

ใบงานการทดลองที่ 3  
เรื่อง คำสั่งพื้นฐานของภาษาซี โครงสร้าง และมาตรฐานการเขียนโค้ด

1. จุดประสงค์ทั่วไป

- 2.1. รู้และเข้าใจหลักภาษาที่ใช้ในการพัฒนาโปรแกรมคอมพิวเตอร์
- 2.2. รู้และเข้าใจมาตรฐานในการเขียนโค้ดโปรแกรมแบบ Codex

2. จุดประสงค์เชิงพฤติกรรม

- 2.1.1. บอกและอธิบายลักษณะของภาษาซี
- 2.1.3. บอกและอธิบายโครงสร้างของภาษาซี
- 2.1.4. บอกคำสั่งที่ใช้ในการการรับค่าและการแสดงผล
- 2.1.5. ฝึกหัดและทดลองใช้คำสั่งรับค่าและแสดงผล
- 2.2.1. บอกและอธิบายการจัดโครงสร้างของงาน
- 2.2.2. บอกและอธิบายการจัดโครงสร้างของโค้ดโปรแกรม
- 2.2.3. ฝึกหัดและทดลองใช้จัดโครงสร้างของโค้ดโปรแกรม
- 2.2.4. ออกแบบแนวทางการใช้จัดโครงสร้างของโค้ดโปรแกรมเพื่อให้ทำงานได้มีประสิทธิภาพมากขึ้น
- 2.2.5. แนะนำแนวทางการใช้จัดโครงสร้างของโค้ดโปรแกรมอย่างเป็นระบบ
- 2.2.6. บอกและอธิบายการตั้งชื่อตัวแปรให้สอดคล้องกับโครงสร้างหลัก
- 2.2.7. ฝึกหัดและทดลองใช้การตั้งชื่อตัวแปรให้สอดคล้องกับโครงสร้างหลัก
- 2.2.8. บอกและอธิบายการนิยามความหมายของตัวแปรและฟังก์ชัน
- 2.2.9. ฝึกหัดและทดลองใช้การนิยามความหมายของตัวแปรและฟังก์ชัน
- 2.2.10. บอกและอธิบายการเว้นวรรคตอน
- 2.2.11. ฝึกหัดและทดลองใช้การเว้นวรรคตอน
- 2.2.12. บอกและอธิบายการใช้งานเครื่องหมายวงเล็บ
- 2.2.13. ฝึกหัดและทดลองใช้การใช้งานเครื่องหมายวงเล็บ

3. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่องที่ติดตั้งโปรแกรม Dev-C

4. ทฤษฎีการทดลอง

- 4.1. จงบอกและอธิบายที่มาและลักษณะพิเศษของ “ภาษาซี”  
ภาษาซีเป็นภาษาที่พัฒนาโปรแกรมคอมพิวเตอร์ที่ใช้งาน มีต้นกำเนิดในการ  
พัฒนาระบบ มีลักษณะการเขียนโปรแกรมเชิงโครงสร้าง ถูกพัฒนามาตั้งแต่  
ปี ค.ศ. 1969-1973 โดย เดนนิส ริทชี (Dennis Ritchie)  
ซึ่งรูปแบบโครงสร้างของภาษาซีถูกนำไปประยุกต์ใช้เพื่อเป็นต้นแบบให้กับโครงสร้าง  
ของหลายภาษาที่พัฒนาในยุคปัจจุบัน เช่น จาวา, ภาษาสคริปต์, พีเอชพี, โปทอน ฯลฯ



#### 4.2. จงบอกและอธิบายโครงสร้างของภาษาซีโดยทั่วไป

```

1. #include <stdio.h>      ทำเพื่อคัดลอกไฟล์ฐานต่างๆ
2. #include <conio.h>      ทำเพื่อคัดลอกไฟล์ฐานต่างๆ
3. int main() {            ส่วนของโปรแกรมหลัก (มีที่รันหลัก)
4.     printf("Hello world");    คำสั่ง printf ของภาษาซี โดยรับค่าจากตัวแปร
5.     return 0;            คำสั่งการคืนค่า ซึ่งถ้าได้โครงสร้างที่ครบถ้วนแล้ว
6. } //end function        ปิดท้ายเพื่อจบการทำงานของโครงสร้างของฟังก์ชันหลัก
    
```

#### 4.3. จงบอกและอธิบายลักษณะการใช้งานคำสั่ง printf พร้อมยกตัวอย่างประกอบ

printf เป็นฟังก์ชันที่ใช้ในการแสดงผลข้อมูลออกทางจอภาพ โดยสามารถกำหนดรูปแบบข้อมูล (format code) และรับค่าข้อมูล (variable code)  
 Ex printf("Hello world"); แสดงผลทางหน้าจอว่า Hello world  
 เมื่อรูปแบบกับข้อมูล

#### 4.4. จงบอกและอธิบายลักษณะการใช้งานคำสั่ง scanf พร้อมยกตัวอย่างประกอบ

scanf เป็นฟังก์ชันซึ่งเป็นฟังก์ชันมาตรฐานสำหรับรับข้อมูลจากคีย์บอร์ด โดยสามารถรับได้ทุกประเภท ไม่ว่าจะเป็นจำนวนเต็ม, ทศนิยม, อักษร  
 Ex printf("Enter figure (Y:N):");, แล้วย้ายตัวชี้ไปยังตัวแปร Y:N  
 scanf("%c", &answer); รับอักษรมาเก็บไว้ในตัวแปร answer

#### 4.5. จงบอกและอธิบายการจัดโครงสร้างของโค้ดโปรแกรมตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

มาตรฐานการเขียนโค้ดแบบ Codex เป็นมาตรฐานที่ใช้ใน Wordpress CMS ซึ่งเป็นเครื่องมือสำหรับเว็บไซต์แบบทำเว็บที่พัฒนาโดยใช้ภาษา PHP  
 ข้อดีคือ สะดวก ตกแต่งง่าย และทั้งชื่อโครงการจะต้องตั้งด้วยชื่อภาษาอังกฤษ (ทั้งหมด 3 ตัว)  
 ไม่ใช่ชื่อที่บอกถึงตัวโครงการเป็นคำศัพท์ง่าย ๆ ได้ใจความ การเขียนชื่อโครงการให้ชัดเจน  
 ตัวอย่าง ( ) เช่น Calculator\_for\_student

#### 4.6. จงบอกและอธิบายการจัดตั้งชื่อตัวแปรตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

ทุกตัวตั้งชื่อด้วยภาษาอังกฤษ (พิมพ์เล็ก) ไม่ขึ้นด้วยตัวอักษรถึงตัวโครงการนั้นๆ  
 เป็นคำที่กระชับได้ใจความ การเขียนชื่อตัวแปรให้ชัดเจน ( ) เช่น Calculator\_for\_student  
 การตั้งชื่อ ให้มีความยาว, งานน้อย, ชื่อโปรแกรมสั้นๆ, ตัวแปรต้องตั้งชื่อให้ตรงกับชื่อของโครงการ  
 แล้วจึงขึ้นด้วยชื่อของตัวแปรที่จะใช้เก็บค่าในครั้งนั้นๆ เช่น กำหนดชื่อของตัวแปรให้เก็บค่าชื่อ  
 calculator\_for\_student\_html, ชื่อของตัวแปรที่รวม calculator\_for\_student\_count-loop

ชื่อของโปรแกรมคือ calculator\_for\_student\_find\_maximum\_value() เป็นต้น



4.7. จงบอกและอธิบายการนิยามตัวแปรและฟังก์ชันตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

```
printf("--I show 'Hello-World' to screen %d time(s)\n",N);
for (i=0;i<N;i++){
    printf( "[%3d] Hello World\n";i );
} //end for
```

4.8. จงบอกและอธิบายการเว้นวรรคตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

การเว้นวรรคหนึ่งช่วงอยู่ในบรรทัดโปรแกรม โดย  
ลักษณะการใช้มี 3 อย่าง คือ ต้องเว้นวรรคตอนในทุกช่วงของ  
การเขียน เช่น ช่วงตัว r = ช่วงเครื่องหมายวงเล็บ =  
Ex int i = 0 , sum = 0 , n = 0 ;

4.9. จงบอกและอธิบายการใช้งานเครื่องหมายวงเล็บตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

เครื่องหมายวงเล็บปีกกา ( ) จะใช้บอกให้เห็นโครงสร้างของโปรแกรม  
เช่น ได้โปรแกรมคืออยู่ภายใต้โครงสร้าง ซึ่ง = ช่วงดังกล่าวเรียกว่า "บล็อก"  
เพื่ออธิบายให้ชัดโปรแกรมอยู่ภายใต้โครงสร้างนี้ทั้งหมดจะกลับเข้าสู่โครงสร้าง  
เดิมเมื่อพบเครื่องหมายวงเล็บปีกกา ( ) int i = 0 , sum = 0 , n = 0 ;

} //end function

5. ลำดับชั้นการปฏิบัติงาน

5.1. การรับและการแสดงผล

5.1.1. จงเขียนโปรแกรมเพื่อรับตัวเลขจำนวนเต็ม ตัวเลขทศนิยม และตัวอักษร เพื่อแสดงผลบนหน้าจอคอมพิวเตอร์

Test case 1	Test Case 2
Input Please enter value : 5 12.23945 F	Input Please enter value : 9 13.1 M
Output 5 12.24 F	Output 9 13.10 M

### 5.1.2. บันทึกโค้ดโปรแกรมดังกล่าว

ผังงาน	โค้ดโปรแกรม
<pre> graph TD     Start([start]) --&gt; Init[กำหนดตัวแปร a, b, c เป็นจำนวนเต็ม; float; char ตามระบ.]     Init --&gt; Prompt[/แสดงคำว่า Please enter value;/]     Prompt --&gt; Input[/รับค่าจำนวนเต็ม float char ลงใน a, b, c/]     Input --&gt; Print[/แสดงค่าจำนวนเต็ม float char ตามตัวแปร a, b, c/]     Print --&gt; End([End])         </pre>	<pre> #include &lt;stdio.h&gt;  int main() {     int a;     float b;     char c;     printf("Please enter value:");     scanf("%d %f %c", &amp;a, &amp;b, &amp;c);     printf("%d %f %c", a, b, c);     return 0; }         </pre>

### 5.2. มาตรฐานการเขียนโค้ดโปรแกรมตามมาตรฐาน Codex

5.2.1. จากโค้ดโปรแกรมต่อไปนี้ จงจัดระเบียบให้อยู่ภายใต้มาตรฐานการเขียนโปรแกรมแบบ Codex พร้อมทั้งแก้ไขข้อผิดพลาด (Bug) เพื่อให้โปรแกรมสามารถทำงานได้ตามปกติ

โค้ดโปรแกรม
<pre> #include &lt;stdio.h&gt;  int main( ) { int i=0;int N=7; printf( "--  Show 'Hello World' to Screen %d line(s) \n", N ) ; for(i=0;i&lt;N;i++) { printf("[%3d] Hello World \n", i );}     char star='*' int j = 0 ; N = 8 ; i = 0 ; j = 0 ;printf( "\n" ) ; while( i &lt; N ) { j = 0 ;while( j &lt; N ) { if( i &lt; j ) printf( "%c ", star ) ; j++ ; } printf( "%s", "\n" ; i += 1 ; }return 0 ; }         </pre>



5.2.2. บันทึกผลโค้ดโปรแกรมที่ถูกจัดระเบียบแล้วลงตารางดังต่อไปนี้

โปรแกรม

```
#include <stdio.h>

int main(){
    int i, j    = 0 ;
    int N      = 7 ;
    char star   = '*';

    printf("--I Show 'Hello World' to Screen %d line(s)\n", N);
    for(i=0; i < N; i++) {
        printf( "[%3d] Hello World\n", i );
    }

    N=8;
    i=0;
    printf("\n");
    while(i<N){
        j=0;
        while(j<N){
            if(i<j){
                printf( "%c", star );
            }
            j++;
        }
        printf( "%s", "\n" );
        i += 1;
    }
    return 0;
}
```

### 5.2.3. บันทึกผลลัพธ์การทำงานของโค้ดโปรแกรมนี้

ผลลัพธ์การทำงานของโปรแกรม

-- Show 'Hello World' to Screen 7 line (5)

```
[0] Hello World
[01] Hello World
[02] Hello World
[03] Hello World
[04] Hello World
[05] Hello World
[06] Hello World
```

```
*****
*****
*****
*****
*****
*****
*****
```

### 6. สรุปผลการปฏิบัติงาน

7. ผู้เรียนได้เรียนรู้เกี่ยวกับขั้นตอนการเขียนโค้ดโปรแกรมแบบ Codex ได้เรียนรู้ว่าควรใช้เครื่องหมายอะไรและใช้วิธีใดในการตรวจสอบความถูกต้องของโค้ด

### 7. คำถามทางการทดลอง

7.1. จงบอกจุดผิดพลาดที่ทำให้มักเกิดสิ่งผิดปกติภายในโค้ดโปรแกรมบ่อยที่สุด พร้อมแนวทางการแก้ไขปัญหาที่เกิดขึ้น

การเว้นวรรค, ตัวพิมพ์ใหญ่-เล็ก, เครื่องหมายต่างๆ  
ตรวจสอบโค้ดในคอมไพเลอร์

7.2. จงบอกประโยชน์ของมาตรฐานการเขียนโค้ดโปรแกรมแบบ Codex พร้อมยกตัวอย่างประกอบ

ทำให้โค้ดดูง่ายต่อการอ่านและถูกจัดรูปแบบ

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

```
int main() {
```

```
    int i, j = 0;
```

```
    int N = 7;
```

```
    char star = '*';
```