
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

2.6.2 การแสดงข้อมูลแบบเรคคอร์ด

การเข้าถึงสมาชิกข้อมูลแบบเรคคอร์ดสามารถนำข้อมูลออกมาแสดงได้โดยใช้เครื่องหมายจุด (.) โดยใช้ชื่อตัวแปรโครงสร้างและเชื่อมด้วยเครื่องหมายจุดตามด้วยสมาชิกโครงสร้างที่ต้องการที่จะเข้าถึง

2.6 โครงสร้างข้อมูลเรคคอร์ด

ตัวอย่างที่ 2.13 การเข้าถึงข้อมูลของสมาชิกในโครงสร้างแบบเรคคอร์ดเพื่อนำข้อมูลมาแสดงผล

```
01  #include<iostream>
02  #include <string.h>
03  using namespace std;
04  struct Books
05  {
06      int id;
07      char name[20];
08      char address[50];
09      char tel[10];
10  }customer;
11
```

```
12  int main( )
13  {
14      customer.id=9999;
15      strcpy( customer.name, "NPRU");
16      strcpy( customer.address, "Logistic Bld.");
17      strcpy( customer.tel, "0123456789");
18      cout<< "Book 1 title : “;
19      cout<< customer.id <<endl;
20      cout<< "Book 1 author : " ;
21      cout<< customer.name <<endl;
22     cout<< "Book 1 subject : ";
23      cout<< customer.address <<endl;
       cout<< "Book 1 id : " << customer.tel;
     return 0;
  }
```

❑ ผลการทำงานของโปรแกรม

```
Book 1 title : 9999
Book 1 author : NPRU
Book 1 subject : Logistic Bld.
Book 1 id : 0123456789
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

2.6.3 การรับข้อมูลและการแสดงผลข้อมูลแบบเรคคอร์ด

- ปกติโครงสร้างข้อมูลแบบเรคคอร์ดจะไม่เก็บข้อมูลเพียงเรคคอร์ดเดียว
- สามารถเก็บข้อมูลได้ครั้งละหลาย ๆ เรคคอร์ด
- โดยใช้ฟังก์ชั่น for เพื่อทำการวนการรับข้อมูล และจะกำหนดจำนวนเรคคอร์ดที่ต้องการเก็บไว้ล่วงหน้าเพื่อทำการจัดเก็บลงฟิลด์ตัวแปรของข้อมูลแบบตัวแปรเรคคอร์ดที่ได้ทำการบันทึกไว้ภายในโปรแกรมนั่นเอง

2.6 โครงสร้างข้อมูลเรคคอร์ด

ตัวอย่างที่ 2.14 จงเขียนโปรแกรมให้สามารถรับข้อมูลเพื่อจัดเก็บลงฟิลด์ตัวแปรของข้อมูลแบบตัวแปรเรคคอร์ดและแสดงข้อมูลทั้งหมดที่ได้บันทึกไว้โดยใช้ฟังก์ชั่น for เพื่อวนการรับข้อมูล

```
01  #include <stdio.h>
02  #include <string.h>
03  #include<iostream>
04  using namespace std;
05  int main()
06  {
07   int num,i;
08  cout<<"Enter number of record :=";
09  cin>>num;
10  struct customer
11  {
12      int id;
13      char name[20];
14      char address[50];
15      char tel[10];
16  } record[num];
17
```

ตัวอย่างที่ 2.14 จงเขียนโปรแกรมให้สามารถรับข้อมูลเพื่อจัดเก็บลงฟิลด์ตัวแปรของข้อมูลแบบตัวแปรเรคคอร์ดและแสดงข้อมูลทั้งหมดที่ได้บันทึกไว้โดยใช้ฟังก์ชั่น for เพื่อวนการรับข้อมูล

```
18      for(i=0; i<num; i++)
19      {
20              cout<<endl<<"Records of Customer :"<<i+1<<endl;
21              cout<<"     Id is:";        cin>>record[i].id ;
22              cout<<"     Name is: ";   cin>>record[i].name;
23              cout<<"     address is:"; cin>>record[i].address;
24              cout<<"     tel is:";       cin>>record[i].tel;
25              cout<<"====================================";
26      }
27  cout<<endl<<"===========show data============";
28      for(i=0; i<num; i++)
29      {
30              cout<<endl<<"Records of Customer :"<<i+1<<endl;
31              cout<<endl<<" Id is:"<<record[i].id ;
32              cout<<endl<<" Name is: "<<record[i].name;
33              cout<<endl<<" address is:"<<record[i].address;
34              cout<<endl<<" tel is:"<<record[i].tel;
35              cout<<endl<<"==========================";
36      }
37        return 0;
38  }
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

ผลการทำงานของโปรแกรม

```
Enter number of record :=3
Records of Customer :1
    Id is:1
    Name is: name_1
    address is:address_1
    tel is:tel_1
========================
Records of Customer :2
    Id is:2
    Name is: name_2
    address is:address_2
    tel is:tel_2
========================
Records of Customer :3
    Id is:3
    Name is: name_3
    address is:address_3
    tel is:tel_3
```

```
==========show data=====
Records of Customer :1
    Id is:1
    Name is: name_1
    address is:address_1
    tel is:tel_1
======================
Records of Customer :2
    Id is:2
    Name is: name_2
    address is:address_2
    tel is:tel_2
=========================
Records of Customer :3
    Id is:3
    Name is: name_3
    address is:address_3
    tel is:tel_3
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

2.6.4 การใช้โปรแกรมย่อยในในตัวแปรแบบเรคคอร์ด

- เป็นโปรแกรมชนิดหนึ่งที่ถูกสร้างขึ้นมาเพื่อทำงานอย่างมีเป้าหมายที่ชัดเจน อีกทั้งโปรแกรมย่อยสามารถเรียกใช้งานได้หลายๆ ครั้ง และยังลดความซ้ำซ้อนของโปรแกรมหลักทำให้ง่ายต่อการพัฒนา
- เนื่องจากโปรแกรมย่อยจะสร้างมาเพื่องานใด ๆ ที่จำเพาะเจาะจงแล้ว การแก้ไขงานก็จะแก้ไขเป็นส่วน ๆ ไม่ต้องแก้ไขทั้งโปรแกรม
- ดังโปรแกรมตัวอย่างต่อไปนี้ จะแบ่งการทำงานของโปรแกรมออกเป็น 2 ส่วนย่อย ๆ คือ

ส่วนแสดงผลข้อมูล และส่วนนำเข้าข้อมูล เพื่อความสะดวกในการใช้งาน และการแบ่งการทำงานของแต่ละส่วนงานให้ชัดเจนยิ่งขึ้น

2.6 โครงสร้างข้อมูลเรคคอร์ด

ตัวอย่างที่ 2.15 การแบ่งการทำงานให้เป็นโปรแกรมย่อย

```
01 #include <stdio.h>
02 #include <string.h>
03 #include<iostream>
04 using namespace std;
05 void display_data(int num);
06 void input_data(int num);
07 struct customer
08 {
09     int id;
10     char name[20];
11     char address[50];
12     char tel[10];
13 } record[3];
14
```

```
15 int main()
16 {
17   input_data(3);
18   cout<<endl<<"=====show data=====";
19   display_data(3);
20   return 0;
21 }
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

ตัวอย่างที่ 2.15 การแบ่งการทำงานให้เป็นโปรแกรมย่อย (ต่อ)

```
22  void input_data(int num)
23  {
24      for(int i=0; i<num; i++)
25      {
26        cout<<endl<<"Records of Customer :"<<i+1<<endl;
27        cout<<"      Id is:"; cin>>record[i].id ;
28        cout<<"      Name is: "; cin>>record[i].name;
29        cout<<"      address is:"; cin>>record[i].address;
30        cout<<"      tel is:"; cin>>record[i].tel;
31      cout<<"=================";
32      }
33   }
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

ตัวอย่างที่ 2.15 การแบ่งการทำงานให้เป็นโปรแกรมย่อย (ต่อ)

```
34  void display_data(int num)
35  {
36     for(int i=0; i<num; i++)
37      {
38        cout<<endl<<"Records of Customer :"<<i+1<<endl;
39        cout<<endl<<"      Id is:"<<record[i].id ;
40        cout<<endl<<"      Name is: "<<record[i].name;
41        cout<<endl<<"      address is:"<<record[i].address;
42        cout<<endl<<"      tel is:"<<record[i].tel;
43        cout<<endl<<"==================";
44      }
45  }
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

2.6.5 การสร้างเมนูฟังก์ชั่น

ในการทำงานเพื่อประมวลผลการทำงาน จะทำงานโดยการรันและทำงานตามขั้นตอนต่าง ๆ ตามข้อกำหนดของโครงสร้างโปรแกรมที่สร้างขึ้น และจบการทำงาน

- แต่บางครั้งผู้ใช้ต้องการเรียกการทำงานบางส่วนเพื่อทำงานซ้ำ ๆ จึงต้องมีการออกแบบโปรแกรมให้ใช้งานได้สะดวกมากยิ่งขึ้นโดยออกแบบการทำงานของโปรแกรมให้ทำงานด้วยการสร้างเมนูฟังก์ชั่นเพื่อเรียกใช้งานได้หลาย ๆ ครั้ง
- ดังโปรแกรมตัวอย่างต่อไปนี้ เป็นการสร้างเมนูเพื่อช่วยในการทำงานและเพื่อให้โปรแกรมมีความสะดวกยิ่งขึ้น เราสามารถใช้ความสามารถของฟังก์ชั่น do....while รวมกับฟังก์ชั่น if...else หรือ switch...case ให้สามารถวนการรับงานจากผู้ใช้โดยการทำงานจะแบ่งเป็น 3 ฟังก์ชั่นย่อย คือ
  - input_data เพื่อรับข้อมูล
  - display_data เพื่อแสดงข้อมูลและ
  - exit เพื่อจบการทำงาน

2.6 โครงสร้างข้อมูลเรคคอร์ด

ตัวอย่างที่ 2.16 การสร้างเมนูเพื่อเพิ่มความสามารถในการเรียกใช้ข้อมูลแบบไม่จำกัดครั้งในการทำงาน

```
01  #include <stdio.h>
02  #include <string.h>
03  #include <stdlib.h>
04  #include <conio.h>
05  #include <iostream>
06  using namespace std;
07
08  void display_data();
09  void input_data();
10
```

```
11  struct customer
12  {
13          int id;
14          char name[20];
15          char address[50];
16          char tel[10];
17  } record[2];
18  int num,i;
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

ตัวอย่างที่ 2.16 การสร้างเมนูเพื่อเพิ่มความสามารถในการเรียกใช้ข้อมูลแบบไม่จำกัดครั้งในการทำงาน

```
19  int main()
20  {
21  char ans;
22  do{
23      cout<<"\n****MAIN MENU*******";
24      cout<<"\n----------------------";
25      cout<<"\n 1. INPUT DATA";
26      cout<<"\n 2. DISPLAY DATA";
27      cout<<"\n 3. EXIT";
28      cout<<"\n----------------------";
29      cout<<"\n select menu(1 2 3):";
30      ans=getche();
```

```
31      switch(ans)
32      {
33      case '1': { input_data(); break; }
34      case '2': { display_data(); break; }
35      case '3' : exit(0);
36      default:  cout<<"\n please enter choice again";
        }
37
39  }while (ans!=3);
39  system("pause");
40  return 0;
41  }
42
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

ตัวอย่างที่ 2.16 (ต่อ 2)

```
43  void input_data()
44  {
45     cout<<"\nEnter number of record :=";
46     cin>>num;
47     for(i=0; i<num; i++)
48     {
49        cout<<"\n Records of Customer :"<<i+1<<endl;
50        cout<<"     Id is:"; cin>>record[i].id ;
51        cout<<"     Name is: "; cin>>record[i].name;
52        cout<<"     address is:"; cin>>record[i].address;
53        cout<<"     tel is:"; cin>>record[i].tel;
54        cout<<"============================";
55     }
56  }
```

2.6 โครงสร้างข้อมูลเรคคอร์ด

ตัวอย่างที่ 2.16 (ต่อ 3)

```
57  void display_data()
58  {
59     for(i=0; i<num; i++)
60     {
61        cout<<endl<<"Records of Customer :"<<i+1<<endl;
62        cout<<endl<<"  Id is:"<<record[i].id ;
63        cout<<endl<<"  Name is: "<<record[i].name;
64        cout<<endl<<"  address is:"<<record[i].address;
65        cout<<endl<<"  tel is:"<<record[i].tel;
66
67        cout<<endl<<"=========================";
68     }
69  }
```

มหาวิทยาลัยราชภัฏนครปฐม