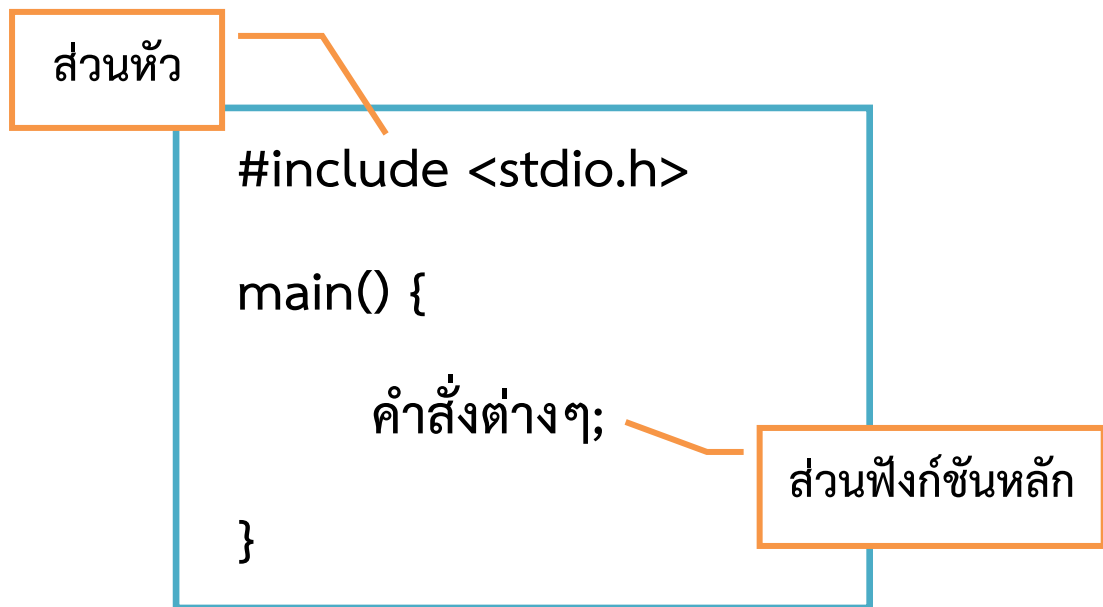


2.1 โครงสร้างโปรแกรม

การเขียนโปรแกรมภาษาซี นักเรียนต้องทำความเข้าใจกับความหมายของส่วนต่างๆ ของโครงสร้างโปรแกรม จะทำให้เขียนโปรแกรมได้ง่ายและเข้าใจที่มากขึ้น โดยโครงสร้างพื้นฐานของโปรแกรมภาษาซีเป็นดังนี้



ส่วนหัวของโปรแกรม จะเป็นส่วนสำหรับเรียกใช้ไฟล์จากไลบรารี (Library) เข้ามาร่วมประมวลผล โดยใช้ `#include` ตามด้วยชื่อไลบรารี ให้อยู่ระหว่างเครื่องหมาย `< >` ซึ่งไลบรารีที่ใช้บ่อยสุดคือ `stdio.h` ย่อมาจาก Standard Input/Output จะเก็บคำสั่งเกี่ยวกับการรับค่า/แสดงค่าของโปรแกรม

ไลบรารีที่พบบ่อยในภาษาซี

| | |
|-----------------------|------------------------------------|
| <code>stdio.h</code> | เกี่ยวกับการรับและแสดงผลข้อมูล |
| <code>conio.h</code> | เกี่ยวกับการรับและแสดงผลข้อมูล |
| <code>math.h</code> | เกี่ยวกับการคำนวณทางคณิตศาสตร์ |
| <code>string.h</code> | เกี่ยวกับการจัดการเกี่ยวกับข้อความ |
| <code>stdlib.h</code> | เกี่ยวกับการแปลงค่าข้อความ |

เพิ่มเติม

ส่วนฟังก์ชันหลักของโปรแกรม จะเป็นส่วนที่จะเก็บคำสั่งต่างๆ ที่จะให้โปรแกรมทำงาน ซึ่งจะมีที่ฟังก์ชันก็ได้ แต่จะต้องมีอย่างน้อยหนึ่งฟังก์ชันคือฟังก์ชัน main (ฟังก์ชันหลัก) โดยเขียนคำสั่งกำกับด้วย void main() ตามด้วยวงเล็บปีกกา { } หนึ่งคู่ คำสั่งทั้งหมดจะถูกเขียนไว้ในวงเล็บปีกกาซึ่งจะมีที่ร้อยก็พันคำสั่งก็ได้ โปรแกรมจะทำการประมวลผลให้ทั้งหมดเมื่อทำการรันโปรแกรม (ในกรณีที่คำสั่งไม่มีข้อผิดพลาด)

2.2 คำสั่งแสดงผล

คำสั่ง printf() เป็นฟังก์ชันสำหรับการแสดงผลหรือออกทางจอภาพ โดยให้พิมพ์ค่าที่ต้องการแสดงไว้ในเครื่องหมาย “ ” คำที่ต้องการแสดงจะเป็นตัวอักษรพิมพ์ใหญ่ พิมพ์เล็ก หรือผสมกันก็ได้ แล้วปิดท้ายคำสั่งด้วยเครื่องหมาย ;

```
printf(“คำที่ต้องการแสดง”);
```

โปรแกรมที่ 2.1 แสดงคำว่า HELLO ออกทางจอภาพ

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

```
main() {
```

```
    printf(“HELLO”);
```

```
}
```

ผลการรันโปรแกรม

HELLO

นอกจากนี้คำสั่ง printf() ยังสามารถแสดงผลร่วมกับการใช้รหัสแบ็กสแลช (Backslash) เช่น `\n` ในการขึ้นบรรทัดใหม่ `\t` ในการเว้นวรรคหนึ่ง tab โดยรหัสแบ็กสแลชจะต้องอยู่ระหว่างเครื่องหมาย “ ” ของคำสั่ง printf() เท่านั้น โดยจะวางไว้หน้าข้อความ หลังข้อความ หรือระหว่างข้อความก็ได้ เมื่อโปรแกรมอ่านเจอ รหัสแบ็กสแลชตรงไหนก็จะแสดงผลที่ตำแหน่งนั้นทันที

โปรแกรมที่ 2.2 แสดงคำว่า HELLO C LANGUAGE สองบรรทัด

```
#include <stdio.h>

main() {

    printf("HELLO\n");

    printf("C\tLANGUAGE");

}
```

ผลการรันโปรแกรม

```
HELLO
C    LANGUAGE
```

โปรแกรมจะทำการแสดงผลคำว่า HELLO จากนั้นพบ \n จึงทำการขึ้นบรรทัดใหม่ แล้วแสดงผลคำว่า C ตามด้วยเว้นวรรคหนึ่ง tab จาก \t แล้วตามด้วยคำว่า LANGUAGE

เพิ่มเติม

รหัสแบ็กสแลชที่พบบ่อยในภาษาซี

| | |
|----|----------------------|
| \n | ให้ขึ้นบรรทัดใหม่ |
| \t | ให้เว้นวรรคหนึ่ง tab |
| \0 | แสดงค่า NULL |
| \\ | แสดงเครื่องหมาย \ |
| \' | แสดงเครื่องหมาย ' |
| \" | แสดงเครื่องหมาย " |
| \a | ให้ส่งเสียงบี๊ป |

สื่อมัลติมีเดียที่ 2.1

หัวข้อ

คำสั่งแสดงผลในภาษาซี

Link

<http://www.sa.ac.th/c/2-1.mp4>

2.3 การแสดงผลตัวเลข

ในกรณีที่ต้องการแสดงผลเป็นตัวเลขด้วยคำสั่ง printf() จะไม่สามารถพิมพ์ตัวเลขเข้าไปใน “ ” โดยตรงได้ จะต้องแสดงผลผ่านรหัสความคุมการพิมพ์ % โดยในภาษาซีนั้นสามารถแสดงผลได้หลายรูปแบบ แต่ในช่วงเริ่มต้นนี้จะให้นักเรียนทดลองใช้เพียงแค่สองรูปแบบคือ การแสดงผลเป็นตัวเลขจำนวนเต็มให้ใช้เครื่องหมาย %d แทนที่ตำแหน่งของตัวเลข และการแสดงผลเป็นตัวเลขจำนวนจริงใช้เครื่องหมาย %f แทนที่ตำแหน่งของตัวเลข

```
printf(“รหัสควบคุม”,ตัวเลขที่ต้องการแสดง);
```

โปรแกรมที่ 2.3 แสดงตัวเลข 10 ออกทางหน้าจอ

```
#include <stdio.h>

main() {

    printf(“%d”,10);

}
```

ผลการรันโปรแกรม

10

โปรแกรมจะนำตัวเลขหลังเครื่องหมาย , ไปแทนที่ตำแหน่งรหัสควบคุมการพิมพ์ แล้วแสดงผลตัวเลขนั้นออกมาทางหน้าจอ



```
printf("%d",10);
```

แล้วถ้านักเรียนพิมพ์ตัวเลขไว้ระหว่าง “ ” เลยจะเป็นอย่างไร ให้ลองทดสอบด้วยโปรแกรมต่อไปนี้

โปรแกรมที่ 2.4 แสดงตัวเลข 10 สองแบบ

```
#include <stdio.h>

main() {

    printf("%d\n",10);

    printf(" 10");

}
```

ผลการรันโปรแกรม

```
10
 10
```

จากผลการรันโปรแกรมจะพบว่าการพิมพ์คำสั่งทั้งสองแบบได้ผลลัพธ์ออกมาเหมือนกันคือ 10 นักเรียนอาจสงสัยว่าแล้วทำไมจะต้องใช้รหัสควบคุมในการแสดงผลตัวเลข ให้นักเรียนทำความเข้าใจว่า 10 ในบรรทัดแรกคือตัวเลข 10 ส่วน 10 ในบรรทัดที่สองคือคำว่า 10 ซึ่งไม่มีคุณสมบัติในการคำนวณใดๆ ทั้งสิ้น ดังนั้นจะใช้รูปแบบไหนก็ขึ้นอยู่กับว่าตัวเลขตัวนั้นจะมีการนำไปใช้ในการคำนวณหรือไม่ หากแสดงค่าเฉยๆ เช่น เขียนโปรแกรมแสดง ชื่อ-เลขที่ของนักเรียน ในส่วนเลขที่ไม่ต้องทำการคำนวณใดๆ ก็สามารถใช้รูปแบบในบรรทัดที่สองในการแสดงผลได้เลย เป็นต้น

โปรแกรมที่ 2.5 ทดสอบการแสดงผลตัวเลข

```
#include <stdio.h>

main() {

    printf("The number is %d\n",10);

    printf("The number is %d\n",10*5);

    printf("The number is %f\n",5.2);

    printf("The number is %d and %d\n",5,6);

    printf("The number is %d and %f",5,5.2);

}
```

ผลการรันโปรแกรม

```
The number is 10

The number is 50

The number is 5.200000

The number is 5 and 6

The number is 5 and 5.200000
```

จากโปรแกรมที่ 2.5 จะแสดงถึงการแสดงผลตัวเลขรูปแบบต่างๆ ซึ่งจะเห็นว่าเราสามารถแสดงผลตัวเลขร่วมกับข้อความอื่นๆได้ภายใน printf() ตัวเดียว โดยแต่ละบรรทัดมีลักษณะดังนี้

บรรทัดที่ 1 แสดงให้เห็นว่าสามารถแสดงผลตัวเลขร่วมกับตัวอักษรได้

บรรทัดที่ 2 แสดงให้เห็นว่าสามารถกำหนดค่าเป็นสูตรคำนวณทางคณิตศาสตร์ได้ โดยโปรแกรมจะทำการคำนวณแล้วแสดงผลลัพธ์ที่ได้แทนที่รหัสควบคุม

บรรทัดที่ 3 แสดงให้เห็นว่าหากต้องการแสดงผลตัวเลขจำนวนจริงต้องใช้รหัสควบคุมเป็น %f

บรรทัดที่ 4 แสดงให้เห็นว่าใน printf() ตัวเดียวสามารถแทรกตัวเลขได้หลายตัวโดยใช้เครื่องหมาย , คั่นไปเรื่อยๆ โดยโปรแกรมจะนำตัวเลขหลัง , มาแทนที่รหัสควบคุมตามลำดับ

บรรทัดที่ 5 แสดงให้เห็นว่าใน printf() ตัวเดี่ยวนอกจากจะสามารถแทรกตัวเลขได้หลายตัวแล้ว ยังสามารถผสมได้ทั้งตัวเลขจำนวนเต็มและจำนวนจริง โดยจะนำตัวเลขมาแทนที่รหัสควบคุมตามลำดับ

จากโปรแกรมที่ 2.5 นักเรียนจะเห็นว่าเมื่อเราแสดงตัวเลข 5.2 ด้วย %f แล้ว กลับได้ผลลัพธ์เป็น 5.200000 (5.2 แล้วตามด้วยศูนย์อีก 5 ตัว) นั่นเป็นเพราะโปรแกรมกำหนดค่าพื้นฐานไว้ให้ตัวเลขทศนิยมแสดงผล 6 หลัก ซึ่งเราสามารถกำหนดจำนวนทศนิยมเองได้โดยการพิมพ์เครื่องหมาย . ตามด้วยจำนวนทศนิยมที่ต้องการ แทรกไว้ระหว่าง % กับ f ดังนี้

```
printf(“%.จำนวนทศนิยมf”,ตัวเลขจำนวนจริง);
```

```
เช่น printf(“%.2f”,5.2); ได้ผลลัพธ์ 5.20
```

```
printf(“%.0f”,5.2); ได้ผลลัพธ์ 5
```

เพิ่มเติม

รหัสควบคุมการพิมพ์ที่พบบ่อยในภาษาซี

| | |
|----|--|
| %d | ให้พิมพ์รูปแบบเลขจำนวนเต็มฐานสิบ |
| %u | ให้พิมพ์รูปแบบเลขจำนวนเต็มไม่มีเครื่องหมาย |
| %f | ให้พิมพ์รูปแบบเลขจำนวนจริง (ทศนิยม) |
| %o | ให้พิมพ์รูปแบบเลขฐานแปด |
| %x | ให้พิมพ์รูปแบบเลขฐานสิบหก |
| %c | ให้พิมพ์ตัวอักษรหนึ่งอักขระ |
| %s | ให้พิมพ์ชุดตัวอักษรหรือข้อความ |
| %% | ให้พิมพ์เครื่องหมาย % |

สื่อมัลติมีเดียที่ 2.2

หัวข้อ

การแสดงผลเป็นตัวเลข

Link

<http://www.sa.ac.th/c/2-2.mp4>

2.4 ตัวดำเนินการทางคณิตศาสตร์

ในการเขียนโปรแกรมคอมพิวเตอร์จะมีการนำตัวดำเนินการทางคณิตศาสตร์ (Operator) มาใช้ในการคำนวณค่าต่างๆ เพื่อให้ได้ผลลัพธ์ตามโจทย์ที่ต้องการ โดยตัวดำเนินการทางคณิตศาสตร์หลักๆ ที่ใช้ในภาษาซีมีดังนี้

| | |
|---|---------------------------|
| + | บวก (Addition) |
| - | ลบ (Subtraction) |
| * | คูณ (Multiplication) |
| / | หาร (Division) |
| % | การหารแบบเอาเศษ (Modulus) |

ตัวดำเนินการทางคณิตศาสตร์ บวก ลบ คูณ หาร เป็นอะไรที่คุ้นเคยในชีวิตประจำวันอยู่แล้ว แต่จะมีตัว % (mod) ที่เพิ่มเติมเข้ามา การ mod คือการหารที่จะเอาเฉพาะเศษที่เหลือโดยไม่สนใจว่าผลหารจะเป็นเท่าไร

เช่น

$$10\%3=1$$

$$10\%5=0$$

$$89451298424982698584184\%2=0$$

$$\begin{array}{r} \cancel{3} \\ 3 \overline{)10} \\ \underline{9} \\ 1 \end{array}$$

← mod

โปรแกรมที่ 2.6

การใช้ตัวดำเนินการทางคณิตศาสตร์

```
#include <stdio.h>

main() {

    printf(“%d”,1+2*3-4);

}
```

ผลการรันโปรแกรม

3

จากโปรแกรมที่ 2.6 เรื่องนี้จะเป็นปัญหาของเด็กไทย (รวมไปถึงผู้ใหญ่) ที่หลายคนคำนวณผลลัพธ์ได้ 5 ซึ่งเป็นปัญหาที่เกิดจากการเรียนในวิชาคณิตศาสตร์ไม่ได้เน้นในเรื่องนี้ ทั้งๆ ที่เป็นพื้นฐานการคำนวณที่ทั่วโลกใช้เป็นมาตรฐานร่วมกัน ในตัวดำเนินการทางคณิตศาสตร์จะมีสิ่งที่เรียกว่า **ลำดับความสำคัญของเครื่องหมาย** โดยเครื่องหมาย *, / และ % มีลำดับสูงกว่า + และ - ดังนั้นโปรแกรมจะทำการคำนวณเริ่มจาก $2*3$ ก่อน แล้วจึงทำการคำนวณด้วยการ + และ - จากซ้ายไปขวา เพราะ + และ - มีลำดับเท่ากัน หากต้องการให้โปรแกรมทำการคำนวณ $1+2$ ก่อน สามารถทำได้โดยใช้ตัวดำเนินการ () มาช่วย เพราะมีลำดับสูงกว่า *, / และ %

()

ทำในวงเล็บก่อนหากวงเล็บซ้อนกันทำวงเล็บในก่อน

* , / , %

มีลำดับเท่ากันทำการคำนวณจากซ้ายไปขวาตามลำดับ

+ , -

มีลำดับต่ำสุดจะทำการคำนวณเป็นลำดับสุดท้าย

ตัวอย่างที่ 2.1

โรงงานผลิตอาหารสัตว์ ได้วันละ 35 กิโลกรัม แบ่งบรรจุลงถุงถุงละ 700 กรัม ในหนึ่งสัปดาห์โรงงานจะผลิตอาหารสัตว์ได้กี่ถุง

```
#include <stdio.h>

main(){

    printf("Total weight that can produce in each day = 35 kg\n");

    printf("Total weight of food in each bag = 700 g\n");

    printf("The number of bags in 7 days = %d",35*1000/700*7);

}
```

ผลการรันโปรแกรม

Total weight that can produce in each day = 35 kg

Total weight of food in each bag = 700 g

The number of bags in 7 days = 350

จากตัวอย่างที่ 2.1 จะเห็นว่าตัวเลขในสองบรรทัดแรกเป็นการแสดงค่าออกทางหน้าจอเฉยๆ ไม่ได้มีการคำนวณในบรรทัดนี้ จึงสามารถพิมพ์ตัวเลขไว้ในเครื่องหมาย “ ” ของคำสั่ง printf() ได้เลย ส่วนในบรรทัดที่สามเป็นการคำนวณโดยการแปลงจำนวนที่ผลิตได้ต่อวันจากกิโลกรัมเป็นกรัมด้วยการคูณ 1000 แล้วนำมาหารด้วยจำนวนที่บรรจุต่อถุง จะได้จำนวนถุงที่ผลิตได้ต่อวันออกมา แล้วทำการคูณ 7 เพื่อให้ได้ผลลัพธ์ในหนึ่งสัปดาห์

ข้อสังเกตว่าเมื่อใดควรจะใช้รหัสควบคุมเป็น %d หรือ %f คือ ให้นักเรียนดูภาพรวมของโจทย์ก่อนว่ามีการใช้ค่าเป็นทศนิยมหรือไม่ จากนั้นดูไปถึงการใช้สูตรคำนวณว่ามีการหารหรือไม่ ถ้ามีการหารให้อนุมานว่ามีโอกาสที่ผลลัพธ์จะออกมาเป็นทศนิยม ให้นักเรียนใช้รหัสควบคุมเป็น %f ในโจทย์นั้น

สื่อมัลติมีเดียที่ 2.3

หัวข้อ

การใช้ตัวดำเนินการทางคณิตศาสตร์

Link

<http://www.sa.ac.th/c/2-3.mp4>

2.5 ตัวแปรและชนิดของข้อมูล

ตัวแปรคือสิ่งที่ตั้งขึ้นมาเพื่อใช้ในการเก็บค่า เนื่องจากการเขียนโปรแกรมมีหลายครั้งที่จะต้องนำค่าต่างๆ มาเก็บไว้เพื่อใช้งานในภายหลัง หรือรับค่าจากภายนอก เช่น โปรแกรมสื่อการเรียนมีการรับชื่อผู้ใช้เมื่อเข้าสู่ระบบ แล้วนำมาแสดงเมื่อทำแบบทดสอบเสร็จ โปรแกรมคำนวณน้ำหนักมีการเก็บค่าน้ำหนักของผู้ใช้มาทำการคำนวณ เป็นต้น การจะสร้างตัวแปรหรือประกาศตัวแปรได้นั้น สิ่งที่ต้องรู้เป็นอันดับแรกคือ ข้อมูลที่ต้องการเก็บเป็นข้อมูลชนิดอะไร ซึ่งในภาษาซีนั้นมีชนิดของข้อมูลหลายประเภท แต่สำหรับช่วงเริ่มต้นขอให้นักเรียนจำแค่ชนิดข้อมูลแบบตัวเลข ดังนี้

int ตัวเลขจำนวนเต็ม (รหัสควบคุม %d)

float ตัวเลขจำนวนจริง (รหัสควบคุม %f)

เมื่อเราระบุได้ว่าต้องการเก็บข้อมูลชนิดอะไร ให้ทำการประกาศตัวแปรตามรูปแบบดังนี้

ชนิดของข้อมูล ชื่อตัวแปร;

ชื่อตัวแปรในภาษาซีนั้นผู้เขียนโปรแกรมจะเป็นผู้ตั้งเองได้ตามต้องการ ภายใต้กฎการตั้งชื่อ ซึ่งกฎการตั้งชื่อนี้จะใช้กับชื่อต่างๆ ในโปรแกรมทั้งหมด เช่น การกำหนดชื่อโปรแกรม ชื่อตัวแปร ชื่อฟังก์ชัน เป็นต้น โดยกฎการตั้งชื่อมีรูปแบบดังนี้

1. จะต้องขึ้นต้นด้วยตัวอักษรภาษาอังกฤษ โดยตัวอักษรพิมพ์ใหญ่ พิมพ์เล็ก ถือเป็นคนละตัวกัน หรือขึ้นต้นด้วยเครื่องหมาย _ (Underscore)
2. ตัวต่อไปสามารถเป็นตัวอักษรผสมกับตัวเลขและเครื่องหมาย _ ก็ได้
3. ชื่อต้องไม่มีการเว้นวรรค
4. ต้องไม่ซ้ำกับคำสงวน ซึ่งมีดังต่อไปนี้

| | | | |
|-----------|-----------------------|-----------------------|-----------|
| abstract | as | base | bool |
| break | byte | case | catch |
| char | checked | class | const |
| continue | decimal | default | delegate |
| do | double | else | enum |
| event | explicit | extern | false |
| finally | fixed | float | for |
| foreach | goto | if | implicit |
| in | in (generic modifier) | int | interface |
| internal | is | lock | long |
| namespace | new | null | object |
| operator | out | out(generic modifier) | override |
| params | private | protected | public |
| readonly | ref | return | sbyte |

ซึ่งคำสงวนนี้นักเรียน**ไม่จำเป็นต้องท่อง** ให้ทำความเข้าใจว่าคำใดที่มีการใช้เป็นคำสั่ง โครงสร้าง หรือฟังก์ชันในการเขียนโปรแกรมภาษาซีแล้ว ห้ามเอามาตั้งชื่อตัวแปร เพราะจะทำให้โปรแกรมไม่สามารถประมวลผลได้ว่านี่คือคำสั่งหรือตัวแปร

ในฐานะที่นักเรียนเป็นโปรแกรมเมอร์มือใหม่ ในช่วงแรกนี้อาจจะตั้งชื่อตัวแปรง่ายๆ ด้วยตัวอักษรภาษาอังกฤษตัวเดียว เช่น a , b , c เป็นต้น เมื่อนักเรียนค่อนข้างคุ้นเคยกับการเขียนโปรแกรมแล้ว ให้เริ่มตั้งชื่อตัวแปรให้สื่อความหมาย เช่น เขียนโปรแกรมคำนวณน้ำหนัก อาจตั้งชื่อตัวแปรเป็น weight เขียนโปรแกรมเก็บค่าอายุ อาจตั้งชื่อตัวแปรเป็น age เป็นต้น

```
int a;  
float b,c;
```

จากรูป เราทำการประกาศตัวแปร a เป็น int หลังจากนั้นเมื่อมีการอ้างถึง a โปรแกรมก็จะรู้ว่านั่นคือตัวแปรที่มีคุณสมบัติในการเก็บค่าตัวเลขจำนวนเต็ม เช่นเดียวกันเมื่อประกาศตัวแปรเป็น float จะหมายถึงตัวแปรตัวนั้นมีคุณสมบัติในการเก็บค่าตัวเลขจำนวนจริง นอกจากนี้ถ้าเรามีการสร้างตัวแปรหลายตัว และตัวแปรเป็นข้อมูลชนิดเดียวกัน สามารถประกาศรวมกันได้ โดยใช้เครื่องหมาย , คั่นระหว่างตัวแปร

โปรแกรมที่ 2.7 เขียนโปรแกรมเก็บค่าตัวเลข 10

```
#include <stdio.h>  
  
main() {  
  
    int a;  
  
    a=10;  
  
    printf(“%d”,a);  
  
}
```

ผลการรันโปรแกรม

10

จากโปรแกรมที่ 2.7 แสดงให้เห็นถึงการกำหนดค่าให้กับตัวแปร สามารถทำได้โดยใช้เครื่องหมาย = ค่าฝั่งขวามือของ = จะถูกเก็บไว้ในตัวเก็บค่าฝั่งซ้ายมือ ซึ่งการกำหนดค่านี้อาจรวมในบรรทัดที่ประกาศตัวแปรก็ได้ เช่น `int a=10;` หลังจากกำหนดค่าให้กับตัวแปรแล้ว หลังจากนั้นเมื่ออ้างถึง a จะหมายถึงค่า 10 ซึ่งเราสามารถนำไปใช้ในการคำนวณหรือแสดงผลก็ได้ โดยใช้วิธีเดียวกับการคำนวณหรือแสดงผลด้วยตัวเลข อย่งในกรณีนี้ต้องการแสดงผลตัวแปร a ก็ใช้รหัสควบคุม `%d` แทนที่ตำแหน่งเหมือนกับการแสดงผลตัวเลขจำนวนเต็ม

ตัวอย่างที่ 2.2

โรงงานผลิตอาหารสัตว์ ได้วันละ 35 กิโลกรัม แบ่งบรรจุลงถุงถุงละ 700 กรัม ในหนึ่งสัปดาห์โรงงานจะผลิตอาหารสัตว์ได้กี่ถุง

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

```
main(){
```

```
float a=35,b=700;
```

```
printf("Total weight that can produce in each day = %.0f kg\n",a);
```

```
printf("Total weight of food in each bag = %.0f g\n",b);
```

```
printf("The number of bags in 7 days = %.0f",a*1000/b*7);
```

```
}
```

ผลการรันโปรแกรม

Total weight that can produce in each day = 35 kg

Total weight of food in each bag = 700 g

The number of bags in 7 days = 350

เพิ่มเติม

ชนิดของข้อมูลที่พบบ่อยในภาษาซี

| | |
|--------|--|
| int | ตัวเลขจำนวนเต็ม มีค่าอยู่ในช่วง -32,768 ถึง 32,767 |
| long | ตัวเลขจำนวนเต็ม มีค่าอยู่ในช่วง -2,147,483,648 ถึง 2,147,483,647 |
| float | ตัวเลขจำนวนจริง มีค่าอยู่ในช่วง 3.4×10^{-38} ถึง 3.4×10^{38} |
| double | ตัวเลขจำนวนจริง มีค่าอยู่ในช่วง 1.7×10^{-308} ถึง 1.7×10^{308} |
| char | ตัวอักษรหนึ่งอักขระ |

นอกจากนี้ยังมีตัวดำเนินการทางคณิตศาสตร์ของภาษาซีอีกกลุ่มหนึ่งที่สามารถใช้ร่วมกับตัวแปร ดังนี้

| | |
|----|----------------------|
| ++ | การเพิ่มค่าขึ้นหนึ่ง |
| -- | การลดค่าลงหนึ่ง |

ตัวดำเนินการ ++ และ - ต้องใช้ร่วมกับตัวแปร โดยเมื่อนำไปวางไว้ข้างหน้าตัวแปรหรือหลังตัวแปรจะทำให้ตัวแปรตัวนั้นเพิ่มหรือลดค่าหนึ่ง เช่น เรามีตัวแปร a มีค่าเป็น 5 เมื่อเขียนคำสั่ง a++; ตัวแปร a จะมีค่าเป็น 6 ทั้งนี้การวางตัวดำเนินการไว้หน้าตัวแปรหรือหลังตัวแปรมีลำดับการทำงานต่างกันเล็กน้อย นักเรียนอาจศึกษาเพิ่มเติมได้ตามตำราเรียนเขียนโปรแกรมทั่วไป

โปรแกรมที่ 2.8 การเพิ่มค่าด้วย ++

```
#include <stdio.h>

main() {

    int a=10;

    a++;

    printf(“%d”,a);

}
```

ผลการรันโปรแกรม

11

เพิ่มเติม

ข้อสังเกตการใช้ตัวดำเนินการ ++ , --

ถึงแม้ว่าตัวดำเนินการ ++ , -- จะสามารถวางไว้หน้าตัวแปรหรือหลังตัวแปรก็ได้ แต่ต้องทำความเข้าใจว่าลำดับการทำงานนั้นจะไม่เหมือนกัน

- a++ จะนำค่าเดิมของตัวแปร a ไปใช้ในคำสั่งบรรทัดนั้นก่อน เมื่อจบคำสั่งจึงเพิ่มค่า
- ++a จะทำการเพิ่มค่าตัวแปร a ก่อน แล้วค่อยนำค่าใหม่ไปใช้ในคำสั่งบรรทัดนั้น

สื่อมัลติมีเดียที่ 2.4

หัวข้อ

การประกาศตัวแปร

Link

<http://www.sa.ac.th/c/2-4.mp4>

2.6 คำสั่งรับค่า

โปรแกรมคอมพิวเตอร์ที่ดีจะต้องมีการสื่อสารสองทางระหว่างผู้ใช้โปรแกรมและโปรแกรม ในหัวข้อที่ผ่านมาโปรแกรมของเราจะเป็นการสื่อสารทางเดียวคือโปรแกรมแสดงผลให้กับผู้ใช้เท่านั้น ดังนั้นในหัวข้อต่อไปนี้จะเป็นการใช้คำสั่งรับค่า เพื่อให้ผู้ใช้สามารถส่งค่าจากคีย์บอร์ดให้กับโปรแกรมเพื่อนำไปประมวลผล หลังจากนั้นโปรแกรมจะแสดงผลลัพธ์ที่ได้กลับมายังผู้ใช้ ซึ่งคำสั่งในการรับค่าของภาษาซีคือคำสั่ง `scanf()` มีรูปแบบดังนี้

```
scanf("รหัสควบคุม",&ตัวแปร);
```

การจะรับค่าจากคีย์บอร์ดได้นั้นอันดับแรกจะต้องรู้ก่อนว่าค่าที่ต้องการรับนั้นเป็นข้อมูลชนิดอะไร ให้ประกาศตัวแปรหนึ่งตัวสำหรับการรับค่านั้น ให้นักเรียนจำไว้ว่าถ้ามีการใช้คำสั่ง `scanf()` ในโปรแกรม จะต้องมีการประกาศตัวแปรอย่างน้อยหนึ่งตัวเสมอ ไม่เช่นนั้นเมื่อรับค่ามาแล้วโปรแกรมก็ไม่รู้ว่าจะนำค่าไปเก็บไว้ที่ใด ระหว่างเครื่องหมาย “ ” ของ `scanf` ให้ระบุรหัสควบคุมโดยกำหนดให้ตรงกับชนิดของข้อมูลที่ได้รับ แล้วตามด้วยเครื่องหมาย & ตามด้วยชื่อของตัวแปรหลังเครื่องหมาย , ซึ่งเครื่องหมาย & นั้นจะเป็นการอ้างถึงตำแหน่งการเก็บข้อมูลของตัวแปร เมื่อพิมพ์ค่าแล้วกด Enter ค่านั้นจะถูกเก็บไว้ในตัวแปรที่กำหนด

```
scanf("%d",&a);
```



**รูปแบบที่มักใช้ผิด

```
scanf("%d",a);
```

✘

```
scanf("%d,&a");
```

✘

```
scanf("%d ",&a);
```

✘

โปรแกรมที่ 2.9

เขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มหนึ่งตัว

```
#include <stdio.h>

main() {

    int a;

    scanf("%d",&a);

}
```

ผลการรันโปรแกรม

_

จากโปรแกรมที่ 2.9 โจทย์ให้เขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มหนึ่งตัว ดังนั้นเราต้องประกาศตัวแปรเป็น int จำนวนหนึ่งตัวสำหรับการรับค่าด้วยคำสั่ง scanf() ซึ่งผลลัพธ์ที่ได้จากการรันโปรแกรม จะปรากฏเครื่องหมาย _ กระจับปวยอยู่ ซึ่งเกิดจากคำสั่ง scanf() เป็นการรอรับค่าจากคีย์บอร์ด เมื่อทำการพิมพ์ค่าแล้วกดปุ่ม Enter ค่านั้นจะถูกเก็บไว้ตัวแปร

จากผลการรันโปรแกรมนักเรียนจะเห็นว่าผลลัพธ์จะมีเพียง _ กระจับปวยอยู่ ผู้เขียนโปรแกรมอาจจะรู้ว่าให้กรอกตัวเลขจำนวนเต็ม แต่การสร้างโปรแกรมนั้นให้เราระลึกอยู่เสมอว่าโปรแกรมที่สร้างอาจมีผู้ใช้โปรแกรมคนอื่นมาใช้ก็ได้ จากโปรแกรมที่ 2.8 ผู้ใช้ไม่สามารถรู้ได้เลยว่าให้ทำอะไร ดังนั้นคำสั่ง scanf() มักใช้ร่วมกับคำสั่ง printf() เพื่อแสดงข้อความให้ผู้ใช้โปรแกรมรู้ว่าในบรรทัดนี้จะให้รับค่าอะไร เมื่อรับค่าไปแล้วจะนำค่านั้นไปทำอะไรก็สามารถแสดงผลออกมาด้วยคำสั่ง printf() เช่นกัน

โปรแกรมที่ 2.10 เขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มหนึ่งตัว

```
#include <stdio.h>

main() {

    int a;

    printf("Input number : ");

    scanf("%d",&a);

    printf("Your number is %d",a);

}
```

ผลการรันโปรแกรม

```
Input number : 10
                ↵
Your number is 10
```

ตัวอย่างที่ 2.3

เขียนโปรแกรมคำนวณจำนวนผลผลิตของโรงงานผลิตอาหารสัตว์ โดยให้รับค่าว่าในหนึ่งวันผลิตได้กี่กิโลกรัม แบ่งบรรจุลงถุงถุงละกี่กรัม แล้วคำนวณออกมาว่า ในหนึ่งสัปดาห์โรงงานจะผลิตอาหารสัตว์ได้กี่ถุง

```
#include <stdio.h>

main(){

    float a,b;

    printf("Total weight that can produce in each day (kg) = ");

    scanf("%f",&a);
```

(มีต่อ)

```
printf("Total weight of food in each bag (g) = ");
scanf("%f",&b);
printf("The number of bags in 7 days = %.0f",a*1000/b*7);
}
```

ผลการรันโปรแกรม

Total weight that can produce in each day (kg) = 50

Total weight of food in each bag (g) = 500

The number of bags in 7 days = 700

** เมื่อเจอโจทย์การเขียนโปรแกรม อันดับแรกนักเรียนควรอ่านโจทย์ทั้งหมดก่อนเพื่อวิเคราะห์ภาพรวมของโจทย์ว่าโจทย์ข้อนี้มีการรับค่ากี่ค่า และค่าที่รับควรเป็น int หรือ float โดยดูว่าค่าที่รับนั้นเป็นทศนิยมหรือไม่ หรือการคำนวณมีการหารหรือไม่ ถ้ามีการหารผลลัพธ์ที่ได้ก็มีโอกาสที่จะออกมาเป็นทศนิยม ดังนั้นถ้าเจอกรณีนี้ให้นักเรียนกำหนดตัวแปรเป็น float

สื่อมัลติมีเดียที่ 2.5

หัวข้อ

การรับค่าจากคีย์บอร์ด

Link

<http://www.sa.ac.th/c/2-5.mp4>

ตัวอย่างที่ 2.4

จงเขียนโปรแกรมแปลงค่าปี พ.ศ. เป็นปี ค.ศ.

```
#include <stdio.h>

main(){

    int a;

    printf("B.E. = ");

    scanf("%d",&a);

    printf("A.D. = %d",a-543);

}
```

ผลการรันโปรแกรม

```
B.E. = 2558
A.D. = 2015
```

** ปี พ.ศ. แปลงเป็นปี ค.ศ. ไม่มี
ทางที่ผลลัพธ์จะมีทศนิยมแน่นอน
ข้อนี้จึงใช้ int

ตัวอย่างที่ 2.5

จงเขียนโปรแกรมแปลงหน่วยการวัด โดยรับค่าเป็นนิ้ว แล้วแสดงผลเป็นเซนติเมตรออกมา (กำหนดให้
1 นิ้ว = 2.54 เซนติเมตร)

```
#include <stdio.h>

main(){

    float a;

    printf("input inch = ");

    scanf("%f",&a);

    printf("centimeter = %.2f",a*2.54);

}
```

ผลการรันโปรแกรม

```
input inch = 12
centimeter = 30.48
```

** สูตรการแปลงหน่วยในข้อนี้มีค่า
เป็นทศนิยม ดังนั้นต้องใช้ float

ตัวอย่างที่ 2.6

** ต่อไปจะทำการตั้งชื่อตัวแปรให้เหมาะสมกับค่าที่รับตามโจทย์

จงคำนวณหาพื้นที่วงกลม โดยรับค่ารัศมีจากคีย์บอร์ด

```
#include <stdio.h>

main(){

    float radius;

    printf("input radius = ");

    scanf("%f",&radius);

    printf("area = %.2f",3.14*radius*radius);

}
```

ผลการรันโปรแกรม

```
input radius = 7
area = 153.86
```

** สูตรพื้นที่วงกลมคือ πr^2 ซึ่ง
นิยามของการยกกำลังสองคือการ
คูณตัวเอง 2 ครั้ง ดังนั้นจึงนำค่า
radius มาคูณ 2 ครั้ง

ตัวอย่างที่ 2.7

จงเขียนโปรแกรมคำนวณภาษี 7% โดยรับค่าราคาสินค้าแล้วแสดงผลว่าต้องเสียภาษีกี่บาท และต้อง
จ่ายจริงกี่บาท

```
#include <stdio.h>

main(){

    float price;

    printf("PRICE = ");

    scanf("%f",&price);

    printf("VAT = %.2f BAHT",price*7/100);

    printf("NET = %.2f BAHT",price*7/100+price);

}
```

ผลการรันโปรแกรม

```
PRICE = 250
VAT = 17.50 BAHT
NET = 267.50 BAHT
```

** โจทย์ข้อนี้มีการรับค่า 1 ค่า แต่
ต้องการผลลัพธ์ 2 อย่าง คือภาษีและ
ราคาสุทธิ เนื่องจากมีการหาร้อยละ
ทำให้มีผลเป็นทศนิยมจึงต้องใช้ float

ตัวอย่างที่ 2.8

จงเขียนโปรแกรมหาพื้นที่สามเหลี่ยม โดยรับค่าฐานและความสูง

```
#include <stdio.h>

main(){

    float base,height;

    printf("BASE = ");

    scanf("%f",&base);

    printf("HEIGHT = ");

    scanf("%f",&height);

    printf("AREA = %.2f",0.5*base*height);

}
```

ผลการรันโปรแกรม

```
BASE = 4
HEIGHT = 5
AREA = 10
```

** โจทย์ข้อนี้มีการรับค่า 2 ค่า จึง
ต้องทำการกำหนดตัวแปร 2 ตัว เป็น
float เพราะสูตรพื้นที่สามเหลี่ยมมี
การหาร 2

