

แผนการสอนประจำบทที่ 1

ชนิดของข้อมูลและนิพจน์ในภาษาซี

หัวข้อสำคัญ

1. ความหมายของภาษาซี
2. ประเภทของตัวแปลภาษาคอมไพเลอร์
3. โครงสร้างโปรแกรมภาษาซีและชนิดของข้อมูล (Data Types)
4. หลักการตั้งชื่อตัวแปร (Variable) และคำสงวน (Reserved Word)
5. การประกาศตัวแปร (Declaration) และการใช้นิพจน์คณิตศาสตร์ (Expression)

วัตถุประสงค์เชิงพฤติกรรม

1. ผู้เรียนสามารถอธิบายความหมายของภาษาซีและโครงสร้างโปรแกรมภาษาซีได้
2. ผู้เรียนสามารถบอกประเภทและอธิบายการทำงานของตัวแปลภาษาคอมไพเลอร์ได้
3. ผู้เรียนสามารถเขียนโปรแกรมประกาศตัวแปรและชนิดข้อมูลของตัวแปรได้

วิธีการสอนและกิจกรรมการเรียนการสอน

1. การบรรยาย
2. การทำแบบฝึกหัด

สื่อที่ใช้ประกอบการสอน

1. เอกสารประกอบการสอน
2. เครื่องคอมพิวเตอร์

การวัดและประเมินผล

1. สังเกตจากความสนใจของผู้เรียน
2. ประเมินจากการตอบคำถามของผู้เรียนและกิจกรรมในชั้นเรียน
3. การทำแบบฝึกหัดท้ายบท

บทที่ 1

ชนิดของข้อมูลและนิพจน์ในภาษาซี

1.1 ความหมายของภาษาซี (C Language)

ผู้คิดค้นพัฒนาโปรแกรมคอมพิวเตอร์ภาษาซีขึ้นคือ นายเดนนิส ริทชี (Dennis Ritchie) ที่ ศูนย์วิจัยเบล (Bell Laboratories) ประเทศสหรัฐอเมริกาเมื่อปี ค.ศ.1972 เป็นภาษาคอมพิวเตอร์ที่ใช้เขียนระบบปฏิบัติการยูนิกซ์ ซึ่งใช้กันแพร่หลายในระบบเครือข่ายคอมพิวเตอร์ในปัจจุบัน ลักษณะเด่น คือ เป็นภาษาที่มีความใกล้เคียงกับภาษาระดับต่ำ (Low-Level Language) จึงทำให้นักพัฒนาโปรแกรมสามารถที่จะกำหนดรายละเอียดของโปรแกรมให้เข้าถึงการทำงานในส่วนต่าง ๆ ของคอมพิวเตอร์ให้มากที่สุดเพื่อให้เกิดความเร็วในการทำงานสูงสุด และเป็นภาษาระดับสูง (High-Level Language) ทำให้ผู้พัฒนาสามารถที่จะพัฒนาโปรแกรมได้ โดยเน้นไปที่การแก้ปัญหาที่ต้องการได้อย่างอิสระโดยไม่ต้องคำนึงถึงฮาร์ดแวร์ใดๆ



ภาพ 1.1 : Dennis Ritchie

ที่มาภาพ 1.1 : <https://www.dailytut.com/linux/god-of-programming-dennis-ritchie-1941-2011.html>

ข้อดี-ข้อเสียของภาษา C

ข้อดี

1. เป็นภาษาที่มีลักษณะเป็นโครงสร้างจึงเขียนโปรแกรมง่าย
2. ชุดคำสั่งสามารถสั่งงานคอมพิวเตอร์ทำงานได้ รวดเร็วกว่าภาษาระดับสูงอื่น ๆ
3. รองรับคำสั่งงานอุปกรณ์ในระบบคอมพิวเตอร์ได้เกือบทุกส่วนของฮาร์ดแวร์

4. คอมไพเลอร์ของภาษาซีทุกโปรแกรมในท้องตลาดจะทำงานอ้างอิง มาตรฐาน (ANSI= American National Standards Institute)
5. สามารถเขียนโปรแกรมประยุกต์ได้หลายระดับ
 - 1) เขียนโปรแกรมจัดระบบงาน (OS) คอมไพเลอร์ของภาษาอื่น
 - 2) โปรแกรมปัญญาประดิษฐ์ (AI = Artificial Intelligent)
 - 3) คำนวณงานทางด้าน วิทยาศาสตร์และวิศวกรรมศาสตร์

ข้อเสีย

ไม่มีตัวจัดการจองหน่วยความจำในตัวเอง เมื่อเวลาเราต้องการจองหน่วยความจำแบบ Dynamic ภาษา C ทำ wrapper เพื่อติดต่อกับ OS เพื่อขอจองหน่วยความจำโดยตรง ซึ่งหน่วยความจำนั้นก็จะถูกจองไปเรื่อยๆ จนกว่าจะมีการยกเลิก ปัญหานี้เรียกว่าหน่วยความจำรั่ว หรือ Memory Leak

1.2 ประเภทของตัวแปลภาษาคอมพิวเตอร์

1.2.1 โปรแกรมแปลภาษาแบบแอสเซมเบล (Assembler)

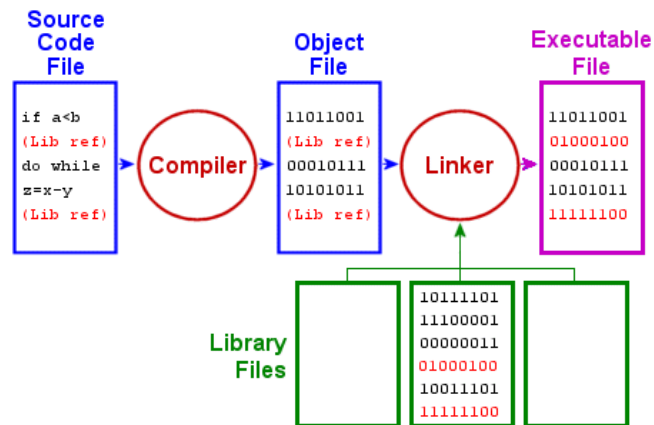
เป็นตัวแปลภาษาแอสเซมบลีซึ่งเป็นภาษาระดับต่ำ ให้เป็นภาษาเครื่อง ตัวอย่างของภาษาแอสเซมบลี แสดงผลลัพธ์เป็นคำว่า “Hello,World!” แสดงดังภาพที่ 1.2 ด้านล่าง

```
mov [02h], 'H'
mov [04h], 'e'
mov [06h], 'l'
mov [08h], 'l'
mov [0ah], 'o'
mov [0ch], ','
mov [0eh], 'W'
mov [10h], 'o'
mov [12h], 'r'
mov [14h], 'l'
mov [16h], 'd'
mov [18h], '!'
```

ภาพ 1.2 : ตัวอย่างของภาษา แอสเซมบลี

1.2.2 โปรแกรมแปลภาษาแบบคอมไพเลอร์ (Compiler)

เป็นตัวแปลภาษาคอมพิวเตอร์ระดับสูงไปเป็นภาษาเครื่อง โดยทำการตรวจสอบความถูกต้องของการเขียนคำสั่งทั้งหมดทั้งโปรแกรมให้เป็น Objected code และจึงการประมวลผลและแสดงผลลัพธ์ของโปรแกรม แต่ถ้าพบข้อผิดพลาดของโปรแกรม ในระหว่างการแปลคำสั่ง เช่น ข้อผิดพลาดทางไวยากรณ์ของโปรแกรมคอมไพเลอร์จะแจ้งข้อผิดพลาดทั้งหมดที่พบ ผู้เขียนโปรแกรมต้องแก้ไขโปรแกรมทั้งหมดถูกต้องทั้งหมดก่อน แล้วจึงคอมไพล์ใหม่อีกครั้ง จนกว่าไม่พบข้อผิดพลาดถึงจะนำโปรแกรมไปใช้งานได้ ขั้นตอนการแปลภาษาแบบคอมไพเลอร์แสดงได้ดังภาพที่ 1.3 เริ่มจากไฟล์ Source code นามสกุล C สั่งให้ตรวจสอบไฟล์ข้อผิดพลาดของทั้งโปรแกรม เพื่อสร้าง Objected code โดยใช้ Compiler เมื่อพบข้อผิดพลาด Compiler จะแจ้งให้ผู้พัฒนาทราบเพื่อแก้ไข เมื่อแก้ไขข้อผิดพลาดเรียบร้อยแล้ว สามารถเข้าสู่การเชื่อมโยงโปรแกรมกับไฟล์ Library ที่เรียกใช้งาน ซึ่งเรียกว่า Linker หลังจากเชื่อมโยงไฟล์ Library เสร็จเรียบร้อยแล้ว สำหรับโปรแกรมภาษาซีจะได้ไฟล์นามสกุล EXE มาเพิ่มอีก 1 ไฟล์ ที่สามารถเรียกใช้งานได้ ภาษาโปรแกรมในกลุ่มนี้ เช่น C, C++, C#, Go เป็นต้น



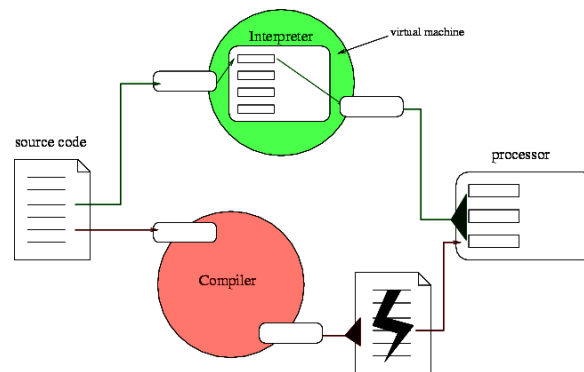
ภาพ 1.3 : การทำงานของตัวแปลภาษา

ที่มาภาพ 1.3 : <https://abhijangda.wordpress.com/2014/07/29/introduction-to-compilers>

1.2.3 โปรแกรมแปลภาษาแบบอินเทอร์พรีเตอร์ (Interpreter)

เป็นตัวแปลภาษาคอมพิวเตอร์ระดับสูงไปเป็นภาษาเครื่อง โดยใช้หลักการแปลคำสั่งครั้งละ 1 คำสั่งให้เป็นภาษาเครื่อง โดยนำคำสั่งที่เป็นภาษาเครื่องนั้นไปทำการประมวลผล และแสดงผลลัพธ์ทันที ถ้าไม่พบข้อผิดพลาดของโปรแกรมจะแปลคำสั่งต่อไปเรื่อย ๆ จนกว่าจะจบโปรแกรม แต่ถ้าพบข้อผิดพลาดของโปรแกรม

ในระหว่างการแปลคำสั่ง เช่น ข้อผิดพลาดทางไวยากรณ์ของภาษา โปรแกรมอินเทอร์พรีเตอร์ก็จะหยุดการทำงาน พร้อมแจ้งข้อผิดพลาด Objected code ของโปรแกรมที่แปลคำสั่งโดยใช้อินเทอร์พรีเตอร์ จะไม่สามารถเก็บไว้ใช้ใหม่ได้ ขั้นตอนการแปลภาษาแบบอินเทอร์พรีเตอร์ แสดงได้ดังภาพที่ 1.4 เริ่มจากไฟล์ Source code สั่งให้ตรวจสอบไฟล์ข้อผิดพลาดของโปรแกรมที่ละบรรทัด โดยใช้ Interpreter เมื่อพบข้อผิดพลาด Interpreter จะแจ้งให้ผู้พัฒนาทราบเพื่อแก้ไข เมื่อแก้ไขข้อผิดพลาดเรียบร้อยแล้ว โปรแกรมจะแสดงผลการทำงานของโปรแกรมที่ละบรรทัด เช่นเดียวกับ Interpreter ภาษาโปรแกรมในกลุ่มนี้ เช่น Python, JavaScript, PHP เป็นต้น



ภาพ 1.4 : ความแตกต่างของ Compiler และ Interpreter

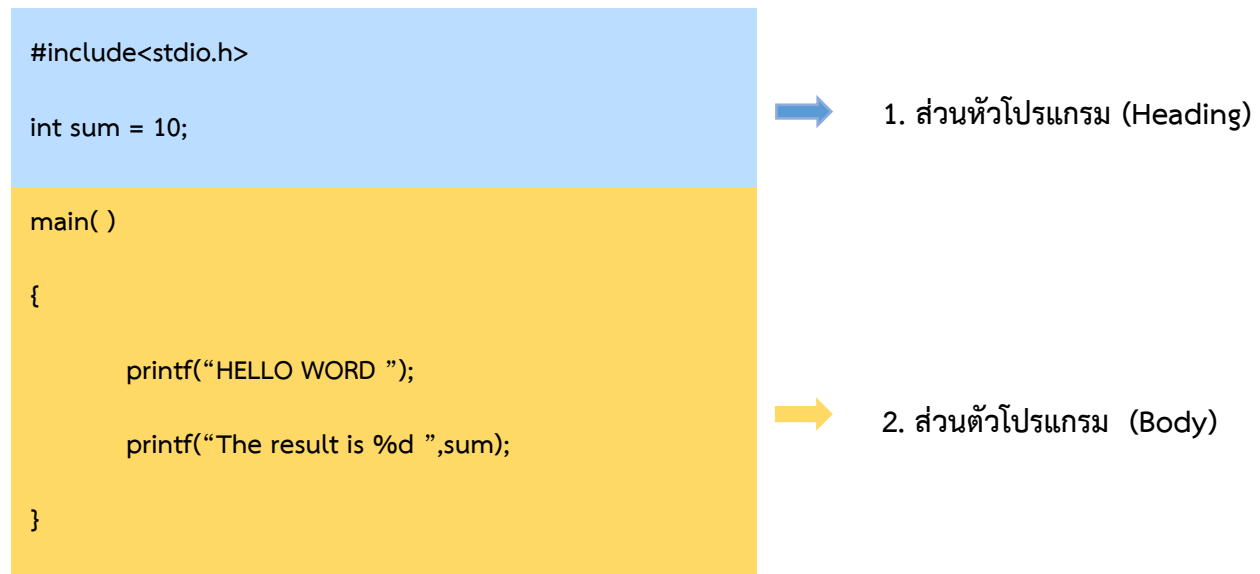
ที่มาภาพ 1.4 : <http://irisclasson.com/2012/07/20/stupid-question-3-what-is-a-compiler-and-an-interpreter-and-what-is-the-difference>

สรุปความแตกต่างของตัวแปลภาษาคอมไพเลอร์

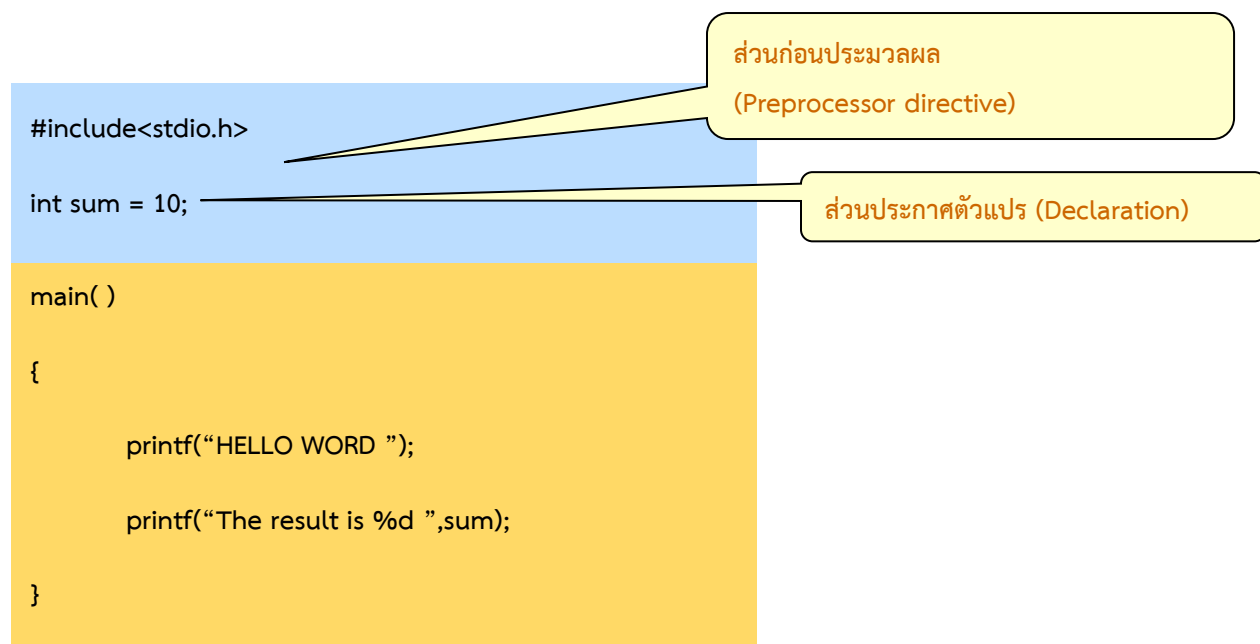
ประเภทตัวแปลภาษา	ข้อดี	ข้อเสีย
คอมไพเลอร์ (Compiler)	1. ทำงานได้เร็ว เนื่องจากทำการแปลผลทีเดียว แล้วจึงทำงานตามคำสั่งของโปรแกรมในภายหลัง	- เมื่อเกิดข้อผิดพลาดขึ้นกับโปรแกรมจะตรวจสอบหาข้อผิดพลาดได้ยาก เพราะทำการแปลผลทีเดียวทั้งโปรแกรม
	2.. เมื่อทำการแปลผลแล้ว ในครั้งต่อไปไม่จำเป็นต้องทำการแปลผลใหม่อีก เนื่องจากภาษาเครื่องที่แปลได้จะถูกเก็บไว้ที่หน่วยความจำ สามารถเรียกใช้งานได้ทันที	
อินเทอร์พรีเตอร์ (Interpreter)	1. หาข้อผิดพลาดของโปรแกรมได้ง่าย เนื่องจากทำการแปลผลทีละบรรทัด	- ช้า เนื่องจากที่ทำงานทีละบรรทัด
	2. เนื่องจากทำงานทีละบรรทัดดังนั้นจึงสั่งให้โปรแกรมทำงานตามคำสั่งเฉพาะจุดที่ต้องการได้	
	3. ไม่เสียเวลาการแปลโปรแกรมเป็นเวลานาน	

1.3 โครงสร้างโปรแกรมภาษาซี

โครงสร้างของโปรแกรมภาษาซีหลักๆ มีทั้งหมด 2 ส่วน ได้แก่ ส่วนหัวโปรแกรม (Heading) และ ส่วนตัวโปรแกรม (Body)



1.3.1. ส่วนหัวโปรแกรม (Heading) ประกอบด้วย 2 ส่วน ได้แก่



1) ส่วนก่อนประมวลผล (Preprocessor directive)

ส่วนที่ระบุให้คอมไพเลอร์เตรียมการทำงานที่กำหนดนี้ก่อนส่วนอื่น หน้าคำสั่งจะขึ้นต้นด้วยเครื่องหมาย # ข้อสังเกต การตั้งชื่อตัวแปรต้องใช้เป็นตัวอักษรพิมพ์ใหญ่เท่านั้น เช่น

- **#include <stdio.h>** ใช้งานเกี่ยวกับนำข้อมูลเข้าและออก
- **#include <math.h>** ใช้งานเกี่ยวกับฟังก์ชันคณิตศาสตร์
- **#define TAX 0.07** เป็นการกำหนดค่าคงที่ให้กับตัวแปร TAX ให้มีค่าเท่ากับ 0.07
- **#define SUBJECT "Calculus"** เป็นการกำหนดค่าคงที่ให้กับตัวแปร SUBJECT ให้มีค่าเท่ากับ Calculus
- **#define NEWTAX TAX +1** เป็นการกำหนดค่าคงที่ให้กับตัวแปร NEWTAX ให้มีค่าเท่ากับ TAX +1 (0.07+1)

2) ส่วนประกาศตัวแปร (Declaration)

เป็นการกำหนดชนิดข้อมูลให้กับตัวแปรที่ใช้ภายในโปรแกรม ถ้าไม่ประกาศตัวแปรจะไม่สามารถนำตัวแปรไปใช้งานได้ การประกาศชื่อตัวแปรเป็นลักษณะเป็น Case-Sensitive (ตัวอักษรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ถือว่าต่างกัน) การประกาศตัวแปรที่ตำแหน่งส่วนก่อนประมวลผลนี้จะส่งผลให้ตัวแปรนี้ทำงานแบบ global variable

รูปแบบคำสั่ง

ชนิดตัวแปร	ชื่อตัวแปร;
------------	-------------

เช่น

- `int book;` เป็นการกำหนดว่าตัวแปร `book` เป็นข้อมูลชนิดจำนวนเต็มหรือ integer เช่น 10,45,3
- `float money;` เป็นการกำหนดว่าตัวแปร `money` เป็นข้อมูลชนิดจำนวนจริง (มีจุดทศนิยม) เช่น 100.00 , 45.75, 50.25

1.3.2 ส่วนตัวโปรแกรม (Body)

โปรแกรมจะเริ่มทำงานที่ main() เป็นต้นไปจนจบ

คำสั่งหลักจะอยู่ภายในเครื่องหมาย {...} หลังคำว่า main()

```
main()
{
    คำสั่งที่1
    คำสั่งที่2
    ....
    ....
    ....
    คำสั่งที่ 10
}
```

ถ้ามีการเรียกใช้ฟังก์ชันที่สร้างขึ้นเอง โปรแกรมจะไปทำงานที่ฟังก์ชันดังกล่าวและจะกลับมาทำงานที่ส่วนโปรแกรมหลักอีกครั้ง

```
main()
{
    คำสั่งที่ 1;
    คำสั่งที่ 2;
    function A() ;
    คำสั่งที่ 3;
    ....
    ....
    ....
    คำสั่งที่ 10;
}

function A()
{
    คำสั่งที่ A1;
    คำสั่งที่ A2;
}
```


1.3.3 ส่วนคำอธิบาย (Comments)

ใช้อธิบายความหมายและหน้าที่ของส่วนต่าง ๆ ของโปรแกรม

ข้อความที่เขียนต้องอยู่ภายในเครื่องหมาย `/* */` เท่านั้น

หรือใช้ `//` เพื่อให้ข้อความที่อยู่หลังเครื่องหมายนี้เป็นส่วนของการอธิบายโปรแกรม

```
1. #include<stdio.h>
2. int sum=5;
3. main( )
4. {      /*My first program */
5.         printf("HELLO WORD "); //comment
6.         printf("The result is %d \n",sum);
7.         /*xxxxxxxxxxxxxxxxxxxx
8.                 This is my comment
9.         xxxxxxxxxxxxxxxxxxxx */
```

สรุปข้อสังเกต

- ใช้ปีกกา {...} เป็นตัวกำหนดขอบเขตของโปรแกรมหลักและฟังก์ชันต่างๆ
- ใช้เครื่องหมาย ; (semi colon) เป็นตัวกำหนดการสิ้นสุดของคำสั่ง
- ใช้เครื่องหมาย , (comma) เป็นตัวคั่นตัวแปรและข้อความ
- ใช้เครื่องหมาย `/* */` และ `//` เป็นการกำหนดข้อความอธิบายโปรแกรมที่ไม่ต้องการให้คอมไพเลอร์ปฏิบัติงาน

1.4 ชนิดของข้อมูล (Data Types)

1. Integer : ข้อมูลเลขจำนวนเต็ม (int)

2. Float : ข้อมูลเลขจำนวนจริงมีทศนิยม (float)

3. Octal : ข้อมูลชนิดเลขฐาน 8

วิธีการเขียน ต้องนำหน้าด้วยเลขศูนย์ (0)

4. Hexadecimal : ข้อมูลชนิดเลขฐาน 16

วิธีการเขียน ต้องนำหน้าด้วยเลขศูนย์ (0x)

5. Character : ข้อมูลชนิดตัวอักษร 1 ตัว (char)

6. String : ข้อมูลชนิดข้อความ (char ชื่อตัวแปร[n];)

1.4.1 จำนวนเต็ม (Integer)

ตัวเลขที่มีค่าเป็นจำนวนเต็ม ไม่มีจุดทศนิยม ไม่มีเลขชี้กำลัง ขอบเขตของจำนวนเต็ม คือ จำนวนเต็มลบ จำนวนเต็มบวก และจำนวนเต็มศูนย์

- แบ่งเป็น 6 ชนิด คือ

short	: 1 ไบต์ : -128 ... 127
unsigned short	: 1 ไบต์ : 0 ... 255
int	: 2 ไบต์ : -32,768 ... 32,767
unsigned int	: 2 ไบต์ : 0... 65,535
long	: 4 ไบต์ : -2,147,483,648 ... 2,147,483,647
unsigned long	: 4 ไบต์ : 0 ... 4,294,967,296

- ตัวอย่าง (Example)

```
int    book;
long   million;
int    age1, age2, age3;
```

1.4.2 จำนวนจริง (Floating point)

จำนวนที่ต้องมีจุดทศนิยมหรือเลขชี้กำลัง ขอบเขตของจำนวนจริง คือ จำนวนจริงลบ

จำนวนจริงบวก และจำนวนเต็มศูนย์ สามารถเขียนในรูปเลขยกกำลัง (Exponent) ได้

- แบ่งเป็น 3 ประเภท คือ

float	: 6 ไบต์ : $-3.4 \times 10^{-38} \dots 3.4 \times 10^{38}$
double	: 12 ไบต์ : $-3.4 \times 10^{-308} \dots 3.4 \times 10^{308}$
long double	: 24 ไบต์ : $-3.4 \times 10^{-4032} \dots 3.4 \times 10^{4032}$

- ตัวอย่าง (Example)

```
float    tax;
float    grade;
double   area;
float    tax, grade, salary;
```

1.4.3 เลขฐาน 8 (Octal)

ระบบเลขฐานรูปแบบหนึ่งของคอมพิวเตอร์ ประกอบด้วยข้อมูล 0..7 วิธีการเขียนเลขฐาน 8 ของภาษาซีต้องนำหน้าด้วยเลขศูนย์ (0) ตัวอย่างเช่น

เลข 1 ฐานแปด 1_8 วิธีการเขียนในภาษาซี 01 = เลข 1 ฐานสิบ

เลข 7 ฐานแปด 7_8 วิธีการเขียนในภาษาซี 07 = เลข 7 ฐานสิบ

เลข 8 ฐานแปด 10_8 วิธีการเขียนในภาษาซี 010 = เลข 8 ฐานสิบ

1.4.4 เลขฐาน 16 (Hexadecimal)

ระบบเลขฐานรูปแบบหนึ่งของคอมพิวเตอร์ ประกอบด้วยข้อมูล 0..9 และตัวอักษร A..F วิธีการเขียนเลขฐาน 16 ของภาษาซีต้องนำหน้าด้วยเลขศูนย์ (0) และตัว X = 0x ตัวอย่างเช่น

เลข 1 ฐานสิบหก 1_{16} วิธีการเขียนในภาษาซี $0x1$ = เลข 1 ฐานสิบ

เลข 7 ฐานสิบหก 7_{16} วิธีการเขียนในภาษาซี $0x7$ = เลข 7 ฐานสิบ

เลข A ฐานสิบหก A_{16} วิธีการเขียนในภาษาซี $0xA$ = เลข 10 ฐานสิบ

เลข 10 ฐานสิบหก 10_{16} วิธีการเขียนในภาษาซี $0x10$ = เลข 16 ฐานสิบ

1.4.5 อักขระ (Character)

ตัวอักษร 1 ตัวที่อยู่ในเครื่องหมาย ‘...’ (Single quote) สามารถเป็นตัวอักษร ตัวเลข หรือสัญลักษณ์พิเศษต่างๆ แต่ข้อมูลดังกล่าวที่ไม่สามารถนำไปใช้คำนวณได้ ตัวอย่างเช่น ‘A’ ‘2’ ‘*’ ‘%’ ค่าของข้อมูลจะเป็นค่าของรหัสมาตรฐาน ASCII (0-255)

- ตัวอย่าง (Example)

```
char    sex;  
  
char    student_year;  
  
char    value;
```

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

ภาพ 1.5 : ตาราง ascii

ที่มาภาพ 1.5 : <http://www.lookuptables.com>

1.4.6 ข้อความ (String)

กลุ่มของตัวอักษรที่นำมาเขียนเรียงกันอยู่ภายในเครื่องหมาย “.....” (Double quote) ข้อมูลของข้อความไม่สามารถนำไปใช้คำนวณ ตัวอย่างเช่น “Computer Programming”, “418113”, “Computer Science”

- ตัวอย่าง (Example)

```
char    name[10];

char    sur_name[20];

char    subject[ ];

{ไม่กำหนดความยาวของข้อความ}
```

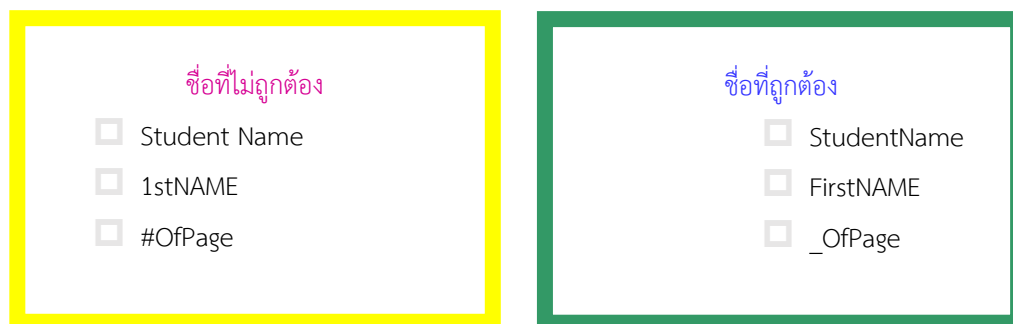
1.5 หลักการตั้งชื่อตัวแปร (Variable)

1.5.1 ชื่อตัวแปร (Variables)

การกำหนดชื่อโปรแกรม ค่าคงที่ ตัวแปร ฟังก์ชัน เพื่อใช้งานในโปรแกรม มีกฎเกณฑ์ดังนี้

- ต้องไม่ใช่คำสงวน
- เริ่มต้นด้วยตัว A..Z หรือ a..z หรือเครื่องหมายขีดเส้นใต้ (_)
- ตัวต่อไปอาจเป็นตัวอักษร (A..Z,a..z) ตัวเลข (0..9) หรือเครื่องหมาย (_) ได้
- การใช้ตัวพิมพ์เล็กหรือพิมพ์ใหญ่จะมีความหมายแตกต่างกัน (Case sensitive) เช่น Name, name, NaMe

- ตัวอย่างการตั้งชื่อ



- ข้อสังเกตในการตั้งชื่อตัวแปร ควรตั้งชื่อให้มีความหมายเข้าใจง่าย เช่น

length	เข้าใจง่ายกว่า	l
price	เข้าใจง่ายกว่า	p
studentname	เข้าใจง่ายกว่า	SN
Area	เข้าใจง่ายกว่า	a,b,c,x,y,z

ควรตั้งชื่อให้เห็นส่วนประกอบต่าง ๆ ได้ง่าย เช่น

student_name	เห็นง่ายกว่า	studentname
part_number	เห็นง่ายกว่า	PARTNUMBER
day_of_week	เห็นง่ายกว่า	dayofweek

1.6 คำสงวน (Reserved Word)

คำที่มีความหมายเฉพาะสำหรับตัวแปลภาษา คอมไพเลอร์จะไม่ยอมให้ใช้ชื่อเหล่านี้ในกรณีอื่น ๆ มีจำนวนทั้งหมด 32 ตัวเช่น int, new, return

C KEYWORDS OR RESERVED WORDS			
auto	break	case	char
const	continue	default	do
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while
double	else	enum	extern
float	for	goto	if

ภาพ 1.6 : คำสงวน

ที่มาภาพ 1.6 : <https://beginnersbook.com/2014/01/c-keywords-reserved-words/>

1.7 การประกาศตัวแปร (Declaration)

ตัวแปร (Variables)

เปรียบเทียบเหมือนกล่องสำหรับเก็บข้อมูล ที่อาจเกิดจากการป้อนเข้าไป หรือเก็บผลจากการคำนวณ ก่อนที่จะเรียกใช้ตัวแปร ต้องประกาศไว้ที่ส่วนประกาศก่อนเสมอ

1.7.1 การประกาศตัวแปร

ใช้กำหนดชื่อตัวแปร และชนิดข้อมูล

รูปแบบ

ชนิดข้อมูล ชื่อตัวแปร;
ชนิดข้อมูล ชื่อตัวแปร₁, ชื่อตัวแปร₂..., ชื่อตัวแปร_n;

หมายเหตุ กรณีประกาศตัวแปรมากกว่า 1 ตัวแปรให้ใช้เครื่องหมาย comma (,) คั่นระหว่างตัวแปร

เช่น

```
int    score ;  
float  height, weight ;
```

1.7.2 การกำหนดค่าตัวแปร

รูปแบบ ชนิดข้อมูล ชื่อตัวแปร = ค่าที่กำหนด ;

เช่น

```
char    subject[8] = "01418113";  
float    score1, score2 = 50;
```

1.8 นิพจน์ (Expression)

กลุ่มของข้อมูล ซึ่งประกอบด้วยตัวถูกกระทำโอเปอเรแลนด์ (operands) และโอเปอเรเตอร์ (Operator) ตัวอย่าง operands เช่น ตัวเลข ค่าคงที่ ตัวแปร ตัวอย่าง operator เช่น -, +, *, and, or ประเภทของนิพจน์แบ่งได้เป็น 2 ประเภท คือ นิพจน์ทางคณิตศาสตร์ (Arithmetic expression) และ นิพจน์ทางตรรกศาสตร์ (Logical expression)

1.8.1. นิพจน์ทางคณิตศาสตร์ (Arithmetic expression)

เป็นนิพจน์ที่ประกอบด้วย Operand ที่เป็นตัวแปร หรือค่าคงที่ที่เป็นตัวเลข เชื่อมกันด้วย Operator ได้แก่ เครื่องหมาย บวก ลบ คูณ หาร

เช่น $(b*b-4*a*c) / (2*a)$ เรียก a, b, c, 4 และ 2 เป็น operand และเรียกสัญลักษณ์ * - และ / ว่า operator

ตัวอย่างการเปลี่ยนนิพจน์ทางพีชคณิตเป็นนิพจน์ภาษาซี

นิพจน์ทางพีชคณิต	นิพจน์ในภาษาซี
5 (number + total)	5* (number + total)
$\frac{(A + B)}{(A - B) - C}$	((A+B) / ((A-B)+C))
X<0 AND Y>1	(X<0) AND (Y>1)

ตัวดำเนินการทางคณิตศาสตร์ (Operators)

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ชนิดข้อมูล
+	บวก	10+15	integer, floating point
-	ลบ	10.-15	integer, floating point
*	คูณ	2*3.0	integer, floating point
/	หาร	Score / 2.0	integer, floating point
% (modular)	หารเอาเศษ	9 mod 5 ผลลัพธ์ = 4	integer

1.8.2 นิพจน์ทางตรรกศาสตร์ (Logical expression)

เป็นนิพจน์ที่เชื่อมกันด้วยโอเปอเรเตอร์ ที่เป็นสัญลักษณ์แสดงการเปรียบเทียบและตรรกศาสตร์ เช่น < , > , = , AND, OR , NOT

เช่น

```
if (num > MAX) numMore = false;
```

นิพจน์ทางตรรกศาสตร์ (num > MAX) มี Operand 2 ตัว คือ num และ MAX และมีเครื่องหมาย > เป็น operator ที่ใช้ในการเปรียบเทียบค่าระหว่าง num และ MAX

ตัวดำเนินการทางการเปรียบเทียบ

ตัวดำเนินการเลขคณิต	ความหมาย	ตัวอย่าง
=	เท่ากับ	Price = 100;
!=	ไม่เท่ากับ	Price != 100;
>	มากกว่า	Price > 10;
<	น้อยกว่า	Price < 10;
>=	มากกว่าเท่ากับ	Price >= 10;
<=	น้อยกว่าเท่ากับ	Price <= 10;

1.8.3 ลำดับการทำงานของตัวโอเปอเรเตอร์

ลำดับความสำคัญ	ตัวดำเนินการ
1	()
2	not
3	++ ,--
4	*, /, mod, and
5	+, -, or, xor
6	=, +=, *=, /=, -=, %=
7	!=, <, >, <=, >=

1.9 ตัวดำเนินการกำหนดค่า

คำสั่ง	คำสั่งเทียบเท่า	คำอธิบาย
<code>a=2;</code>	<code>a=2</code>	นำค่า 2 มาใส่ค่าในตัวแปร a
<code>a+=2;</code>	<code>a=a+2</code>	นำค่า a มา + 2 แล้วเก็บใน a
<code>a-=2;</code>	<code>a=a-2</code>	นำค่า a มา - 2 แล้วเก็บใน a
<code>a*=2;</code>	<code>a=a*2</code>	นำค่า a มา * 2 แล้วเก็บใน a
<code>a/=2;</code>	<code>a=a/2</code>	นำค่า a มา / 2 แล้วเก็บใน a
<code>a%=2;</code>	<code>a=a%2</code>	นำค่า a มา % 2 แล้วเก็บใน a
<code>a&=2;</code>	<code>a=a&2</code>	นำค่า a มา & 2 แล้วเก็บใน a
<code>a =2;</code>	<code>a=a 2</code>	นำค่า a มา 2 แล้วเก็บใน a
<code>a^=2;</code>	<code>a=a^2</code>	นำค่า a มา ^ 2 แล้วเก็บใน a

1.10 ตัวดำเนินการเพิ่มค่าและลดค่า

ภาษาซีมีการใช้ตัวดำเนินการเพิ่มค่า (++) และลดค่า (--) โดยใช้คำสั่งแบบลดรูป ซึ่งการวางตำแหน่งคำสั่ง มี 2 แบบ ได้แก่

- 1) เติมนำ (Prefix) -> $a = ++b$, $a = --b$
- 2) เติมห้ (Postfix) -> $a = b++$, $a = b--$

ตัวอย่างคำสั่งตัวดำเนินการเพิ่มค่าและลดค่า ผลลัพธ์สุดท้ายของคำสั่งตัวดำเนินการเพิ่มค่าและลดค่า ($b=4, d=4$)

คำสั่ง	คำสั่งเทียบเท่า	ค่าสุดท้าย	สรุปคำสั่ง
$a = b++;$	$a = b;$	$a = 4$	Postfix
	$b = b+1;$	$b = 5$	
$a = b--;$	$a = b;$	$a = 4$	Postfix
	$b = b-1;$	$b = 3$	
$c = ++d;$	$d = d+1;$	$c = 5$	Prefix
	$c = d;$	$d = 5$	
$c = --d;$	$d = d-1;$	$c = 3$	Prefix
	$c = d;$	$d = 3$	

คำถามท้ายบทที่ 1

1. จงอธิบายข้อดีและข้อเสียของภาษาซี
2. จงอธิบายวิธีการทำงานของตัวแปลภาษาแบบคอมไพเลอร์ (Compiler) ในการแปลง Source Code ไปเป็นภาษาเครื่องจงอธิบายหน้าที่ของส่วนก่อนประมวลผล (Preprocessor directive) ของภาษาซี พร้อมยกตัวอย่างประกอบ
3. ภาษาซีใช้เครื่องหมายใดเพื่อใช้เป็นส่วนคำอธิบาย (Comments) คำสั่งหรือโปรแกรมของแบบ 1 บรรทัดและหลายบรรทัด
4. จงอธิบายความแตกต่างระหว่างชนิดข้อมูลอักขระ (Character) และข้อความ (String)
5. จงอธิบายวิธีการแสดงผลค่าของจำนวนเต็มเป็นเลขฐานสิบ ฐานแปด และฐานสิบหก ของภาษาซี
6. จงอธิบายความหมายของคำสงวน (Reserved Word)
7. จงอธิบายความหมายของนิพจน์ทางตรรกศาสตร์ (Logical expression) พร้อมยกตัวอย่างประกอบ
8. จงตั้งชื่อตัวแปรพร้อมทั้งกำหนดชนิดของข้อมูล (Data Types) ให้กับตัวแปรดังต่อไปนี้

ตัวแปร	ชื่อตัวแปร	ชนิดของข้อมูล
1) ค่าหอพัก		
2) ชื่อนิสิต		
3) ความสูงของนิสิต		
4) อายุของนิสิต		
5) เกรดเฉลี่ย		
6) รหัสวิชาเรียน		
7) ชื่อวิชาเรียน		
8) ปีการศึกษา		
9) หน่วยการใช้ไฟฟ้าต่อเดือน		
10) จำนวนหนังสือในห้องสมุด		

9. จงหาผลลัพธ์จากนิพจน์ดังต่อไปนี้

นิพจน์	ผลลัพธ์
1) $9 \% (12 - 5)$	
2) $2 + 4 * 5$	

นิพจน์	ผลลัพธ์
3) $5 \% 3 * 2$	
4) $1 / 2 * 3$	
5) $7 + 3 * (2 - 5) \% 4$	
6) $17.5 / 5 + 2 * 3$	
7) $12 \% 3 + 3 * 2$	
8) $15 / (2*3) \% 6$	
9) $1+2+3*5+1*4$	
10) $(15 - 8 + 2) + (30 \% 8 / 5)$	

10. จงเขียนคำสั่งเทียบเท่า ค่าสุดท้าย พร้อมทั้งสรุปคำสั่งของคำสั่งตัวดำเนินการเพิ่มค่าและลดค่า

เมื่อกำหนดให้ $a = 4$ และ $c = 5$

คำสั่ง	คำสั่งเทียบเท่า	ค่าสุดท้าย	สรุปคำสั่ง
$b = a++;$		$a =$	
		$b =$	
$b = a--;$		$a =$	
		$b =$	
$d = ++c;$		$c =$	
		$d =$	
$d = --c;$		$c =$	
		$d =$	