

บทที่ 2 ภาษาซี การเขียนโปรแกรมเบื้องต้น และคำสั่งแสดงผล

ศท211 วิทยาการคอมพิวเตอร์ 1

โดย อาจารย์ภาณุวัฒน์ เมฆะ

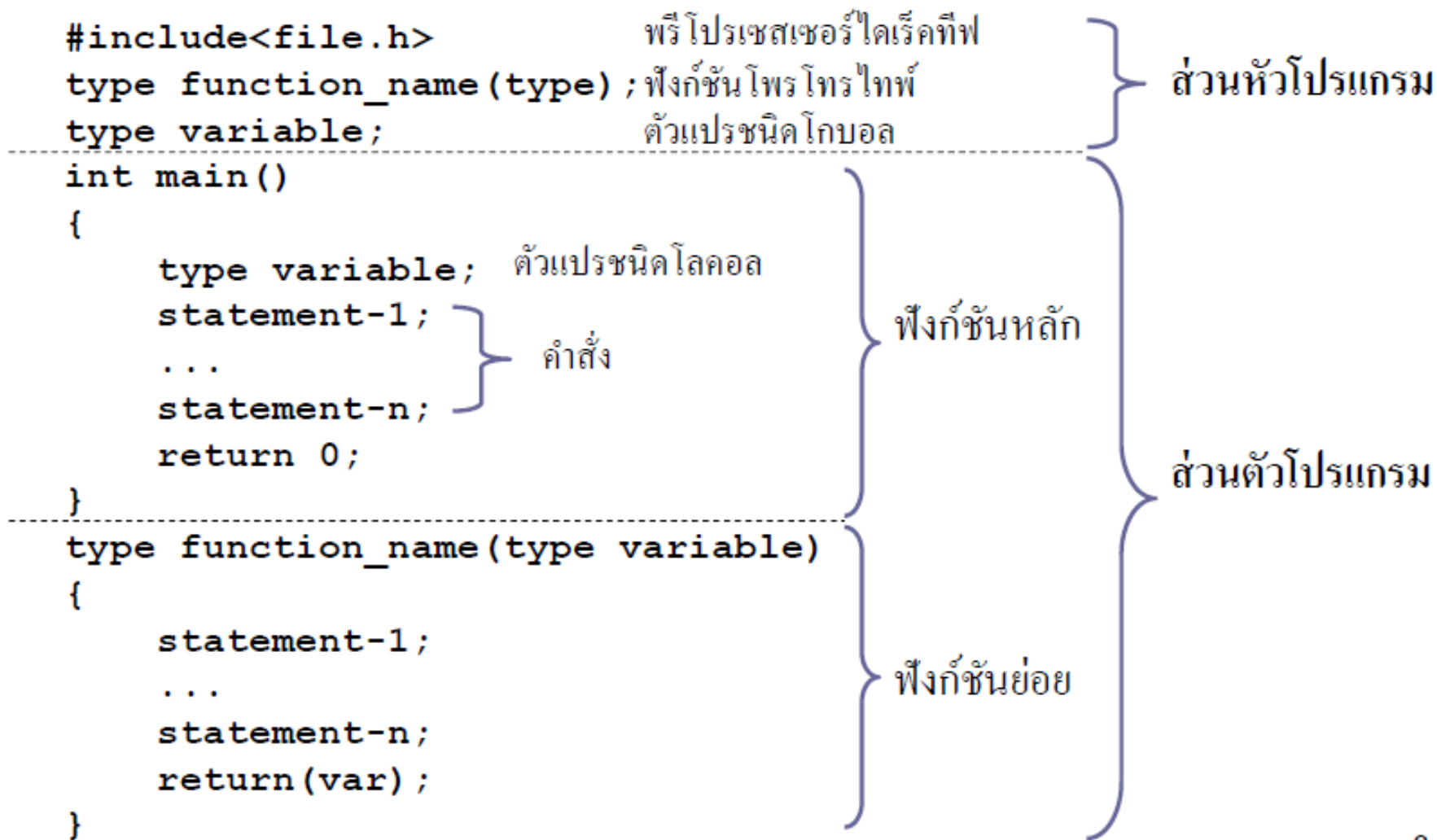
สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้

วัตถุประสงค์

- สามารถเขียน โปรแกรมภาษาซีอย่างง่าย ๆ ได้
- สามารถแสดงข้อความออกทางจอภาพได้
- สามารถแสดงข้อมูลชนิดต่างๆ ออกทางจอภาพได้
- สามารถกำหนดรูปแบบการแสดงผลออกทางจอภาพได้

โครงสร้างโปรแกรมภาษาซี

ภพ211 วิทยาการคอมพิวเตอร์1



โครงสร้างโปรแกรมภาษาซี

- ส่วนหัวโปรแกรม (Header Program)
 - พรีโปรเซสเซอร์ไดเรกทีฟ (Preprocessor Directive)
 - `#include`
 - `#define`
 - ฟังก์ชันโปรโตไทป์ (Function Prototype Declaration)
 - ตัวแปรชนิดโอบอล (Global Variable Declaration)

2.1 โครงสร้างโปรแกรมภาษาซี

- ส่วนตัวโปรแกรม (Body Program)
 - ฟังก์ชันหลัก (Main Function)
 - ตัวแปรชนิดโลคอล (Local Variable Declaration)
 - คำสั่ง (Statement)
 - ฟังก์ชันอื่น ๆ (Functions)
- คำอธิบายโปรแกรม (Comment Program)

2.2 프리프로세서เซอร์โค레킵

- ส่วนประมวลผลก่อน เป็นส่วนที่สั่งให้คอมไพเลอร์เตรียมการทำงานที่กำหนดไว้ ก่อนที่จะทำงานในฟังก์ชันหลัก

```
#include<file-name.h>
```

ให้โปรแกรมดึงคำสั่งจาก *file-name.h* ใน Include Directory

```
#include"file-name.h"
```

ให้โปรแกรมดึงคำสั่งจาก *file-name.h* ใน Current Directory

file-name คือชื่อไฟล์นามสกุล h

ตัวอย่างคำสั่งฟรีโปรเซสเซอร์ไคเร็คทีฟ

```
#include<stdio.h>
```

ทำการเรียกไฟล์ `stdio.h` เพื่อให้สามารถใช้คำสั่งทั่วไปได้

```
#include<conio.h>
```

ทำการเรียกไฟล์ `conio.h` เพื่อให้สามารถใช้คำสั่งจัดการหน้าจอ และคำสั่งรับและแสดงผลเพิ่มเติม

```
#include<math.h>
```

ทำการเรียกไฟล์ `math.h` เพื่อให้สามารถใช้คำสั่งเกี่ยวกับคณิตศาสตร์ได้

```
#include<string.h>
```

ทำการเรียกไฟล์ `string.h` เพื่อให้สามารถใช้คำสั่งจัดการเกี่ยวกับข้อความได้

ฟังก์ชันหลัก

- ฟังก์ชันหลักเป็นฟังก์ชันที่โปรแกรมภาษาซีต้องมีอยู่เสมอ เพราะคอมไพเลอร์ของภาษาซีจะเริ่มต้นการทำงานจากฟังก์ชันหลัก
- ตัวอย่างการเขียนฟังก์ชันหลักที่ไม่มีการทำงานใดๆ

```
int main(void)
{
    return (0) ;
}
```

```
int main()
{
    return 0 ;
}
```


กฎเกณฑ์การใช้คำสั่งในภาษาซี

- ใช้เครื่องหมาย semi colon ; เป็นจุดสิ้นสุดคำสั่ง
- ใช้อักขรตัวเล็กสำหรับเรียกใช้คำสั่ง (statement)
- ใช้เครื่องหมาย comma , สำหรับคั่นตัวแปร และพารามิเตอร์
- หากคำสั่งใดมีคำสั่งส่วนย่อยภายในหลาย ๆ คำสั่ง ให้ใช้เครื่องหมายปีกกา { } สำหรับกำหนดขอบเขต

2.3 การแสดงผลด้วยคำสั่ง **printf**

- การเขียนโปรแกรมจำเป็นต้องมีการแสดงผล เพื่อให้ผู้ใช้งานทราบว่าโปรแกรมสามารถทำงานอะไร จำเป็นต้องป้อนค่าอะไรบ้าง และเมื่อโปรแกรมทำงานเสร็จ ผลลัพธ์ที่ได้เป็นอย่างไร
- คำสั่งที่ใช้สำหรับการแสดงผลในภาษาซีมีหลายคำสั่ง แต่ที่สามารถใช้งานได้ครอบคลุมและนิยมใช้กัน คือ คำสั่ง `printf`

รูปแบบของคำสั่ง `printf`

```
printf (format-string, data-list);
```

- `format-string` คือรูปแบบของข้อความซึ่งจะประกอบด้วยข้อความธรรมดา, คำรหัส ASCII และ ส่วนแสดงชนิดข้อมูล โดย `format-string` จะอยู่ในเครื่องหมาย Double quote " "
- `data-list` คือชื่อข้อมูล หรือตัวแปรที่จะทำการแสดงผลตามส่วนแสดงชนิดข้อมูลใน `format-string`

หมายเหตุ คำสั่ง `printf` ต้องเรียกใช้ Preprocessor Directive `#include<stdio.h>`

ตัวอย่างการใช้งานคำสั่ง `printf` ที่ไม่ถูกต้อง

```
1 int main()  
2 {  
3     printf("hello world!");  
4     return 0;  
5 }  
6  
7
```

```
1>Compiling...
```

```
1>helloworld.cpp
```

```
1>d:\temp\test\test\helloworld.cpp(3) : error C3861: 'printf': identifier not found
```


```
1>Build log was saved at "file:///d:/temp/test/test/Debug/BuildLog.htm"
```

```
1>Test - 1 error(s), 0 warning(s)
```

```
=====  
Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped =====
```



ตัวอย่างการใช้งานคำสั่ง `printf` ที่ถูกต้อง



```
#include<stdio.h>
int main()
{
    printf ("Hello World!");
    return 0;
}
```

```
Hello World!
```

รูปแบบของข้อความในคำสั่ง printf

- ข้อความธรรมดา เป็นส่วนที่แสดงตัวอักษร ตัวเลขออกโดยตรง
- คำรหัส ASCII เป็นส่วนที่ควบคุมรูปแบบการแสดงผล เช่น การจัดย่อหน้า การขึ้นต้นบรรทัดใหม่ เป็นต้น
- ส่วนแสดงชนิดข้อมูล เป็นการกำหนดรูปแบบของชนิดข้อมูลที่จะแสดงผลข้อมูล หรือตัวแปร
 - ส่วนแสดงชนิดข้อมูลแบบปกติ
 - ส่วนแสดงชนิดข้อมูลที่กำหนดรูปแบบการแสดงผล

ข้อความธรรมดา

- การแสดงข้อความธรรมดาเป็นการใช้คำสั่ง printf ให้แสดงข้อความที่ต้องการออกทางจอภาพ โดยจะอยู่ในส่วนของ format-string ในคำสั่ง (อยู่ในเครื่องหมาย Double quote " ")

```
printf (format-string, data-list);
```

- เช่น คำสั่งในตัวอย่างโปรแกรมที่ผ่านมา

```
printf ("Hello World!");
```

จะทำการแสดงข้อความ Hello World! ออกทางจอภาพหลังจาก
สั่ง Run โปรแกรม

ตัวอย่างการใช้งานคำสั่ง `printf` หลายๆคำสั่ง

```
#include<stdio.h>
#include<conio.h>
int main()
{
    printf ("Hello World!");
    printf ("This is my first Program.");
    printf ("I am a programmer.");
    return 0;
}
```


ค่ารหัส ASCII

- ค่ารหัส ASCII เป็นอักขระพิเศษที่ใช้งานในภาษาซี ซึ่งใช้ควบคุมการแสดงผลในคำสั่ง printf ในส่วนของ format-string

ค่ารหัส ASCII	การใช้งาน
<code>\t</code>	เว้นช่องว่างทุก 1 แท็บ (8 ช่องตัวอักษร)
<code>\n</code>	ขึ้นบรรทัดใหม่
<code>\0</code>	เป็นอักขระว่าง
<code>\'</code>	แสดง single quote ออกทางจอภาพ
<code>\"</code>	แสดง double quote ออกทางจอภาพ
<code>\\</code>	แสดง backslash ออกทางจอภาพ

ตัวอย่างการใช้งานคำสั่ง `printf` โดยใช้ค่ารหัส ASCII

```
#include<stdio.h>
int main()
{
    printf ("Hello World!\n");
    printf ("This is my first Program.");
    printf ("\n\tI\'m a programmer.");
    return 0;
}
```

ส่วนแสดงชนิดข้อมูล

- ในการเขียนโปรแกรมที่มีการประยุกต์ขึ้น เช่น โปรแกรมคำนวณเลข จะต้องมีการแสดงผลลัพธ์ของค่าที่ทำการคำนวณ ซึ่งคำสั่ง `printf` สามารถแสดงผลค่าตัวแปรได้ แต่จะต้องเขียนโปรแกรมให้มีความสัมพันธ์กันในส่วนของ `format-string` และ `data-list`

```
printf (format-string, data-list);
```

- โดยที่ใน `format-string` จะมีส่วนแสดงชนิดข้อมูล
- และใน `data-list` จะมีข้อมูล หรือตัวแปรที่จะแสดงผล

การกำหนดส่วนแสดงชนิดข้อมูลแบบต่างๆ

ส่วนแสดงชนิดข้อมูล	การใช้งาน
%d	แสดงผลข้อมูลชนิดจำนวนเต็ม
%u	แสดงผลข้อมูลชนิดจำนวนเต็มบวก (ไม่คิดเครื่องหมาย)
%o	แสดงผลข้อมูลเป็นเลขฐานแปด
%x	แสดงผลข้อมูลเป็นเลขฐานสิบหก
%f	แสดงผลข้อมูลชนิดจำนวนทศนิยม (6 ตำแหน่ง)
%e	แสดงผลข้อมูลเป็นจำนวนทศนิยมและอยู่ในรูปยกกำลัง
%c	แสดงผลข้อมูลชนิดอักขระ
%s	แสดงผลข้อมูลชนิดข้อความ
%p	แสดงผลข้อมูลชนิดตัวชี้ตำแหน่ง

ตัวอย่างการใช้งานคำสั่ง `printf` แสดงผลข้อมูล

```
#include<stdio.h>
int main()
{
    printf ("My name is : %s\n", "Kmitl");
    printf ("My point : %d\n", 10+40+49);
    printf ("Grade : %c\n", 'A');
    printf ("GPA : %f", 3.99);
    return 0;
}
```


การกำหนดรายชื่อตัวแปรในคำสั่ง **printf**

- การแสดงผลตัวแปร โดยใช้คำสั่ง `printf` สามารถใช้งานโดยการใส่ข้อมูล หรือรายชื่อตัวแปรในส่วนของ `data-list`
- หากไม่ต้องการแสดงผลตัวแปร ไม่ต้องมีส่วนของ `data-list`
- หากต้องการแสดงผลข้อมูล หรือตัวแปรมากกว่าหนึ่งตัวในคำสั่งให้ใช้ เครื่องหมาย `comma` , สำหรับคั่นรายชื่อตัวแปร โดยจะทำการแสดงผลตามลำดับตัวแปร และส่วนการแสดงชนิดข้อมูล

รายชื่อตัวแปรในคำสั่ง printf

ภพ211 วิทยาการคอมพิวเตอร์1

```
printf ("...%?... ", data);
```



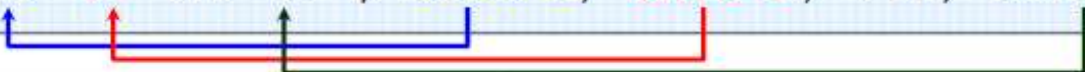
```
...data...
```

ตัวอย่าง

```
printf ("...%d...", 19);
```

```
...19...
```

```
printf ("%? %? ... %?", data-1, data-2, ..., data-n);
```



```
data-1 data-2 ... data-n
```

ตัวอย่าง

```
printf ("%d-%d-%d", 19, 1, 1980);
```

```
19-1-1980
```

ตัวอย่างการใช้งานคำสั่ง **printf** แสดงผลข้อมูลหลายตัว-1

```
#include<stdio.h>
int main()
{
    printf ("Age = %d, GPA = %f\n",17,3.75);
    printf ("Programming: %f\nMechanics: %f",4.0,3.5);
    return 0;
}
```


ตัวอย่างการใช้งานคำสั่ง **printf** แสดงผลข้อมูลหลายตัว-2

```
#include<stdio.h>
int main()
{
    printf ("Subject : %s(%d) \n", "Programming", 2552);
    printf ("Point : %d\nGrade : %c", 99, 'A');
    return 0;
}
```

ส่วนแสดงชนิดข้อมูลที่กำหนดรูปแบบการแสดงผล

```
printf ("%m?", data);
```

- เป็นการจองพื้นที่หน้าจอบ้างจำนวน m ตัวอักษร แล้วแสดงผล data ชิดด้านขวาของพื้นที่ส่วนที่จองไว้ โดยชนิดข้อมูลตาม ? (หากความยาวเกินส่วนที่จองไว้ก็จะเลื่อนออกไป)

```
printf ("% -m?", data);
```

- เป็นการจองพื้นที่หน้าจอขนาด m ตัวอักษร แล้วแสดงผล data ชิดด้านซ้ายของพื้นที่ส่วนที่จองไว้ โดยชนิดข้อมูลตาม ?

ตัวอย่างการใช้งานคำสั่ง `printf` โดยมีการกำหนดรูปแบบ-1

```
#include<stdio.h>
int main()
{
    printf ("123456789012345678901234567890");
    printf ("\n%20d*",46);
    printf ("\n%-20d*",46);
    printf ("\n%3d*",46);
    printf ("\n%3d*",2550);
    return 0;
}
```

```
123456789012345678901234567890
                                46*
46                                *
  46*
2550*
```


2.4 ส่วนแสดงชนิดข้อมูลที่กำหนดรูปแบบการแสดงผล

```
printf ("% .n?", data) ;
```

- เป็นการกำหนดให้แสดงจำนวนทศนิยม n ตำแหน่ง สำหรับ %f หรือแสดงอักขระจำนวน n ตัว สำหรับ %s

```
printf ("%m.n?", data) ;
```

- เป็นการจองพื้นที่หน้าจอขนาด m ตัวอักษรแล้วแสดงผล data จำนวนทศนิยม n ตำแหน่ง สำหรับ %f หรือแสดงอักขระจำนวน n ตัว สำหรับ %s

ตัวอย่างการใช้งานคำสั่ง `printf` โดยมีการกำหนดรูปแบบ-3

```
printf ("123456789012345678901234567890");  
printf ("\n%20s*", "programming");  
printf ("\n%-20s*", "programming");  
printf ("\n%.3s*", "programming");  
printf ("\n%20.3s*", "programming");  
printf ("\n%-20.3s*", "programming");
```

```
123456789012345678901234567890
```

```
    programming*
```

```
programming      *
```

```
pro*
```

```
                pro*
```

```
pro                *_
```

ตัวอย่างการใช้งานคำสั่ง **printf** โดยมีการกำหนดรูปแบบ-4

```
printf ("123456789012345678901234567890");  
printf ("\n%20f*", 1234.56789);  
printf ("\n%-20f*", 1234.56789);  
printf ("\n%.3f*", 1234.56789);  
printf ("\n%20.3f*", 1234.56789);  
printf ("\n%-20.3f*", 1234.56789);
```

```
123456789012345678901234567890  
      1234.567890*  
1234.567890      *  
1234.568*  
      1234.568*  
1234.568      *  
      _
```

2.5 คำอธิบายโปรแกรม

- คำอธิบายโปรแกรมเป็นส่วนที่เพิ่มในโปรแกรมเพื่อช่วยให้ผู้ที่เขียน หรือผู้ที่อ่านโปรแกรมสามารถเข้าใจกับตัวโปรแกรมได้ง่ายขึ้น
- โดยที่คำอธิบายโปรแกรมนี้อาจจะไม่มีผลต่อการทำงานของโปรแกรม คือเวลาที่ทำการคอมไพล์ จะข้ามข้อความที่อยู่ในส่วนอธิบายโปรแกรมไป

รูปแบบคำอธิบายโปรแกรม

```
/* Comment Sentence 1  
   Comment Sentence 2  
   ...  
   Comment Sentence n */
```

```
/* Comment Sentence */
```

```
// Comment Sentence
```

โจทย์ : โปรแกรมแสดงผลข้อมูลส่วนตัว

- จงเขียนโปรแกรมแสดงผล ชื่อ-นามสกุล อายุ
– ตัวอย่างการแสดงผล

```
Name      : Kmitl  
Surname   : Engineering  
Age       : 46
```

โจทย์ : โปรแกรมรายได้

- จงเขียนโปรแกรมแสดงรายได้ต่อเดือน แล้วแสดงว่าต่อสัปดาห์ คิดเป็นเท่าใด (ทศนิยม 2 ตำแหน่ง) ต่อวันคิดเป็นเท่าใด (ทศนิยม 3 ตำแหน่ง)
 - ตัวอย่างการแสดงผล

Salary	:	6500
Money/Week	:	1625.00
Money/Day	:	232.143

โจทย์ : โปรแกรมแสดงเกรด

- จงเขียนโปรแกรมแสดงผล เกรดวิชา Programming, Drawing, Mechanics, Math1 พร้อมแสดงเกรดเฉลี่ย (ทศนิยม 2 ตำแหน่ง)
– ตัวอย่างการแสดงผล

Programming	:	A
Drawing	:	B+
Mechanics	:	B+
Math	:	B
GPS	:	3.50

คำถามท้ายบท

1. แสดงผลของโปรแกรมต่อไปนี้

```
#include<stdio.h>
int main()
{
    /* float  radius, pi, area;
    pi = 22/7; // pi = 3.14;
    printf ("Enter Radius of Circular : ");
    scanf ("%f",&radius);
    area = pi * radius * radius; */
    printf ("Programming Language",2008);
    return 0;
}
```

คำถามท้ายบท (ต่อ)

2. จงเขียนส่วนของโปรแกรมให้แสดงผลดังต่อไปนี้

12\06\2008

King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand.

3. จงเขียนส่วนของโปรแกรมให้แสดงผลตามเงื่อนไขต่อไปนี้

บรรทัดที่ 1 แสดง รหัสประจำตัว นศ.

บรรทัดที่ 2 แสดง ชื่อ นามสกุล

บรรทัดที่ 3 แสดง วัน/เดือน/ปี เกิด

บรรทัดที่ 8 แสดงข้อความ <http://www.ce.kmitl.ac.th>

ซิดขอบขวาของบรรทัด

บรรทัดที่ 10 แสดงข้อความ **bye bye** ตรงกลางบรรทัด

ซินจ่าว หมายถึง สวัสดีในภาษาเวียดนาม

