

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. บอกคำสั่งที่ใช้ในการรับค่าและการแสดงผล
- 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”

เป็นภาษาซี ใช้สำหรับคอมพิวเตอร์ขนาดเล็ก... ที่พัฒนาขึ้น  
ในทางวิศวกรรม มีลักษณะการเขียนโปรแกรมเชิงโครงสร้าง ถูกพัฒนา  
ในปี ค.ศ. 1969-1973 โดย Dennis Ritchie มีลักษณะการทำงาน  
ร่วมกับตัวแปล คอมไพเลอร์ คำสั่งเงื่อนไข การเขียนให้ตัวแปลทำงาน  
การกำหนดโครงสร้างของข้อมูลในรูปแบบต่างๆ ที่พัฒนาขึ้น  
มาอีกคือ ตัวแปลตัวอื่น



4.2. จงบอกและอธิบายโครงสร้างของภาษาซีโดยทั่วไป

1.2. จงเขียนโปรแกรม C เพื่อแสดงข้อความดังต่อไปนี้

```
#include <stdio.h> ส่วนการนำเข้าชุดคำสั่งพื้นฐานต่าง ๆ
#include <conio.h> ส่วนการนำเข้าชุดคำสั่งพื้นฐานต่าง ๆ
int main() { ส่วนของโปรแกรมการทำงานหลัก (โปรแกรมหลัก)
    printf("Hello world"); คำสั่ง printf อยู่ภายใต้โครงสร้างของฟังก์ชันหลัก
    return 0; คำสั่ง return อยู่ภายใต้โครงสร้างของฟังก์ชันหลัก
} // end function ปิดท้ายโปรแกรมการทำงานหลัก
```

4.3. จงบอกและอธิบายลักษณะการใช้งานคำสั่ง printf พร้อมยกตัวอย่างประกอบ

สำหรับ การแสดงค่าโดยให้ภาษาซี โดยรูปแบบของคำสั่งจะประกอบด้วย  
ตัว printf ตามด้วยเครื่องหมายวงเล็บเปิด ("แสดงเนื้อหา")" ตามด้วย  
เครื่องหมายวงเล็บปิด และเครื่องหมายอัฒภาค (;) โดยที่เครื่องหมายวงเล็บปิด และ  
เครื่องหมายอัฒภาคจะขึ้นบรรทัดใหม่ และตามด้วย printf("Hello worle");

4.4. จงบอกและอธิบายลักษณะการใช้งานคำสั่ง scanf พร้อมยกตัวอย่างประกอบ

เพื่อทำซ้ำในกรณีรับค่าจากแป้นพิมพ์ของตัว i ซ้ำวน โดยวิธีที่ 2 จะวนไป  
scanf จากตัว a จะวนกลับวนไปวนมาวนการรับค่าที่ 1 ของตัว a 2 ส่วนสุดท้าย  
คือส่วนของตัว b แทนที่ แทนค่าของตัว a และตัว b  
scanf ("%d", &a);

4.5. จงบอกและอธิบายการจัดโครงสร้างของโค้ดโปรแกรมตามมาตรฐาน Codex พร้อมยกตัวอย่าง

ประกอบ

[illegible]

4.6. จงบอกและอธิบายการตั้งชื่อตัวแปรตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

១. បង្កើតកំណត់ត្រាសម្រាប់រកចំនួននិស្សិតដែលបានចុះឈ្មោះ  
 ២. ប្រើប្រាស់រូបមន្តដើម្បីគណនាចំនួននិស្សិតដែលបានចុះឈ្មោះ  
 ៣. ប្រើប្រាស់រូបមន្តដើម្បីគណនាចំនួននិស្សិតដែលបានចុះឈ្មោះ  
 Calculator for student count loop



4.7. จงบอกและอธิบายการนิยามตัวแปรและฟังก์ชันตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

จะสร้างชื่อตัวแปรภาษาอังกฤษ ตัวชื่อตามชื่อโคธแรก แล้ว  
 จอจินห์ ตัวอยู่ก่อนตัวแรกจะหาค่าใช้จากในส่วนของ...  
 เช่น ชื่อของโปรแกรมย่อย calculator\_find\_maximum value()

4.8. จงบอกและอธิบายการเว้นวรรคตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

จะสร้างบรรทัดตอนในภาคข้อควรระวังของมาตรฐาน  
 โปรแกรม อย่างเป็น int = 0, sum = 0, n = 0;  
 printf ("Hello world");

4.9. จงบอกและอธิบายการใช้งานเครื่องหมายวงเล็บตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

เมื่อวงเล็บปิดกันระหว่างวงเล็บทั้ง 2 ให้เว้นวรรคหนึ่งครั้ง  
 เช่น int main() { เมื่อเคสของนาย { ให้ใส่ชื่อสมาชิก  
 ชื่อคน Top นึงคือ เช่น if (num1 < num2) {  
 num1 = num2 ;

## 5. ลำดับขั้นตอนการปฏิบัติงาน

### 5.1. การรับและการแสดงผล

5.1.1. จงเขียนโปรแกรมเพื่อรับตัวเลขจำนวนเต็ม ตัวเลขทศนิยม และตัวอักษร เพื่อแสดงผลบนหน้าจอคอมพิวเตอร์

| Test case 1                                | Test Case 2                            |
|--|--|
| Input<br>Please enter value : 5 12.23945 F | Input<br>Please enter value : 9 13.1 M |
| Output<br>5 12.24 F                        | Output<br>9 13.10 M                    |



### 5.1.2. บันทึกโค้ดโปรแกรมดังกล่าว

| ผังงาน  | โค้ดโปรแกรม   |
|---|---|
| <pre> graph TD     Start([start]) --&gt; Input[/Please Enter Value :/]     Input --&gt; Decl[/int num1, float num2, char abc ;/]     Decl --&gt; Print[/printf num1, num2, abc/]     Print --&gt; Stop([stop])         </pre> | <pre> #include &lt;stdio.h&gt; #include &lt;conio.h&gt;  int main() {     int num1 ;     float num2 ;     char abc ;     printf("Please Enter Value:");     scanf("%d %f %c", &amp;num1, &amp;num2, &amp;abc);     printf("%d %f %c", num1, num2, abc);     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

โค้ดโปรแกรม

```
#include <stdio.h>

int main() {
    int i = 0;
    int N = 7;
    printf("--I show 'Hello World' to screen %d line(s) \n", N);

    for (i = 0; i < N; i++) {
        printf("[%3d] Hello World \n", i);
    }

    char star = '*';
    int j = 0;
    N = 8;
    i = 0;
    j = 0;
    printf("\n");
    while (i < N) {
        j = 0;
        while (j < N) {
            if (i < j)
                printf("%d", star);
            j++;
        }
        printf("%s", "\n");
        i += 1;
    }
}
```



### 5.2.3. บันทึกผลลัพธ์การทำงานของโค้ดโปรแกรมนี้

#### ผลลัพธ์การทำงานของโปรแกรม

-- I show 'Hello World' to screen 7 line(s)

```
[0] Hello World
[1] Hello World
[2] Hello World
[3] Hello World
[4] Hello world
[5] Hello world
[6] Hello world
```

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

#### 6. สรุปผลการปฏิบัติงาน

- การเขียน code ที่ไม่ใช้มาตรฐาน code x อาจทำให้ยาก  
ไม่สามารถเขียนต่อได้ ทำให้คนที่เขียน code ต่อลำบาก
- การเขียน code ตามมาตรฐานของ code x จะช่วยลด  
ข้อผิดพลาด ทำให้คนที่เขียน code ต่อทำงานได้ง่ายขึ้น

#### 7. คำถามทางการทดลอง

7.1. จงบอกจุดผิดพลาดที่ทำให้มักเกิดสิ่งผิดปกติภายในโค้ดโปรแกรมบ่อยที่สุด พร้อมแนวทางการแก้ไขปัญหานั้น

จุดผิดพลาดบ่อยที่สุด คือ ( ) และ { }  
เพราะการแก้ไขไม่ตรงจุด ทำให้เครื่องอ่าน ( ) และ { } ไม่เป็น  
เพราะข้อผิดพลาดในวงเล็บ

7.2. จงบอกประโยชน์ของมาตรฐานการเขียนโค้ดโปรแกรมแบบ Codex พร้อมยกตัวอย่างประกอบ  
ดูง่าย เข้าใจง่าย จุดคือที่เขียน code ต่อสามารถ  
เขียนต่อได้ มีความสะอาด สวยงาม