

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”

เป็นภาษาซีสำหรับคอมพิวเตอร์ ไม่เหมือน คอมไพเลอร์ ที่มีคอมไพเลอร์ในการพัฒนา
แบบ มีลักษณะการเขียนโปรแกรมเชิงโครงสร้าง ถูกพัฒนามาในช่วงปี ค.ศ 1969-1973
โดยเดนนิส ริทซ์ มีจุดประสงค์เพื่อใช้ในการสอนวิชาคอมพิวเตอร์ในมหาวิทยาลัย
ที่มีขนาดเล็กและไม่มีคอมพิวเตอร์ และเหมาะสำหรับการพัฒนาที่มีขนาดเล็ก
อันเนื่องมาจากในคอมพิวเตอร์

4.2. จงบอกและอธิบายโครงสร้างของภาษาซีโดยทั่วไป

ภาษาซีเป็นภาษาซีไพล์เป็นส่วนใหญ่ในการเริ่มต้นให้กับการพัฒนาซอฟต์แวร์
เพื่อใช้ในการทดลองและพัฒนาโปรแกรม

4.3. จงบอกและอธิบายลักษณะการใช้งานคำสั่ง printf พร้อมยกตัวอย่างประกอบ
ให้สั้นรัดกุม `printf(" ");`

4.4. จงบอกและอธิบายลักษณะการใช้งานคำสั่ง scanf พร้อมยกตัวอย่างประกอบ
ให้สั้นรัดกุม `scanf("%s", &name);`

4.5. จงบอกและอธิบายการจัดโครงสร้างของโค้ดโปรแกรมตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

การตั้งชื่อตัวแปร จะต้องตั้งชื่อตัวแปรตามกฎ (เช่น ห้ามมีเครื่องหมายพิเศษ) ให้เป็นชื่อที่สื่อความ
ถึงตัวแปรนั้น เป็นลักษณะที่สื่อความหมาย การตั้งชื่อตัวแปรให้มีความหมาย
ชัดเจน () เช่น AA_BB

4.6. จงบอกและอธิบายการตั้งชื่อตัวแปรตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ
การตั้งชื่อให้มีความหมาย และใช้ตัวอักษรพิมพ์เล็ก การใช้ในลักษณะ
เช่น ชื่อของตัวแปรเป็นชื่อ calculator_for_student

4.7. จงบอกและอธิบายการนิยามตัวแปรและฟังก์ชันตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

เราต้องรู้ชื่อตัวแปรที่เรากำลังใช้เพื่อที่จะใช้มันได้ถูกต้อง
การใช้งานในส่วนนี้ เช่น ชื่อของโปรแกรม

4.8. จงบอกและอธิบายการเว้นวรรคตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

เราต้องรู้ชื่อตัวแปรที่เรากำลังใช้เพื่อที่จะใช้มันได้ถูกต้อง
การใช้งานในส่วนนี้ เช่น ชื่อของโปรแกรม

4.9. จงบอกและอธิบายการใช้งานเครื่องหมายวงเล็บตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

เราต้องรู้ชื่อตัวแปรที่เรากำลังใช้เพื่อที่จะใช้มันได้ถูกต้อง
การใช้งานในส่วนนี้ เช่น ชื่อของโปรแกรม

5. ลำดับขั้นการปฏิบัติงาน

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]) --> Input[/please enter value/] Input --> Decl[/int a, float b, char c/] Decl --> Print[/printf a,b,c/] Print --> Stop([stop]) </pre>	<pre> #include <stdio.h> int main() { int a; float b; char c; printf("Please enter value: "); scanf("%d %f %c", &a, &b, &c); printf("a = %d, b = %f, c = %c", a, b, c); return 0; } </pre>

5.2 มาตรฐานการเขียนโค้ดโปรแกรมตามมาตรฐาน Codex

5.2.1. จากโค้ดโปรแกรมต่อไปนี้ จงจัดระเบียบให้อยู่ภายใต้มาตรฐานการเขียนโปรแกรมแบบ Codex พร้อมทั้งแก้ไขข้อผิดพลาด (Bug) เพื่อให้โปรแกรมสามารถทำงานได้ตามปกติ

โค้ดโปรแกรม
<pre> #include <stdio.h> int main() { int i=0; int N=7; printf("— 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("%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("[%d] 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++;
    }
}
```

5.2.3. บันทึกผลลัพธ์การทำงานของโค้ดโปรแกรมนี้

ผลลัพธ์การทำงานของโปรแกรม
<pre>--!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 *</pre>

6. สรุปผลการปฏิบัติงาน

.....

7. คำถามทางการทดลอง

7.1. จงบอกจุดผิดพลาดที่ทำให้มักเกิดสิ่งผิดปกติภายในโค้ดโปรแกรมบ้อยที่สุด พร้อมแนวทางการแก้ไขที่เกิดขึ้น

.....

7.2. จงบอกประโยชน์ของมาตรฐานการเขียนโค้ดโปรแกรมแบบ Codex พร้อมยกตัวอย่างประกอบ

.....

