# หน่วยเรียนที่ 2

# การเขียนโปรแกรมระดับสูงด้วยภาษาซี

ภาษาซีจัดเป็นภาษาระดับกลางที่มีลักษณะเป็นภาษาที่มีโครงสร้าง (Structure) เป็นภาษาที่สูงกว่าภาษาแอสแซมบลี ทำให้ผู้เขียนโปรแกรมเข้าใจคำสั่งได้ง่ายขึ้น และสามารถเขียนโปรแกรมได้อย่างคล่องตัวโดยไม่มีข้อจำกัดในการวางตำแหน่งฟังก์ชันในโปรแกรม ภาษาซีจึงเป็นภาษาที่ง่ายต่อการเข้าใจและมีการนำไปใช้งานแพร่หลายทั่วโลก

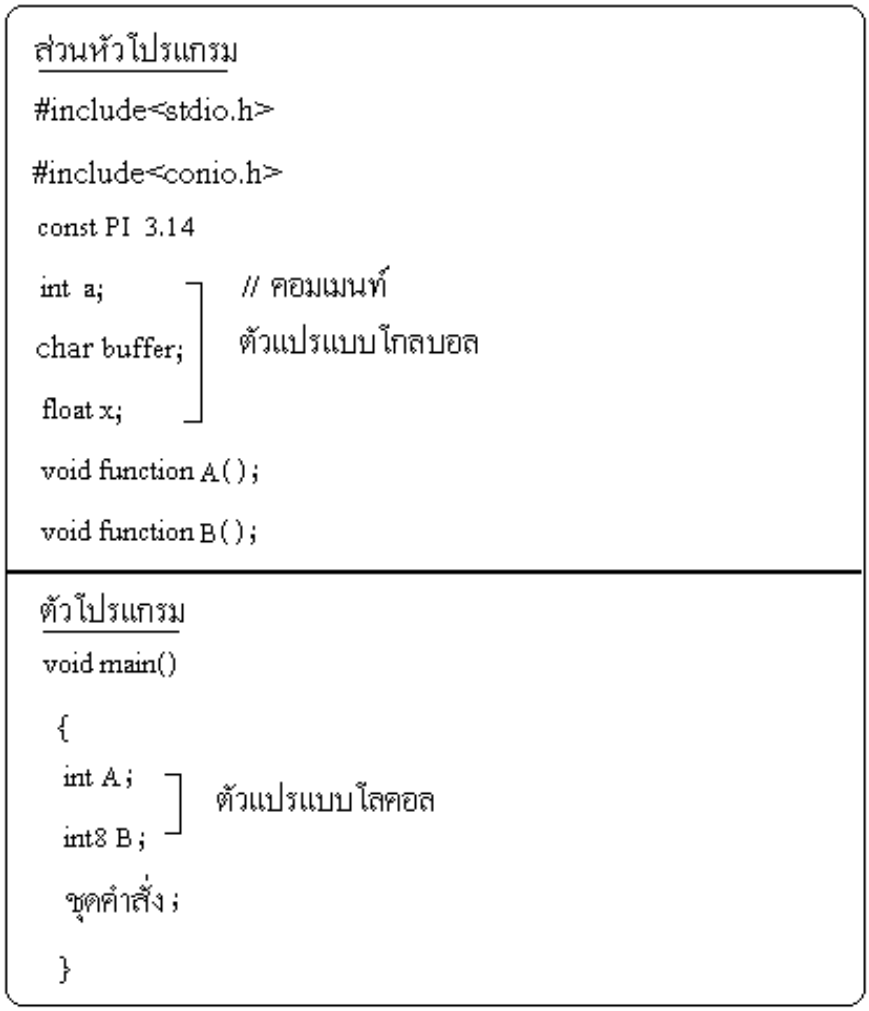
## 2.1 โครงสร้างโปรแกรมภาษาซี

การเขียนโปรแกรมภาษาซีซึ่งเป็นโปรแกรมที่มีโครงสร้างนั้น จะต้องทำความเข้าใจโครงสร้าง หลักการ กฎกติกาและรูปแบบการใช้คำสั่งของการเขียนโปรแกรมภาษาซีเสียก่อน จึงจะทำให้เขียนโปรแกรมภาษาซีได้อย่างถูกต้องและเหมาะสม โดยโครงสร้างของภาษาซีประกอบไปด้วย ส่วนหัวโปรแกรม การประกาศตัวแปร และประกอบด้วยฟังก์ชันต่างๆ คือทั้งฟังก์ชันหลักและฟังชันก์ย่อยต่างๆ เพื่อประกอบกันเป็นโครงสร้างของภาษาซี โดยมีลักษณะของโครงสร้างดังภาพที่ 2.1 และมีส่วนประกอบของโครงสร้างดังนี้

### 2.1.1 ส่วนหัวโปรแกรม

ส่วนหัวโปรแกรม (Header) จะเป็นส่วนที่อยู่ตอนต้นของโปรแกรม เพื่อสำหรับการประกาศการนำไฟล์ที่เกี่ยวข้องเข้ามาร่วมในโปรแกรม เป็นคำสั่งที่ขึ้นต้นด้วยเครื่องหมาย # (Double Cross) ซึ่งเป็นส่วนที่ต้องนำเข้ามาประมวลผลในลำดับแรก (Preprocessor) ก่อนที่จะไปคอมไพล์การทำงานของฟังก์ชันอื่นๆ โดยจะเริ่มด้วยคำสั่ง #include เพื่อนำไฟล์ที่เกี่ยวข้องกับการเขียนโปรแกรมมาใช้งานกับโปรแกรมที่กำลังเขียนอยู่ เช่นไฟล์ที่เกี่ยวกับฟังก์ชันมาตรฐานการรับข้อมูล การแสดงผล การคำนวณ ที่ผู้เขียนโปรแกรมสามารถเรียกใช้ได้ทันทีโดยไม่ต้องเขียนโปรแกรมเอง เพียงแต่ให้มีการอ้างถึง เช่น การแสดงข้อความทางจอภาพด้วยคำสั่ง prinf() ซึ่งก่อนใช้คำสั่งนี้จะต้องมีการอ้างถึงฟังก์ชันมาตรฐานที่เกี่ยวกับการรับข้อมูลและเอาท์พุต โดยใช้คำสั่ง

#include <stdio.h> เสียก่อนใช้คำสั่ง printf() ไม่เช่นนั้นจะทำให้เกิดความผิดพลาดขึ้นเมื่อทำการคอมไพล์




**ภาพที่ 2.1** ลักษณะโครงสร้างของภาษาซี

นอกจากนี้ในส่วนหัวโปรแกรมยังสามารถกำหนดค่าคงที่ หรือกำหนดค่า ของชื่อที่ได้นิยามขึ้นในโปรแกรม โดยใช้ใช้ไดเร็กทีฟ #define เพื่อระบุค่าของชื่อที่นิยามขึ้นมา เช่น

```
#define ON      1       //  กำหนดให้ ON มีค่าเป็น 1
#define OFF     0       //  กำหนดให้ OFF มีค่าเป็น0
#define PI      3.142   // กำหนดให้ PI  มีค่าเป็นหรือเท่ากับ3.142
#define True 1          // เป็นการกำหนดว่า หากนำคำว่า True ไปใช้ ณ
                        //ส่วนใดของโปรแกรม True นั้นมีค่าเท่ากับ 1
```

### 2.1.2 ค่าคงที่

ค่าคงที่ (Constants) คือค่าที่ไม่มีการเปลี่ยนแปลงค่าในขณะประมวลผลโปรแกรมของคอมพิวเตอร์ ซึ่งการใช้ค่าคงที่จะทำให้เกิดความสะดวกในการพัฒนาโปรแกรมเมื่อต้องเปลี่ยนค่าตัวเลขที่มีค่าซ้ำๆกันหลายครั้งในโปรแกรม ซึ่งหากต้องการเปลี่ยนค่าคงที่จะสามารถเปลี่ยนค่าคงที่ที่ประกาศไว้เพียงจุดเดียว ไม่ต้องไล่เปลี่ยนค่าทุกๆ จุดที่ใช้ค่าคงที่นั้น โดยการประกาศค่าคงที่มีรูปแบบการประกาศค่างคงที่ดังนี้

***Const ชนิดของข้อมูล ชื่อค่าคงที่ = ค่า;***

เช่น const float pi =3.1215; // หมายถึง ค่าคงที่ pi เป็นค่าคงที่ชนิด float มีค่าเท่ากับ
// 3.1415
const int Max = 15; // หมายถึง ค่างคงที่ Max เป็นค่างที่ชนิด int มีค่าเท่ากับ 15

### 2.1.3 การคอมเมนท์

การคอมเมนท์ (Comments) คือข้อความหมายเหตุที่คอมไพเลอร์ไม่นำไปคอมไพล์ร่วมด้วยเป็นหมายเหตุที่ผู้พัฒนาโปรแกรมสามารถใช้เป็นข้อความที่สื่อถึงแนวคิด หรือความหมายการใช้คำสั่งในบรรทัดต่างๆ ทำให้ผู้พัฒนาโปรแกรมสามารถระลึกแนวคิดในการพัฒนาโปรแกรมได้เมื่อย้อนกลับมาตรวจดูโปรแกรมอีกครั้งหลังจากเว้นจากการพัฒนาโปรแกรมนั้นๆมานาน และสามารถใช้เป็นข้อความสื่อสารถึงเหตุผลในการใช้คำสั่งต่างๆ ไปยังนักพัฒนาโปรแกรมคนอื่นๆ หากมีการพัฒนาโปรแกรมร่วมกันมากกว่าหนึ่งคน การคอมเมนท์มีรูปแบบการใช้งาน 2แบบ คือ

1) การคอมเมนท์แบบบรรทัดเดียว เป็นการคอมเมนท์ให้จบภายในหนึ่งบรรทัด ส่วนใหญ่จะคอมเมนท์ไว้หลังการใช้คำสั่งหรือฟังก์ชัน โดยจะใช้เครื่องหมาย // และตามด้วยข้อความที่ต้องการคอมเมนท์หรือหมายเหตุ ตัวอย่างเช่น

```
int     flag;          // flag   show status  loop
```

หมายความว่า ข้อความ flag show status loop เป็นเพียงข้อความอธิบายความหมายไม่ใช่เป็นคำสั่งที่คอมพิวเตอร์ต้องนำไปประมวลผล

2) การคอมเมนท์แบบหลายบรรทัด เป็นการคอมเมนท์หลายๆบรรทัด ซึ่งอาจจะเป็นข้อความบรรยาย จุดประสงค์ หรือกระบวนการทำงานของโปรแกรม ซึ่งจำเป็นต้องใช้พื้นที่ในการอธิบายหลายบรรทัดโดยข้อความคอมเมนท์จะต้องอยู่ภายในเครื่องหมาย */ และ /* ตัวอย่างเช่น

```
*/This is Example Comments
  For  C Programming  */
```

ซึ่งข้อความหรือคำสั่งต่างๆ ที่อยู่ระหว่างเครื่องหมาย /* และ */ คอมไพเลอร์จะไม่นำไปทำการคอมไพล์

### 2.1.4. ฟังก์ชัน

ฟังก์ชัน (Function) คือกลุ่มคำสั่งต่างๆที่ถูกลำดับการทำอย่างใดอย่างหนึ่งไว้ ภายในเครื่องหมาย { และ } ซึ่งมี 2 ลักษณะคือฟังก์ชันหลักและฟังก์ชันย่อย โปรแกรมภาษาซีจะประกอบด้วยฟังก์ชันหลัก และฟังก์ชันย่อยต่างๆ โดยฟังก์ชันแรกที่จะเริ่มต้นทำงาน คือฟังก์ชันหลัก main() เสมอ นั่นคือโปรแกรมจะต้องมีฟังก์ชันชื่อว่า main( ) ซึ่งเป็นฟังก์ชันหลัก จึงจะสามารถทำการ คอมไพล์โปรแกรมผ่านได้ ซึ่งลักษณะของฟังก์ชัน main( ) จะต้องเป็นฟังก์ชันที่ไม่มีการส่งค่าไปยังฟังก์ชันอื่นและจะต้องไม่มีการส่งค่ากลับมายังชื่อฟังก์ชัน โดยเราสามารถใช้คำว่า void นำหน้าฟังก์ชัน main( ) สามารถเขียนได้ดังนี้ void main(void)

```
void main(void)
{
  คำสั่งประกาศตัวแปร;
  คำสั่งหรือฟังก์ชันต่าง ๆ;
}
```

ฟังก์ชันย่อยเป็นโปรแกรมย่อยเพื่อทำงานอย่างใดอย่างหนึ่ง เมื่อโปรแกรมหลักต้องการเรียกใช้งาน โดยฟังก์ชันย่อยมี 2 ลักษณะคือฟังก์ชันที่คอมไพเลอร์เตรียมไว้ให้และฟังก์ชันที่ผู้พัฒนาสร้างเอง ซึ่งโครงสร้างของฟังก์ชันนั้นจะประกอบไปด้วยคำสั่งต่างๆ และอยู่ภายในเครื่องหมาย { และ } เสมอ

## 2.2 ชนิดข้อมูลและตัวแปร

ตัวแปรคือชื่อเรียกหน่วยความจำที่กำหนดขึ้นเป็นชื่อต่างๆที่ผู้พัฒนาตั้งขึ้นเพื่อความสะดวกในการเรียกชื่อหน่วยความจำที่ใช้เก็บค่าข้อมูลที่ได้จากการประมวลผลของโปรแกรม ซึ่งเมื่อมีการประกาศตัวแปร จะเป็นการจองหน่วยความจำจำนวนหนึ่งสำหรับเก็บข้อมูล ดังนั้นการประกาศตัวแปรจะต้องระบุด้วยว่าเป็นตัวแปรข้อมูลชนิดใด เนื่องจากตัวแปรแต่ละชนิดจะมีการจองหน่วยความจำไม่เท่ากัน ซึ่งชนิดข้อมูลของตัวแปร เช่น char int float และ double และมีรูปแบบการประกาศตัวแปรโดยทั่วไปดังนี้

***ชนิดข้อมูล ชื่อตัวแปร;***

เช่น

```
char    ch;     // ประกาศตัวแปร ch เป็นข้อมูลชนิด char
int     i;      // ประกาศตัวแปร i เป็นข้อมูลชนิด int
```

### 2.2.1 ชนิดข้อมูล

ชนิดข้อมูลในการเขียนโปรแกรมด้วยภาษาซี จะเป็นตัวกำหนดขนาดความจุให้แก่ตัวแปรนั้นๆ ดังนั้นการเลือกใช้ชนิดของข้อมูลจึงต้องใช้ความระวังในการเลือกชนิดข้อมูล ซึ่งหากเลือกชนิดข้อมูลที่ต้องจองหน่วยความจำไม่ถูกต้อง จะทำให้ตัวแปรนั้นเก็บค่าข้อมูลได้ไม่ถูกต้อง ซึ่งชนิดข้อมูลและขนาดหน่วยความจำดังแสดงในตารางที่ 2.1

**ตารางที่ 2.1** ชนิดข้อมูลต่างๆ

| ชนิดข้อมูล | ข้อมูลที่เก็บ | ขนาดหน่วยความจำ | จำนวนที่สามารถเก็บค่า |
|---|---|---|---|
| char | ตัวเลขหรืออักขระ | 1 ไบต์ | -128 ถึง 127 |
| int | เลขจำนวนเต็ม | 2 ไบต์ | 32768 ถึง 32767 |
| float | เลขจำนวนจริง | 4 ไบต์ | $3.4x10^{-38}$ ถึง $3.4x10^{38}$ |
| double | เลขจำนวนจริง | 8 ไบต์ | $1.7x10^{-308}$ ถึง $1.7x10^{308}$ |

เช่น

```
int  a, b, c=7;          /* เป็นการประกาศตัวแปร  a, b, และ c,  เป็น  int  และ
                            กำหนด ค่าตัวแปร  c  มีค่า  7  */
float pi = 3.14;         /* ประกาศตัวแปร pi เป็นตัวแปรชนิด float และมีค่าเป็นเลข
                            ทศนิยม 3.14 */
```

นอกจากนี้เพื่อความคล่องตัวและความสามารถในการเก็บข้อมูลเพิ่มเติม ในภาษาซียังมีตัวแปรชนิดข้อมูลแบบคิดเครื่องหมายบวกหรือเครื่องหมายลบ (Signed) และไม่คิดเครื่องหมายเพิ่มเติมขึ้นเพื่อความสะดวกในการจัดเก็บข้อมูลต่างๆ ดังในตารางที่ 2.1

**ตารางที่ 2.2** ชนิดข้อมูลประเภทคิดเครื่องหมายและไม่คิดเครื่องหมาย

| ชนิดข้อมูล | ความหมาย | ขนาดหน่วยความจำ | จำนวนที่สามารถเก็บค่า |
|---|---|---|---|
| char | คิดเครื่องหมาย | 1 ไบต์ | -128 ถึง 127 |
| int | คิดเครื่องหมาย | 2 ไบต์ | -32,768 ถึง 32,767 |
| short | คิดเครื่องหมาย | 2 ไบต์ | -32,768 ถึง 32,767 |
| short int | คิดเครื่องหมาย | 2 ไบต์ | -32,768 ถึง 32,767 |
| long | คิดเครื่องหมาย | 4 ไบต์ | -2,147,483,648 ถึง 2,147,483,647 |
| long int | คิดเครื่องหมาย | 4 ไบต์ | -2,147,483,648 ถึง 2,147,483,647 |
| unsigned char | ไม่คิดเครื่องหมาย | 1 ไบต์ | 0 ถึง 255 |
| unsigned | ไม่คิดเครื่องหมาย | 2 ไบต์ | 0 ถึง 65,535 |
| unsigned int | ไม่คิดเครื่องหมาย | 2 ไบต์ | 0 ถึง 65,535 |
| unsigned short | ไม่คิดเครื่องหมาย | 2 ไบต์ | 0 ถึง 65,535 |
| unsigned long | ไม่คิดเครื่องหมาย | 4 ไบต์ | 0 ถึง 4,294,967,295 |
| signed | คิดเครื่องหมาย | 2 ไบต์ | -32,768 ถึง 32,767 |
| signed int | คิดเครื่องหมาย | 2 ไบต์ | -32,768 ถึง 32,767 |
| signed short | คิดเครื่องหมาย | 2 ไบต์ | -32,768 ถึง 32,767 |
| long | คิดเครื่องหมาย | 4 ไบต์ | -2,147,483,648 ถึง 2,147,483,647 |

### 2.2.2 กฎการตั้งชื่อตัวแปร

ตัวแปรที่จะประกาศเพื่อใช้เก็บข้อมูลนั้นจะต้องมีการตั้งชื่อเพื่อการเรียกใช้งานให้สะดวกจะต้องตั้งชื่อให้มีความหมายเพื่อง่ายต่อการทำความเข้าใจ และควรตั้งให้ถูกต้องตามเงื่อนไขดังนี้

1) ต้องขึ้นต้นด้วยตัวอักขระ a ถึง z, A ถึง Z หรือขีดล่าง (Under Score) เท่านั้น ไม่ควรขึ้นต้นด้วยตัวเลข

2) ชื่อที่ตั้งขึ้นห้ามเว้นวรรค

3) ห้ามตั้งชื่อซ้ำภายในฟังก์ชันเดียวกัน

4) ไม่ควรตั้งชื่อที่ยาวเกินไป

5) ชื่อตัวแปรต้องไม่มีอักษรพิเศษเหล่านี้ประกอบในชื่อตัวแปร เช่น !, @, #, $, %, ^, &,* เป็นต้น

6) ห้ามตั้งชื้อตัวแปรซ้ำกับคำสงวนของภาษาซี

### 2.2.3 คำสงวน

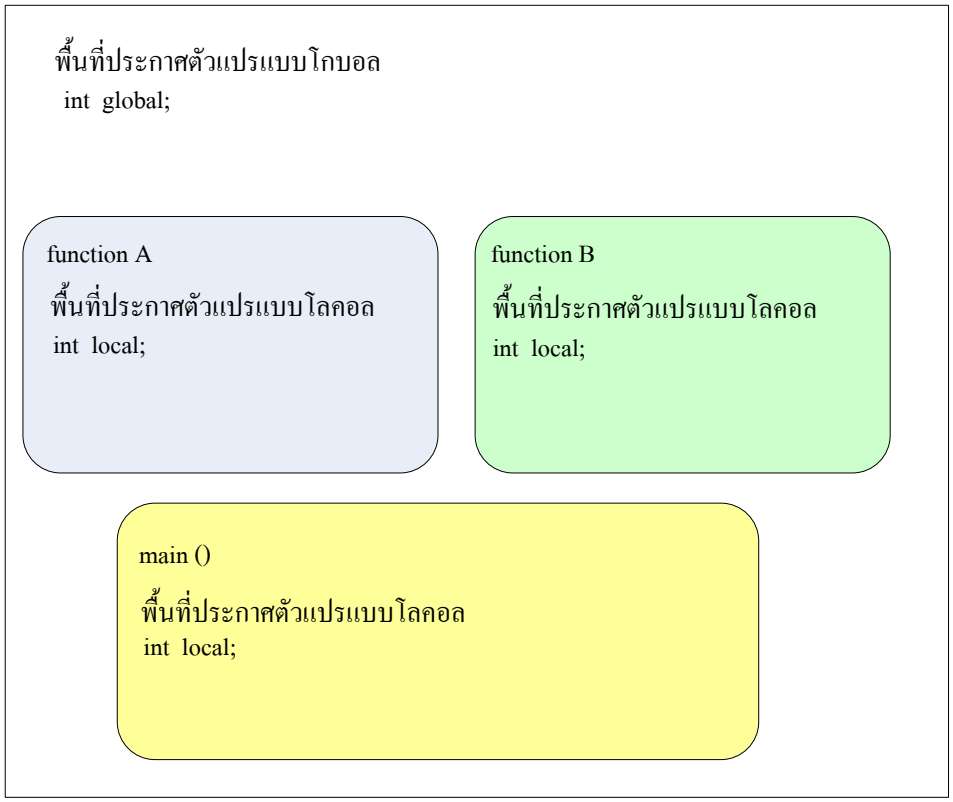
คำสงวนหรือคีย์เวิร์ด (Keyword) เป็นชื่อเรียก หรือคำที่สงวนไว้ในภาษาซี ไม่สามารถนำชื่อเหล่านี้ไปตั้งชื่อตัวแปร ค่าคงที่ และชื่อฟังก์ชัน หากการตั้งชื่อตัวแปรไปซ้ำกับคำสั่งหรือคำสงวนเหล่านี้ จะทำให้คอมไพเลอร์ไม่สามารถแยกได้ว่า ข้อความใดเป็นคำสั่งหรือตัวแปร จะทำให้เกิดความผิดพลาด ขณะคอมไพล์ โดยคำสงวนในภาษาซีมีดังต่อไปนี้

**ตารางที่ 2.3** คำสงวนในภาษาซี

| | | | |
|---|---|---|---|
| auto | double | int | struct |
| break | else | long | switch |
| case | enum | register | typedef |
| const | float | short | unsigned |
| continue | for | signed | void |
| defaule | goto | sizeof | volatile |
| do | if | static | while |

**2.2.4 ขอบเขตการใช้ตัวแปร**

การประกาศตัวแปรในภาษาซีมี 2แบบคือ ตัวแปรแบบโกบอล(Global Variable) และตัวแปรแบบโลคอล (Local Variable) โดยตัวแปรโกลบอลเป็นตัวแปรที่ต้องประกาศใช้งานก่อนฟังก์ชัน main() ซึ่งค่าเริ่มต้นของค่าตัวแปรมีค่าเป็น 0 โดยอัตโนมัติ ตัวแปรแบบโกลบอลนี้ทุกๆ ฟังก์ชันแม้ฟังก์ชันย่อยสามารถเรียกใช้งานได้ โดยข้อมูลในตัวแปรยังเป็นค่าปัจจุบันหรือค่าเดิมก่อนถูกเรียกใช้ตัวแปร ส่วนตัวแปรแบบโลคอล เป็นตัวแปรที่ประกาศในฟังก์ชันย่อย หรือฟังก์ชัน main() ซึ่งตัวแปรแบบโลคอลจะถูกเรียกใช้ได้ภายในฟังก์ชันย่อยที่ประกาศเท่านั้น ไม่สามารถเรียกใช้ข้ามฟังก์ชันได้ดังภาพที่ 2 2. ตัวแปร global เป็นตัวแปรแบบโกบอลดังนั้นฟังก์ชัน A() , ฟังก์ชัน B() หรือฟังก์ชัน main() สามารถเรียกใช้งานได้ แต่ตัวแปรชื่อ Local จะถูกเรียกใช้ได้เฉพาะภายในเฉพาะของแต่ละฟังก์ชัน เช่นตัวแปรแบบโลคอลในฟังก์ชัน A() จะสามารถเรียกใช้ตัวแปรดังกล่าวได้เฉพาะภายใน ฟังก์ชัน A() เท่านั้น ฟังก์ชัน B() หรือฟังก์ชัน Main() ไม่สามารถเรียกใช้งานตัวแปรโลคอลในฟังก์ชัน A() ได้ แม้ชื่อตัวแปรโลคอลในแต่ละฟังก์ชั้นจะชื่อเดียวกันแต่คอมไพเลอร์จะถือว่าเป็นคนละตัวแปร




**ภาพที่ 2.2** ลักษณะขอบเขตของการประกาศตัวแปร

## 2.3 การรับและแสดงผลข้อมูล

การรับและแสดงผลข้อมูลนี้หมายถึงการรับค่าข้อมูลอินพุตจากคีย์บอร์ดให้แก่ตัวแปร และการนำค่าข้อมูลแสดงผลบนจอภาพ โดยใช้ฟังก์ชันที่คอมไพเลอร์เตรียมไว้ให้แล้วคือฟังก์ชัน scanf() และฟังก์ชัน printf()

### 2.3.1 การรับข้อมูล

เป็นการรับข้อมูลจากการป้อนค่าด้วยคีย์บอร์ดให้แก่ตัวแปรด้วยฟังก์ชัน scanf() เพื่อนำไปสู่การประมวล การคำนวณ หรือเป็นเงื่อนไข ทั้งนี้ขึ้นกับลักษณะการเขียนโปรแกรมรับข้อมูลไปทำอะไร โดยมีรูปแบบการใช้ดังนี้

***scanf("รูปแบบ", อาร์กิวเมนต์1, อาร์กิวเมนต์2,….);***

โดยที่ รูปแบบ หมายถึงชนิดข้อมูลที่จะเก็บในตัวแปรอาร์กิวเมนต์ เช่นจำนวนเต็ม จำนวนจริง หรือสตริง และอาร์กิวเมนต์ หมายถึงตัวแปรที่เก็บค่าข้อมูลที่รับซึ่งผู้เขียนโปรแกรมจะต้องกำหนดชนิดของข้อมูลที่ต้องการรับให้สอดคล้องกับข้อมูลที่จะป้อนจากคีย์บอร์ด และกำหนดตัวแปรเก็บข้อมูลในตำแหน่งอาร์กิวเมนต์ เช่น ต้องการรับข้อมูลเดือน และปี เป็นจำนวนเต็ม จะมีลักษณะการใช้คำสั่งดังนี้

```
int month, year;  // ประกาศตัวแปรก่อนนำไปใช้
scanf("%d %d",&month,%&year);
```

จากตัวอย่าง รูปแบบคือ "%d %d " กำหนดให้รู้ว่าต้องรับข้อมูลจำนวนเต็ม 2 ตัว โดยที่ข้อมูลนั้นแยกออกจากกันด้วยช่องว่าง %d จะรับข้อมูลเลขจำนวนเต็มแล้วไปเก็บไว้ในตำแหน่งของตัวแปร &month และ %d ตัวที่สองจะรับข้อมูลไปเก็บไว้ในตำแหน่งของตัวแปร &year

รูปแบบการรับข้อมูลในภาษาซี จะขึ้นต้นด้วยสัญลักษณ์ % โดยมีรูปแบบการรับข้อมูลและรูปแบบการใช้งานดังในตารางที่ 2.4

**ตารางที่ 2.4** รูปแบบการรับข้อมูล

| รูปแบบ | ลักษณะรูปแบบการใช้งาน |
|---|---|
| %d, %l | รับข้อมูลเป็นจำนวนเต็มและแปลงข้อมูลเป็น int |
| %ld | รับข้อมูลเป็นจำนวนเต็มและแปลงข้อมูลเป็น long |
| %u | รับข้อมูลเป็นจำนวนเต็มบวกและแปลงข้อมูล unsigned |
| %o | รับข้อมูลเป็นจำนวนเต็มบวกของเลขฐาน 8 และแปลงข้อมูลเป็น unsigned |
| %x | รับข้อมูลเป็นจำนวนเต็มบวกของเลขฐาน 16 และแปลงข้อมูลเป็น unsigned |
| %c | รับข้อมูลเป็นอักขระ 1 ตัว |
| %s | รับข้อมูลเป็นสตริง |
| %f | รับข้อมูลเป็นจำนวนจริงและแปลงข้อมูลเป็น float |
| %lf | รับข้อมูลเป็นจำนวนจริงและแปลงข้อมูลเป็น double |

### 2.3.2 การแสดงผลข้อมูล

การแสดงผลข้อมูลเป็นการนำข้อความ ค่าข้อมูลของตัวแปรมาแสดงผลบนจอภาพ เพื่อให้ผู้ใช้งานสามารถเห็นค่าข้อมูลทางจอภาพด้วยคำสั่ง printf() ซึ่งมีรูปแบบการใช้งานดังนี้

***printf("ข้อความ %รูปแบบ" , อาร์กิวเมนต์1, อาร์กิวเมนต์2,……);***

โดยที่ ข้อความ หมายถึง ข้อความสตริงที่กำหนดไว้ ในคำสั่ง printf()

"%รูปแบบ" หมายถึง ชนิดของข้อมูลและตำแหน่งที่ที่ต้องการให้แสดงผล และ อาร์กิวเมนต์ หมายถึง ค่าข้อมูลที่ต้องการให้แสดงผล เช่น

```
int age =30;
printf("I 'm  %d  year old",age);
```

ผลลัพธ์ที่จอภาพที่ได้จะแสดงข้อความ I'm 30 year old โดย %d จะเป็นรูปแบบการแสดงผลตัวเลขจำนวนเต็ม จากตัวแปร age โดยรูปแบบการแสดงผลยังมีหลายรูปแบบ เช่นการแสดงตัวเลขจำนวนเต็ม การแสดงตัวอักษร โดยมีรูปแบบการแสดงผลดังตารางที่ 2.5

**ตารางที่ 2.5** รูปแบบการแสดงผล

| รูปแบบ | ลักษณะรูปแบบการใช้งาน |
|---|---|
| %c | ใช้กำหนดตำแหน่งที่จะแสดง อักขระ 1 ตัว (single character) |
| %d | ใช้กำหนดตำแหน่งแสดงเลขจำนวนเต็ม(integerหรือ int ) 1 จำนวนในรูปเลขฐานสิบ |
| %e | ใช้แสดงตัวเลขที่เป็นจุดทศนิยม (floating point ,float)ในรูป e เช่น 2.15e+2 คือแทนค่า 215 |
| %f , %lf | ใช้กับข้อมูลเป็น float และ double |
| %g | ใช้กับข้อมูล float |
| %h | ใช้กับ short integer |
| %o | ใช้แสดง integer ในรูปเลขฐานแปด |
| %x | ใช้แสดง integer ในรูปเลขฐานสิบหก |
| %s | ใช้แสดงข้อความ |
| %p | ใช้แสดง address ของตัวแปรพอยน์เตอร์ |

นอกจากนี้คำสั่ง printf() ยังมีรหัสควบคุมการแสดงผลเพื่อให้การแสดงผลบนจอภาพสะดวกและง่ายต่อการอ่านมากขึ้น เช่นการแสดงข้อความแล้วขึ้นบรรทัดใหม่ เป็นต้น ซึ่งจะขึ้นต้นรหัสควบคุมด้วยเครื่องหมาย " \ " โดยมีรหัสการใช้งานดังตารางที่ 2.6

**ตารางที่ 2.6** รหัสควบคุมการแสดงผลบนจอภาพร่วมกับคำสั่ง printf()

| รหัสควบคุม | ความหมาย |
|---|---|
| \a | ส่งเสียงบี๊พ |
| \b | ลบตัวอักษรที่อยู่ด้านซ้าย1ตำแหน่ง (backspace) |
| \t | เลื่อนตำแหน่งในการพิมพ์ครั้งต่อไป 1 แท็บ(แนวราบ horizontal) |
| \n | ขึ้นบรรทัดใหม่ |
| \v | เลื่อนตำแหน่งไป 1 แท็บแนวดิ่ง |
| \r | เลื่อนไปต้นบรรทัด |
| \" | แสดงเครื่องหมาย " |
| \' | แสดงเครื่องหมาย ' |
| \? | แสดงเครื่องหมาย ? |

นอกจากนี้คำสั่ง printf() จะใช้ในการแสดงข้อความที่ต้องการบนจอภาพได้โดยไม่จำเป็นต้องส่งค่าตัวแปร หรืออาร์กิวเมนต์ เช่น

```
printf("Hello Goodmorning \n");
```

ผลลัพธ์ที่ได้บนจอภาพคือ

Hello Goodmorning และเคอร์เซอร์จะขึ้นบรรทัดใหม่

**ตัวอย่างที่ 2.1** โปรแกรมการรับข้อมูลและแสดงผลบนจอภาพ

```
#include <stdio.h>
int  age;
void main()
{
  printf("Hello  Goodmorning\n");
                // แสดงข้อความ Hello Goodmornig แล้วขึ้นบรรทัดใหม่
  printf("Please Enter number your old : ");
  scanf("%d",&age);
  printf("\n Your are %d  years old",age);
}
```

เมื่อทำการรันโปรแกรม จะลำดับการทำงานและแสดงข้อความดังภาพที่ 2.3

```
Hello  Goodmorning
Please Enter number your old : 20
Your are 20 years old
```

**ภาพที่ 2.3** ผลลัพธ์บนจอภาพของโปรแกรมในตัวอย่างที่ 2.1

**ตัวอย่างที่ 2.2** โปรแกรมบวกเลขจำนวนเต็ม

```
#include <stdio.h>
int  x,y,z;
void main()
{
  printf("Please Enter first Number =");   scanf("%d",&x);
                  // แสดงข้อความให้พิมพ์ตัวเลขค่าแรก  และรอรับค่าให้กับตัวแปร x
  printf("\nPlease Enter second Number =");   scanf("%d",&y);
                  // แสดงข้อความให้พิมพ์ตัวเลขค่าที่สอง  และรอรับค่าให้กับตัวแปร y
  z = x + y;  //  นำค่าตัวแปร x บวกกับ y  และเก็บไว้ในตัวแปร z
  printf(" \n Result X + Y = %d ",z);
                  // แสดงผลลัพธ์ของการบวกระหว่างค่า x และ y
}
```

เมื่อทำการรันโปรแกรม จะมีลำดับการทำงานและแสดงข้อความดังภาพที่ 2.4 และให้ทดลองป้อนตัวเลขแก่โปรแกรม

```
Please Enter first Number =5
Please Enter second Number =13
Result X + Y =18
```

**ภาพที่ 2.4** ผลลัพธ์บนจอภาพของโปรแกรมในตัวอย่างที่ 2.2

## แบบฝึกหัดท้ายหน่วยเรียนที่ 2

1) ให้หาข้อผิดพลาดของการประกาศตัวแปรต่อไปนี้ และแก้ไขให้

```
int  $dollar, num1-3, birth day;
char  x,y,x;
float  a, float;
int i,j
```

2) ให้หาจุดผิดพลาดของโปรแกรมต่อไปนี้ และแก้ไขให้ถูกต้อง

```
#include <stdio.h>
 void main {
 float a1:
printf("Enter number:");
scanf("s",a2)
printf("Your number is %.2d" a2);
```

3) ให้แสดงผลลัพธ์ของการทำงานของโปรแกรมต่อไปนี้

```
#include <stdio.h>
void main() {
char  newline = '\n';
printf("Start Program");
printf("%c",newline);
printf("***This is My first Program ***\n");
printf("I will try it");
}
```

4) ให้เขียนโปรแกรม แสดงข้อความต่อไปนี้บนจอภาพ

```
Mr.Somchai   Longla      21  years
Grade Point Average      3.5
Height                              1.65  m
Weight                              62.5  kg
Computer Programming  A
```

# ใบงานที่ 2
# การแสดงผลและรับข้อมูล

## จุดประสงค์

1) ทดลองการใช้คำสั่งแสดงผลบนจอภาพ
2) ทดลองเขียนคำสั่งรับข้อมูลจากคีย์บอร์ด

คอมพิวเตอร์ประกอบไปด้วยส่วนประกอบต่างๆเช่น ซีพียู (CPU) หน่วยความจำ (RAM) เมนบอร์ด จอภาพ และคีย์บอร์ด และพอร์ตอินพุตเอาท์พุตแบบต่างๆ ซึ่งโปรแกรมที่จะทำการเขียนขึ้นมานั้นจะถูกประมวลผลโดยซีพียู ซึ่งอาจจะต้องนำข้อมูลที่เป็นอินพุตจากคีย์บอร์ดและนำไปประมวลผล แล้วแสดงผลนั้นทางจอภาพเพื่อให้เราได้ทราบผลลัพธ์ที่ได้จากการประมวลผล และวนรับข้อมูลจากการป้อนข้อมูลเป็นวัฏจักรไม่สิ้นสุดจนกว่าจะจบกระบวนการทำงาน ซึ่งในภาษาซีก็จะมีคำสั่งที่เกี่ยวข้องกับอุปกรณ์อินพุตจากคีย์บอร์ด และคำสั่งแสดงข้อความทางจอภาพเพื่อให้การรับค่าอินพุตและแสดงผลลัพธ์มีวัฏจักรการทำงาน

## การทดลองที่ 2.1 การแสดงข้อความบนจอภาพ

การแสดงข้อความบนจอภาพเป็นการใช้คำสั่งหรือฟังก์ชัน printf() เพื่อแสดงข้อความตัวเลข หรือสัญลักษณ์ที่อยู่ภายในเครื่องหมาย " ข้อความ" ที่อยู่ในเครื่องหมายวงเล็บ บนจอภาพ เช่นต้องการแสดงข้อความ Good morning ก็จะใช้คำสั่ง printf("Good morning"); โดยจะเริ่มแสดงข้อความในบรรทัดแรกที่ด้านซ้ายบนสุดของจอภาพ เป็นลำดับแรก และหากต้องการแสดงข้อความใดบนจอภาพก็สามารถใช้คำสั่ง printf() โดยมีการทดลองดังนี้

1) เปิดโปรแกรมคอมไพเลอร์ Turbo C และ สร้างไฟล์ใหม่
2) เขียนโปรแกรมลงในอีดิตเตอร์ตามตัวอย่างโปรแกรมดังนี้

```
#include <stdio.h>
#include<conio.h>
void main(void)
```

```
{
  clrscr();
  printf(" ****************************\n");
  printf("*    This is My first C Programing    *\n");
  printf("****************************");
}
```

3) บันทึกไฟล์ในเป็นไฟล์ program2_1.c

4) เลือกคำสั่ง Compile และให้บันทึกผล

……

……

……

5) เลือกคำสั่ง RUN และให้บันทึกผลการทดลอง

……

……

……

……

……

## การทดลองที่ 2.2 การรับค่าอินพุตและแสดงผล

การรับค่าอินพุตเป็นการรับค่าข้อมูลจากการกดปุ่มคีย์บอร์ดไปเก็บไว้ในตัวแปรที่เขียนโปรแกม ด้วยคำสั่ง scanf(); แล้วเก็บไว้ในตัวแปร เช่น scanf(%d,&num); เป็นการรับค่าตัวเลขจำนวนเต็มบวก แล้วเก็บไว้ในตัวแปร num โดยสามารถนำค่าอินพุตที่รับค่าเข้ามาแสดงผลบนจอภาพด้วยคำสั่ง printf() ซึ่งมีลำดับขั้นตอนการทดลองดังนี้

1) เปิดโปรแกรมคอมไพเลอร์ Turbo C และ สร้างไฟล์ใหม่

2) เขียนโปรแกรมลงในอีดิตเตอร์ตามตัวอย่างโปรแกรมดังนี้

```
#include<stdio.h>
#include<conio.h>
void main()
{
```

```
int radius;
float result;
 clrscr();
 printf(" *********************************\n");
 printf("*    This is Find  Circle  Area Program    *\n");
 printf("*********************************\n");
 printf("\n");
 printf("\n");
 printf(" Please Enter the meter of Radius = ");  scancf("%f",&num1);

 result = pi*radius*radius;
 printf("\nThe circle Area is =%.2f square meter ",result);
 getch();
}
```

3) บันทึกไฟล์ในเป็นไฟล์ program2_2.c

4) เลือกคำสั่ง Compile และให้บันทึกผล

……

……

……

……

……

5) เลือกคำสั่ง RUN และให้บันทึกผลการทดลอง

……

……

……

……

……

6) สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

**งานที่มอบหมาย**

1) จงเขียนโปรแกรมให้แสดงข้อความดังนี้บนจอภาพ

```
      HHH HHH HHH                 HHH HHH HHH
   HHH         HHH             HHH         HHH
   HHH         HHH             HHH         HHH
   HHH         HHH             HHH         HHH
   HHH HHH HHH HHH HHH         HHH HHH HHH HHH HHH
               HHH             HHH         HHH
               HHH             HHH         HHH
               HHH             HHH         HHH
      HHH HHH HHH                 HHH HHH HHH
```