

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"
ภาษาซีเป็นภาษาที่ใช้สำหรับการพัฒนาโปรแกรมคอมพิวเตอร์
ที่มีความยืดหยุ่นในการพัฒนาโปรแกรม มีความปลอดภัยสูง
ถูกพัฒนามาในช่วงปี ค.ศ. 1969-1973 โดยเดนนิส ริตซ์
ดร. เจฟฟรีย์ โทมัส เรโนลด์ส และ ออกแบบมาให้นักคอมพิวเตอร์
ใช้เพื่อ การจัดการกับข้อมูล การจัดการกับข้อมูล
.....
.....
.....

4.2. จงบอกและอธิบายโครงสร้างของภาษาซีโดยทั่วไป

โครงสร้างของภาษาซี มี 3 ส่วน โครงสร้างโปรแกรม ดังนี้
1. ส่วนหัวของโปรแกรม ใช้เป็นข้อความเพื่อเตรียมการใดก่อนการประมวลผลโปรแกรม
2. ส่วนฟังก์ชันหลัก คือ ฟังก์ชัน `main` ซึ่งโปรแกรมจะทำงานที่ฟังก์ชันนี้
3. ส่วนตัวโปรแกรม คือ ส่วนที่เขียนคำสั่งเพื่อใช้โปรแกรม
ทำงานตามที่ต้องการ

4.3. จงบอกและอธิบายลักษณะการใช้งานคำสั่ง printf พร้อมยกตัวอย่างประกอบ

คำสั่ง `printf` ใช้สำหรับพิมพ์ข้อความไปยังหน้าจอ (output) รูปแบบคำสั่งคือ `printf("ข้อความ");`
ภาษาซีมีฟังก์ชันสำหรับอ่านค่าข้อมูลจากแป้นพิมพ์ (input) ซึ่งใช้คำสั่ง `scanf`
ดังนี้ `scanf("รูปแบบข้อมูล", &ตัวแปร);` เช่น `printf("Hello world");`

4.4. จงบอกและอธิบายลักษณะการใช้งานคำสั่ง scanf พร้อมยกตัวอย่างประกอบ

`scanf` คือ คำสั่งที่ใช้สำหรับอ่านค่าข้อมูลจากแป้นพิมพ์ (input) รูปแบบคำสั่งคือ `scanf("รูปแบบข้อมูล", &ตัวแปร);`
ลักษณะการใช้งาน `scanf` เช่น `scanf("%d", &num);` ซึ่ง `%d` คือ รูปแบบข้อมูล (Place holder)
และ `num` คือ ตัวแปร (variable) ที่ใช้เก็บค่า เช่น `scanf("%d", &num);`

4.5. จงบอกและอธิบายการจัดโครงสร้างของโค้ดโปรแกรมตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

ตัวอย่างการจัดโครงสร้างโปรแกรมตามมาตรฐาน Codex เช่น
`int main() {`
 `int i = 0; // เพื่อคำนวณหาผลรวมของตัวเลข 1 ถึง 10`
 `int ans = 0;`

4.6. จงบอกและอธิบายการตั้งชื่อตัวแปรตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

การตั้งชื่อตัวแปรตามมาตรฐาน Codex มีดังนี้
`case char const continue` เป็นต้น
การตั้งชื่อตัวแปรต้องปฏิบัติตามกฎต่อไปนี้ 1. ชื่อตัวแปรต้องเป็น Lower case 2. case 2 case
Lower case calculator_count = 0; case 1. in calculator_count = 100;

5.1.2. บันทึกโค้ดโปรแกรมดังกล่าว

ผังงาน	โค้ดโปรแกรม
<pre> graph TD Start([เริ่ม]) --> Input[รับค่า] Input --> Process[ประมวลผล] Process --> Output[แสดงผล] Output --> End([จบ]) </pre>	<pre> #include <stdio.h> int main() { char Text; float Decimal; printf("Please enter value:"); scanf("%f %c", &Decimal, &Text); printf("%f %c", Decimal, Text); 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 ("==> show 'Hello world' to screen %d lines in", 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 ("%c", star); j++;
        }
        printf ("%s", "\n"); i++;
    }
    return
```

}

5.2.3. บันทึกผลลัพธ์การทำงานของโค้ดโปรแกรมนี้

ผลลัพธ์การทำงานของโปรแกรม	
-- show 'Hello World' to serce 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

ใช้โปรแกรมแก้ไขโค้ดโปรแกรมแบบ Codex ทำให้เราเห็นโค้ดโปรแกรมไว้แก้ไข
แก้ไขผิด

7. คำถามทางการทดลอง

7.1. จงบอกจุดผิดพลาดที่ทำให้เกิดสิ่งผิดปกติภายในโค้ดโปรแกรมบ้อยที่สุด พร้อมแนวทางการแก้ไขปัญหานั้น

การให้บรรทัดไม่ได้เหมือนกัน เราต้องให้เป็นโปรแกรมให้เหมือนกัน
คือให้บรรทัดให้เหมือนกัน

7.2. จงบอกประโยชน์ของมาตรฐานการเขียนโค้ดโปรแกรมแบบ Codex พร้อมยกตัวอย่างประกอบ

คือ การที่โปรแกรมเมอร์ทุกคนในทีมสามารถใช้โค้ดโปรแกรมเดียวกันได้
เช่น การที่โปรแกรมเมอร์ทุกคนในทีมสามารถใช้โค้ดโปรแกรมเดียวกันได้