บทที่ 3

พื้นฐานภาษาซี

โครงสร้างการเขียนโปรแกรมภาษาซี

ชื่อ Library ที่ต้องการใช้
(Preprocessor Directive)

จุดเริ่มต้นของโปรแกรม
(Start ใน Flowchart)

ชุดคำสั่ง (แต่ละขั้นตอนใน Algorithm)

จุดสิ้นสุดของโปรแกรม
(End ใน Flowchart)

```c
#include <stdio.h>

int main()
{
    char fullname[50];
    printf("Hello %s", fullname);
    return 0;
}
```

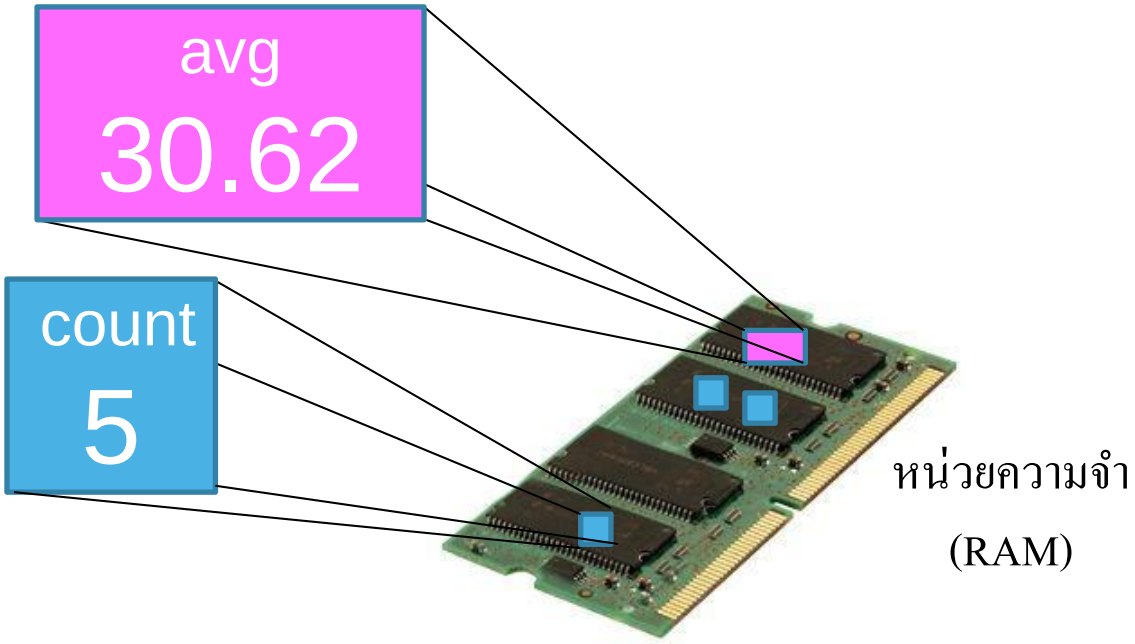
โครงสร้างการเขียนโปรแกรมภาษาซี

โครงสร้างภาษาซีสามารถเขียนในแบบใดแบบหนึ่งก็ได้

```c
#include <stdio.h>

int main()
{
    printf("Hello world!\n");
    return 0;
}
```

```c
#include <stdio.h>

void main()
{
    printf("Hello world!\n");

}
```

ตัวแปร (Variables)



ชนิดของตัวแปร (Data Type)

ตัวอักขระ

Character

เลขจำนวนเต็ม

Integer

เลขจำนวนจริง

Floating-point

เลขจำนวนจริง 2 เท่า

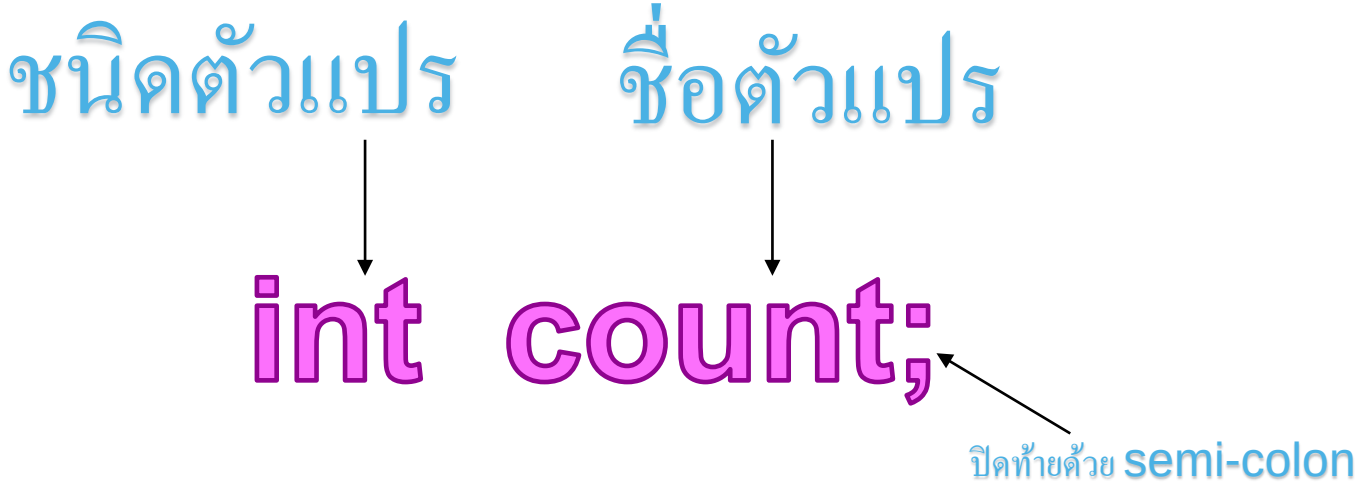
Double

ชนิดของตัวแปร (Data Type)

ชนิดข้อมูล	ชื่อเต็ม	ความหมาย	ขนาด (Byte)	ช่วงข้อมูล
char	character	ตัวอักขระ	1	-128 ถึง 127
int	integer	เลขจำนวนเต็ม รวมเครื่องหมาย	2	-32,768 ถึง 32,767
unsigned int	unsigned integer	เลขจำนวนเต็ม ไม่รวมเครื่องหมาย + หรือ -	2	0 ถึง 65,535
long int	long integer	เลขจำนวนเต็มแบบยาว รวมเครื่องหมาย	4	-2,147,483,648 ถึง 2,147,483,647
float	floating-point	เลขจำนวนจริง	4	3.4×10^{-38} ถึง 3.4×10^{38}
double		เลขจำนวนจริง 2 เท่า	8	3.4×10^{-308} ถึง 3.4×10^{308}

การประกาศตัวแปร

การประกาศตัวแปร (Variable Declaration) คือ การจองพื้นที่บน RAM เพื่อใช้ในการพักข้อมูล มีรูปแบบดังนี้



การเขียนโค้ดโปรแกรมด้วยภาษาซีจะต้องประกาศตัวแปรก่อนเสมอ จึงจะนำข้อมูลไปเก็บได้ ต่างกับ RAPTOR ที่สามารถใช้ตัวแปรใหม่ได้เลย โดยไม่ต้องประกาศ

ตัวอย่าง

ประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ count

ประกาศตัวแปรชนิดอักขระ ชื่อ grade

```
#include <stdio.h>

void main() {

    int count;
    char grade;

}
```

ตัวอย่าง

```c
#include <stdio.h>

void main()
{
    int count;
    int num;
    int i;
}
```

ประกาศตัวแปรชนิดจำนวนเต็ม 3 ตัวแปร

```c
#include <stdio.h>

void main()
{
    int count, num, i;

}
```

ตัวแปรชนิด String

- String คือ ตัวแปรที่เก็บรายการของอักขระตั้งแต่ 1 ตัวขึ้นไป ซึ่งมักจะอยู่ในรูปแบบคำหรือข้อความ
- รูปแบบการประกาศตัวแปรที่เป็น String

```
char  variableName[n];
```

  - n คือ ความจุของอักขระที่เก็บได้

ตัวอย่าง

ประกาศตัวแปรชนิด String ชื่อ studentName

ประกาศตัวแปรชนิด String ชื่อ message

```
#include <stdio.h>

void main()
{
    char  studentName[15];
    char  message[100];
}
```

ชื่อตัวแปร

- ขึ้นต้นด้วยตัวอักษร a-z, A-Z หรือ _ เท่านั้น
- ประกอบด้วยตัวอักษร a-z, A-Z, 0-9 หรือ _ เท่านั้น ห้ามมีสัญลักษณ์ใดๆ
- เป็น Case-sensitive เช่น count ต่างกับ Count
- ไม่ตรงกับคำสั่งในภาษา C หรือ Reserved word

Reserved word

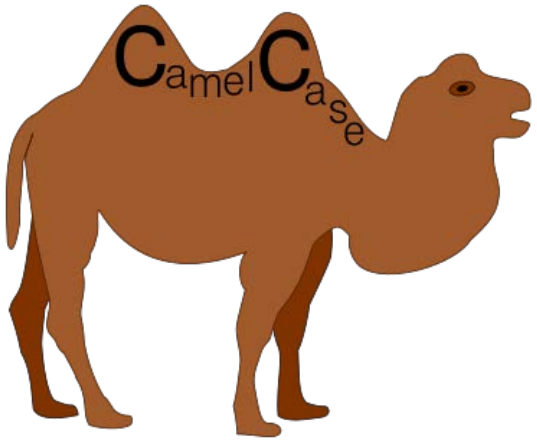
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

ชื่อตัวแปรที่ดี

ชื่อตัวแปร สำหรับเก็บนามสกุลพนักงาน	
employeeLastName	ดี
empLastName	ดี - emp ช่วยย่อ employee ให้สั้นลง
emlstnam	ถูกกฎ แต่คลุมเครือ
lastNameOfTheEmployeeInQuestion	ถูกกฎ แต่ยาวเกินไป
employeelastname	ถูกกฎ แต่อ่านยาก

รูปแบบชื่อตัวแปร

- **Camel case** คือ การตั้งชื่อตัวแปรที่มีคำ 2 คำขึ้นไป (Compound Words) โดยเริ่มต้นแต่ละคำด้วยตัวพิมพ์ใหญ่ เช่น employeeLastName, productNumber



- **Snake case** คือ การตั้งชื่อตัวแปรที่มีคำ 2 คำขึ้นไป โดยคั่นแต่ละคำด้วยเครื่องหมาย _ เช่น employee_last_name, product_number

กิจกรรม

- ชื่อตัวแปรต่อไปนี้ ถูก หรือ ผิด

1) semester grade

2) fall2005_grade

3) GradeInCS2013

4) MY_GRADE

5) return

6) _number1

7) 911site

8) *var

9) int-int

10) var001

11) string.1

12) str91var

13) TaxRate

การกำหนดค่าให้ตัวแปร

- ใช้เครื่องหมาย เท่ากับ (=) ในการกำหนดค่าใดๆ ให้กับตัวแปร
- กำหนดตัวเลขให้กับตัวแปร

```
count = 12;
```

- กำหนดอักขระให้กับตัวแปร

```
grade = 'B';
```

ไม่ใช้ " กำหนด เช่น grade = "B";

- กำหนดให้ผลการคำนวณเก็บลงตัวแปร

```
avg = sum/30;
```

ตัวอย่าง

กำหนดค่า 12 ให้กับตัวแปร count

กำหนดอักขระ F ให้กับตัวแปร grade

นำค่าที่เก็บใน count มาบวก 2 แล้วนำผลที่ได้ไปเก็บทับในตัวแปร count

```
#include <stdio.h>

void main() {
    int count;
    char grade;
    count = 12;
    grade = 'F';
    count = count + 2;
}
```

การเกิด Overflow Value

- หากค่าที่กำหนดให้ตัวแปรมีความจุไม่เพียงพอจะเกิด **"Overflow Value"**
- เกิดจากการกำหนดค่าไม่ตรงกับชนิดของตัวแปรที่ประกาศ
- ควรพิจารณาทั้งสองฝั่งของเครื่องหมาย = ว่าเป็นชนิดเดียวกันหรือไม่

char grade;

grade = 35.559;

กิจกรรม

- การกำหนดค่าต่อไปนี้ถูกต้องหรือไม่ กำหนดให้ เมื่อประกาศตัวแปรดังนี้

```
char grade;
int quizScore, homeworkScore;
```

1) grade = quizScore;
2) homeworkScore = quizScore;
3) homeworkScore = "9";
4) quizScore = homeworkScore + 25;
5) 100 = homeworkScore;
6) grade = 4;
7) homeworkScore + 1 = quizScore;

การกำหนดค่าเริ่มต้นให้ตัวแปร

การกำหนดค่าเริ่มต้นให้ตัวแปร (Variable Initialization) คือ การประกาศตัวแปรพร้อมกับการกำหนดค่าแรกไว้ในตัวแปร

กำหนดค่าเริ่มต้นให้กับตัวแปร count

กำหนดค่าเริ่มต้นให้กับตัวแปร grade

กำหนดค่าเริ่มต้นให้กับตัวแปร fullname

```
#include <stdio.h>

void main() {
   int count = 0;
   char grade = 'A';
   char fullname[10] = "John";
    count = count + 2;
}
```

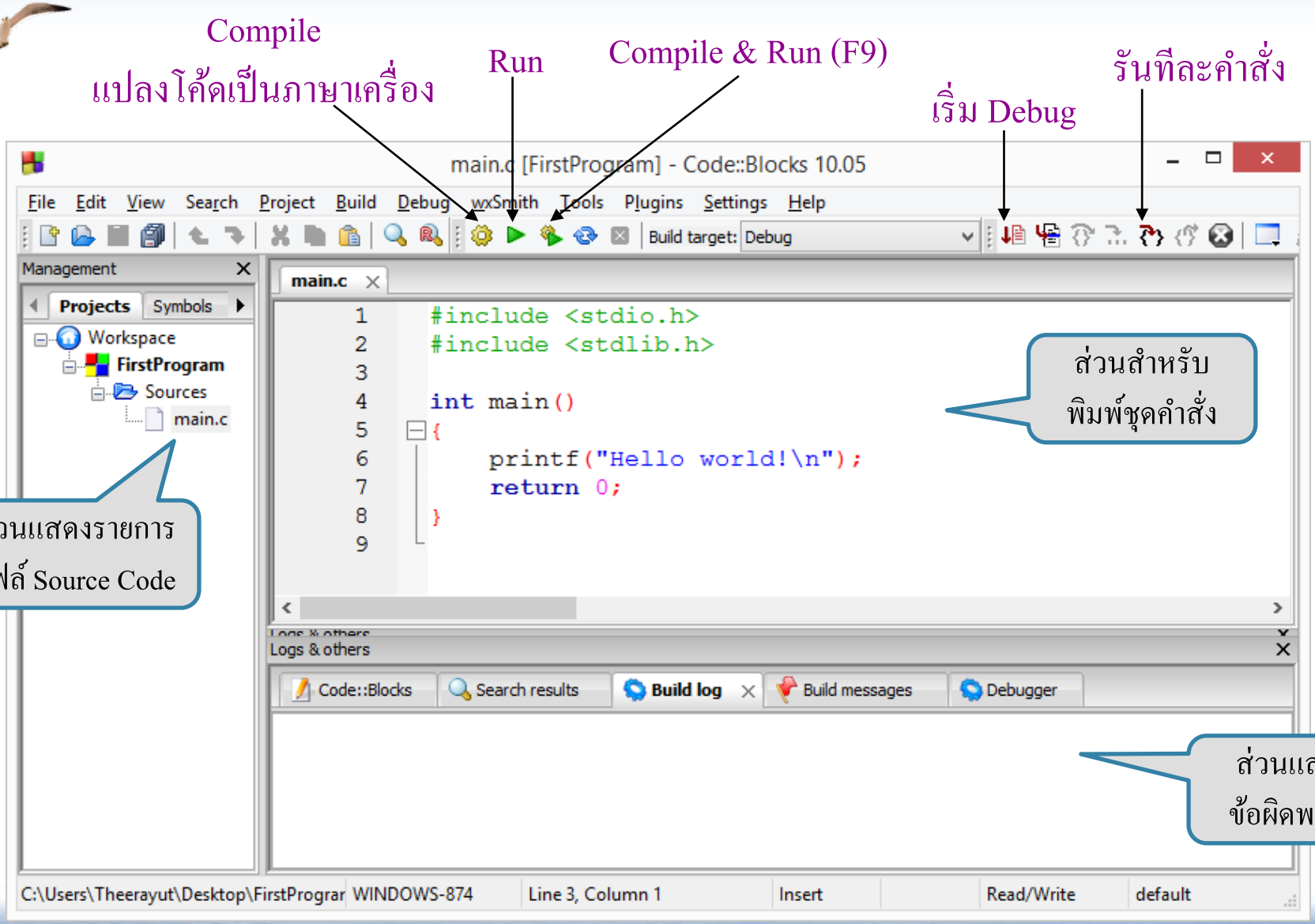
เครื่องมือในการพัฒนาโปรแกรม

- Editor
  - NotePad
  - NotePad++
  - Atom
  - Sublime
- Compiler และ Debugger
  - GCC
  - Turbo C

IDE

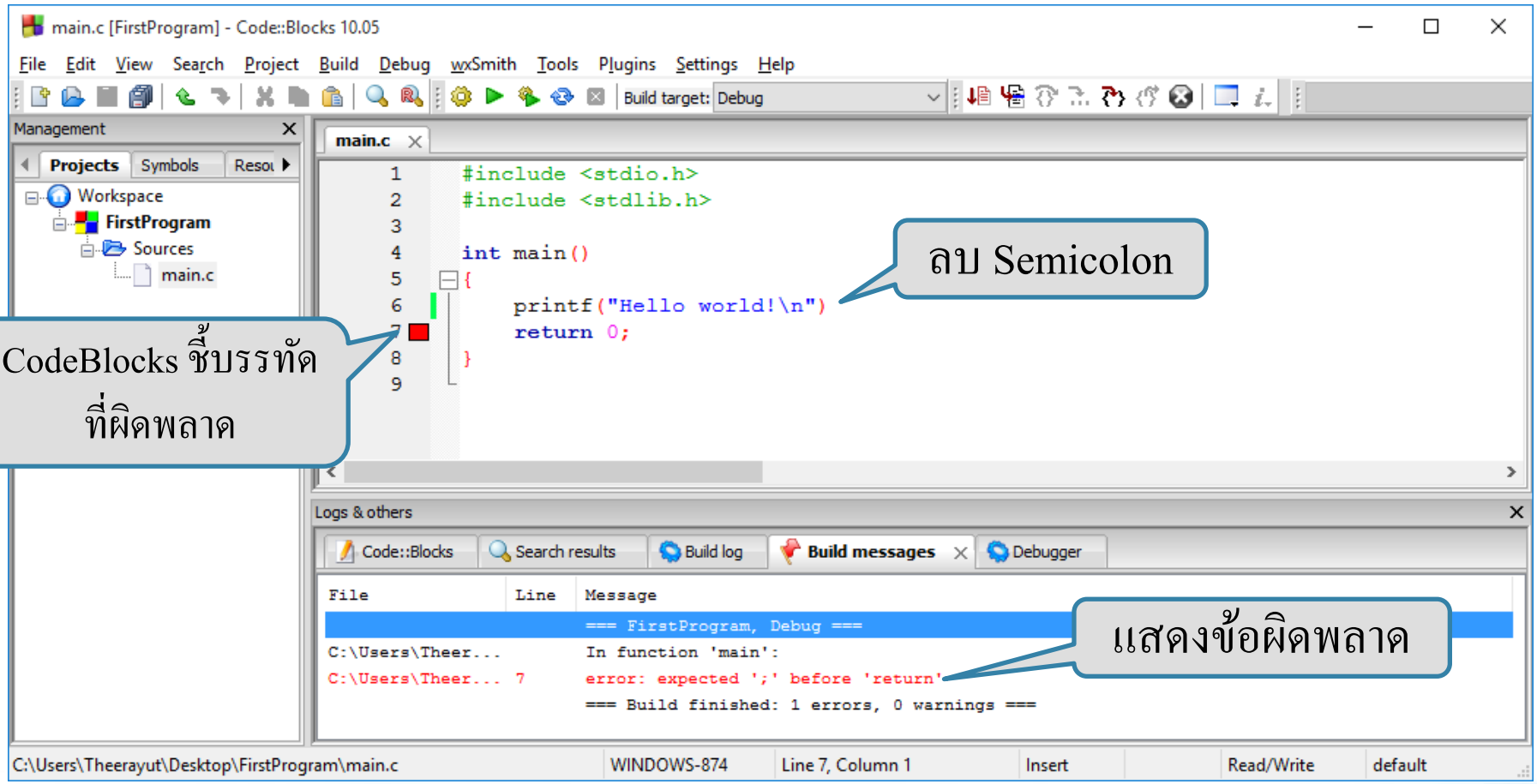
- IDE ย่อมาจาก Integrated Development Environment
- IDE คือ เครื่องมือที่ช่วยในการพัฒนาโปรแกรมโดยรวบรวมเอาเครื่องมือที่จำเป็นในการพัฒนาโปรแกรม เช่น Editor, Compiler, Debugger
- IDE สำหรับภาษาซีในปัจจุบัน
  - Code Blocks
  - Dev-C++
  - Turbo C++
  - Microsoft Visual C++

CodeBlock


Compile
แปลงโค้ดเป็นภาษาเครื่อง
Run
Compile & Run (F9)
เริ่ม Debug
รันทีละคำสั่ง
main.c [FirstProgram] - Code::Blocks 10.05
File Edit View Search Project Build Debug wxSmith Tools Plugins Settings Help
Build target: Debug
Management
Projects Symbols
Workspace
FirstProgram
Sources
main.c
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6 printf("Hello world!\n");
7 return 0;
8 }
9
ส่วนสำหรับ
พิมพ์ชุดคำสั่ง
ส่วนแสดงรายการ
ไฟล์ Source Code
Logs & others
Code::Blocks
Search results
Build log
Build messages
Debugger
ส่วนแสดง
ข้อผิดพลาด
C:\Users\Theerayut\Desktop\FirstPrograr
WINDOWS-874
Line 3, Column 1
Insert
Read/Write
default

กิจกรรม

ลบเครื่องหมาย Semicolon (;) ที่บรรทัดใดบรรทัดหนึ่งออก และ Compile หลังจากนั้นให้อ่านข้อความที่ปรากฏในส่วน Build messages



คำสั่งแสดง Output ทางหน้าจอ

- ใช้คำสั่ง printf ซึ่งมีรูปแบบดังนี้

```
printf("ข้อความ");
```

- สามารถใช้อักขระหลีก (Escape Character) ในข้อความได้
  - \n ขึ้นบรรทัดใหม่
  - \t เว้นช่องว่าง 1 tab
  - \" แสดงเครื่องหมาย "
  - \' แสดงเครื่องหมาย '

กิจกรรม

- จงเขียนโค้ดโปรแกรมภาษาซีเพื่อแสดงข้อความที่มีรูปแบบดังตัวอย่างด้านล่างนี้ โดยใช้อักขระหลีก \t และ \n ในการจัดรูปแบบการแสดงผล

```
=====================================
Code    Employee Name          Age
=====================================
e053    John Smith             22
e054    Chris Haggy            29
=====================================
```

การแทรกคำอธิบาย (Comment)

คำอธิบาย คือ ส่วนที่ Compiler จะข้ามการแปล

และไม่มีการแปลงเป็นภาษาเครื่อง

```
/* This is a plus calculation program.
 * By Mr.Kasang Nomoney */

#include <stdio.h>

void main()  {
    int value1, value2, sum;
    printf("Enter two value: ");
    scanf("%d %d",  &value1, &value2); // input two values
    sum = value1 + value2;  // calculate sum of input values
    // display output
    printf("Sum = %d\n", sum);
}
```

คำอธิบายแบบหลายบรรทัด

คำอธิบายแบบบรรทัดเดียว

คำสั่งแสดง Output ทางหน้าจอ

- แสดง Output พร้อมกับค่าในตัวแปร

printf("รูปแบบการแสดงผล", ชื่อตัวแปรหรือค่าที่ต้องการแสดงผล);

- รายการตัวแปรสำหรับแสดงค่า จะต้องมีลำดับสอดคล้องกับรหัสรูปแบบที่ระบุในรูปแบบการแสดงผล
- รหัสรูปแบบ
  - %d จำนวนเต็ม (integer)
  - %f เลขทศนิยม (floating-point)
  - %c อักขระ (character)
  - %s String หรือ ชุดอักขระ

ตัวอย่าง

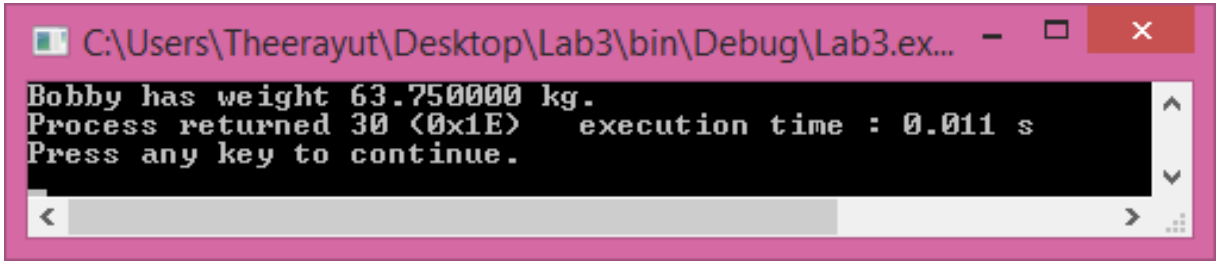
```
#include <stdio.h>
void main() {
    char grade = 'B';
    float gpa;
    gpa = 3.2;
    printf("Your grade is %c and GPA is %f", grade, gpa);
}
```

ผลลัพธ์

Your grade is B and GPA is 3.200000

กิจกรรม

- จงเพิ่มคำสั่งในการแสดงค่าในตัวแปรออกทางหน้าจอ โดยให้มีผลลัพธ์ดังภาพ




```
#include <stdio.h>
void main()
{
    char name[20] = "Bobby";
    float weight = 63.75;
    _____________________________
}
```

เพิ่มคำสั่งที่นี่

การจัดรูปแบบทศนิยม

- กำหนดรูปแบบทศนิยมโดยการระบุตัวเลขหลัง % เช่น

จองพื้นที่แสดงผลไว้ 9 ตำแหน่ง นับรวมจุดด้วย

แสดงทศนิยม 4 ตำแหน่ง

```
float dollar = 35.79856;
printf("Today dollar rate: %9.4f\n", dollar);
printf("Today dollar rate: %09.4f\n", dollar);
printf("Today dollar rate: %.2f", dollar);
```

```
Today dollar rate:   35.7986
Today dollar rate: 0035.7986
Today dollar rate: 35.80
```

การจำกัดช่องว่างสำหรับแสดงผล

- ใส่ตัวเลขหลังเครื่องหมาย % เพื่อจำกัดช่องว่างสำหรับการแสดงผล เช่น

```
char blood[3] = "AB";
printf("My blood type is %5s", blood);
```

```
My blood type is ^^^AB
```

```
int baht = 653;
printf("Remain %10d", baht);
```

```
Remain ^^^^^^^653
```

กิจกรรม

- จงเขียน Output ของจากการใช้คำสั่งจัดรูปแบบดังนี้

```
printf("%.1f, %.2f\n", 5.756, 5.756);

printf("Lab score is %3d and %03d\n", 2, 6);

printf("Annual income is %012.2f", 56788.369);
```

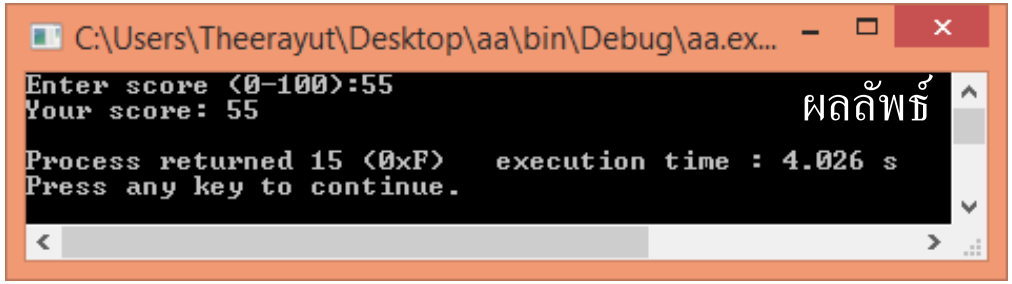
การรับข้อมูล (Input)

- ใช้คำสั่ง scanf เพื่อรอรับ Input จากผู้ใช้ทางคีย์บอร์ด

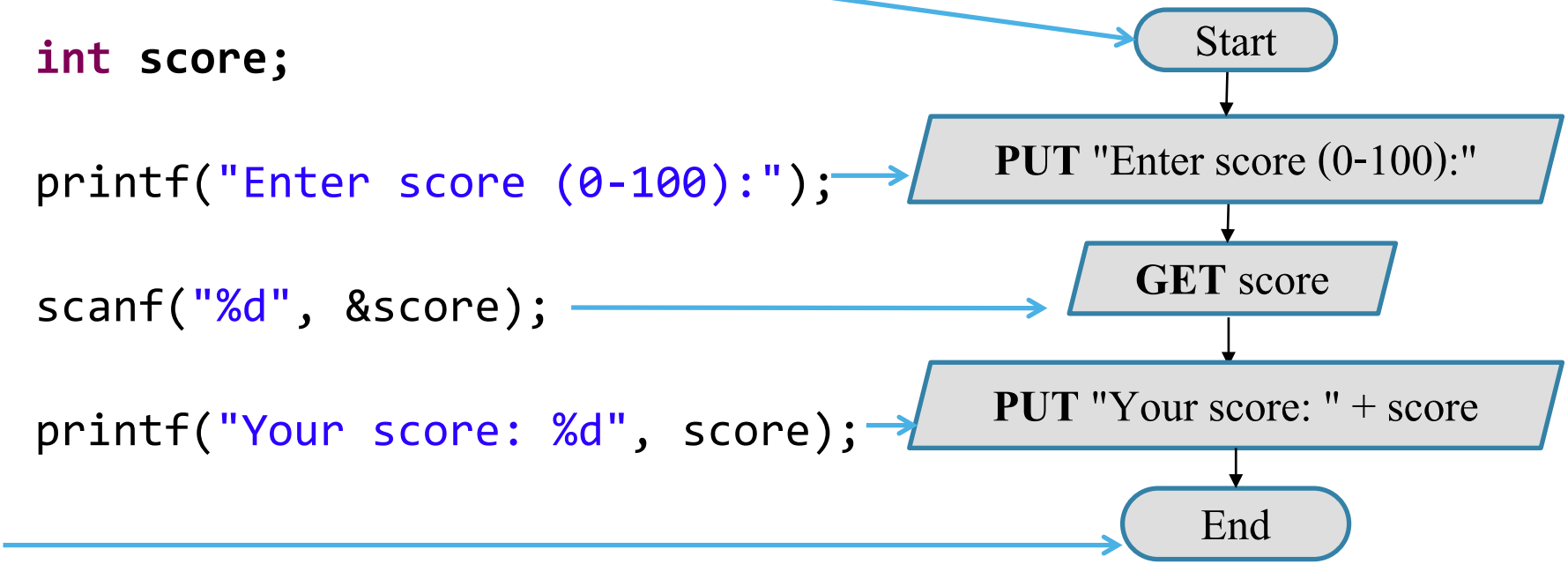
```
scanf("รูปแบบการรับค่า", ชื่อตัวแปรสำหรับเก็บค่า);
```

- ใส่สัญลักษณ์ & (ampersand) หน้าชื่อตัวแปรเสมอ
- การรับ input ต้องมีการประกาศตัวแปรก่อน เพื่อใช้พักค่าที่ผู้ใช้กรอก
- กำหนดรูปแบบของการรับค่าให้ตรงกับชนิดของตัวแปร โดยใช้รหัสรูปแบบเช่นเดียวกับคำสั่ง printf เช่น จะรับตัวเลข จะกำหนดเป็น %d
- ค่าที่ผู้ใช้กรอกจะถูกนำมาเก็บบนตัวแปรเมื่อผู้ใช้กดปุ่ม Enter

ตัวอย่างการรับค่าและแสดงผลลัพธ์

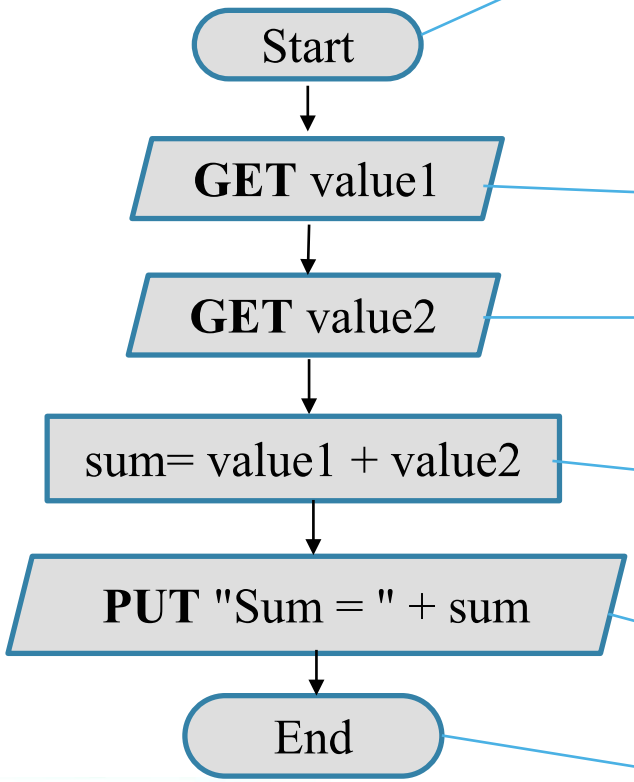



```
#include <stdio.h>

void main()  {

  int score;

  printf("Enter score (0-100):");

  scanf("%d", &score);

  printf("Your score: %d", score);

}
```



ตัวอย่างการรับค่า ประมวลผล และแสดงผลลัพธ์

```
Enter value 1:12
Enter value 2:3
Sum = 15
```
ผลลัพธ์




```c
#include <stdio.h>

void main() {

    int value1, value2, sum;

    // 1. Input
    printf("Enter value 1:");
    scanf("%d",  &value1);
    printf("Enter value 2:");
    scanf("%d",  &value2);

    // 2. Processing
    sum = value1 + value2;

    // 3. Output
    printf("Sum = %d\n", sum);

}
```

ตัวอย่างการรับค่าตามรูปแบบที่กำหนด

```c
#include <stdio.h>

void main()
{

    int value1, value2, sum;

    printf("Enter two value: ");
    scanf("%d %d", &value1, &value2);

    sum = value1 + value2;

    printf("Sum = %d\n", sum);
}
```

รับค่าตัวเลข 2 ตัวพร้อมกัน โดยคั่นด้วยช่องว่าง

```
Enter two value: 5 3
Sum = 8
```

ผลลัพธ์

การรับข้อมูลที่เป็นสตริง

- การรับค่า String อาจไม่ต้องใส่ & หน้าชื่อตัวแปรก็ได้ เช่น

```
scanf("%s", productName);
```

- ใส่ตัวเลขหลัง % เพื่อจำกัดจำนวนอักขระของ String ที่จะรับ

```
scanf("%25s", productName);
```

- กรณีรับค่า String ที่ผู้ใช้อาจกรอกช่องว่างมาด้วยจะใช้ %s ไม่ได้ ต้องใช้รูปแบบดังนี้

```
scanf(" %[^\n]", productName);
```

มี Space หน้า % ด้วย

[^\n] หมายถึง อ่านทุกตัวยกเว้น \n

กิจกรรม

- โปรแกรมรับชื่อ-สกุล อายุ ของตนเอง แล้วแสดงที่หน้าจอ จงเติมคำในช่องว่างให้สมบูรณ์

```
#include <stdio.h>
void main()
{
    char fname[20];
    int age;
    printf("Enter your name:");
    scanf("____", _______);
    printf("Enter your age:");
    scanf("____", _______);
    printf("_____________________", ______, ______);
}
```

ผลลัพธ์

```
Enter your name:Theerayut
Enter your age:35
Theerayut has 35 years old.
```

การรับข้อมูลที่เป็นอักขระ

- จะต้องใส่ช่องว่างหน้า %c เสมอ

```
scanf(" %c", &grade);
```

มี Space หน้า % ด้วย