

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. พิกัดและทดลองใช้คำสั่งรับค่าและแสดงผล
- 2.2.1. บอกและอธิบายการจัดโครงสร้างของงาน
- 2.2.2. บอกและอธิบายการจัดโครงสร้างของโค้ดโปรแกรม
- 2.2.3. พิกัดและทดลองใช้จัดโครงสร้างของโค้ดโปรแกรมเพื่อให้ทำงานได้มีประสิทธิภาพมากขึ้น
- 2.2.4. ออกแบบแนวทางการใช้จัดโครงสร้างของโค้ดโปรแกรมอย่างเป็นระบบ
- 2.2.5. แนะนำแนวทางการใช้จัดโครงสร้างของโค้ดโปรแกรม
- 2.2.6. บอกและอธิบายการตั้งชื่อตัวแปรให้สอดคล้องกับโครงสร้างหลัก
- 2.2.7. พิกัดและทดลองใช้การตั้งชื่อตัวแปรให้สอดคล้องกับโครงสร้างหลัก
- 2.2.8. บอกและอธิบายการนิยามความหมายของตัวแปรและฟังก์ชัน
- 2.2.9. พิกัดและทดลองใช้การนิยามความหมายของตัวแปรและฟังก์ชัน
- 2.2.10. บอกและอธิบายการเว้นวรรคตอน
- 2.2.11. พิกัดและทดลองใช้การเว้นวรรคตอน
- 2.2.12. บอกและอธิบายการใช้เครื่องหมายวงเล็บ
- 2.2.13. พิกัดและทดลองใช้การใช้เครื่องหมายวงเล็บ

3. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่องที่ติดตั้งโปรแกรม Dev-C

4. ทฤษฎีการทดลอง

4.1. จงบอกและอธิบายที่มาและลักษณะพิเศษของ “ภาษาซี”

ภาษาซี ถูกพัฒนาขึ้นเมื่อปี ค.ศ. 1969-1970
โดยนักวิจัยของบริษัทเบลล์แล็บส์
ซึ่งมีจุดประสงค์เพื่อใช้ในการพัฒนาโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์
ที่มีขนาดเล็กและมีความเร็วสูง
ภาษาซีจึงถูกพัฒนาขึ้นโดยบริษัทเบลล์แล็บส์

4.2. จงบอกและอธิบายโครงสร้างของภาษาซีโดยทั่วไป

ภาษาซีประกอบด้วยส่วนประกอบหลัก ๆ ดังนี้
1. preprocessor directives (คำสั่งก่อนการคอมไพล์)
2. include (การรวมไฟล์)
3. global variable (ตัวแปรระดับโลก)
4. function (ฟังก์ชัน)
5. main function (ฟังก์ชันหลัก)

4.3. จงบอกและอธิบายลักษณะการใช้คำสั่ง printf พร้อมยกตัวอย่างประกอบ

printf เป็นคำสั่งที่ใช้สำหรับส่งข้อมูลไปยังหน้าจอ
คำสั่ง printf มีรูปแบบการใช้งานดังนี้
1. รูปแบบทั่วไป: printf (format, argument);
2. format: เป็นสตริงที่กำหนดรูปแบบการแสดงผลของข้อมูล
3. argument: เป็นข้อมูลที่จะถูกส่งไปยังหน้าจอ

ตัวอย่างการใช้งาน printf
1. การพิมพ์ข้อความธรรมดา: printf ("Hello World");
2. การพิมพ์ตัวเลข: printf ("The value of x is %d", x);
3. การพิมพ์ค่าจริง: printf ("The value of x is %f", x);
4. การพิมพ์ค่าตัวอักษร: printf ("The value of x is %c", x);
5. การพิมพ์ค่าสตริง: printf ("The value of x is %s", x);

4.5. จงบอกและอธิบายการใช้โครงสร้างของโปรแกรมตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

โครงสร้างของโปรแกรมมาตรฐาน Codex ประกอบด้วย
1. preprocessor directives (คำสั่งก่อนการคอมไพล์)
2. include (การรวมไฟล์)
3. global variable (ตัวแปรระดับโลก)
4. function (ฟังก์ชัน)
5. main function (ฟังก์ชันหลัก)

4.6. จงบอกและอธิบายการตั้งชื่อตัวแปรตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

การตั้งชื่อตัวแปรตามมาตรฐาน Codex มีดังนี้
1. ชื่อตัวแปรต้องเป็นภาษาอังกฤษ
2. ชื่อตัวแปรต้องเป็นคำเดียวหรือหลายคำที่คั่นด้วยขีดล่าง
3. ชื่อตัวแปรต้องไม่ซ้ำกับชื่อฟังก์ชันหรือชื่อตัวแปรอื่น
4. ชื่อตัวแปรต้องไม่ขึ้นต้นด้วยตัวเลข
5. ชื่อตัวแปรต้องไม่ขึ้นต้นด้วยตัวอักษรพิเศษ

4.7. จงบอกและอธิบายการนิยามตัวแปรและฟังก์ชันตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

จะต้องนิยามชื่อตัวแปรก่อน ต้องใส่ชื่อให้ตรงกับชื่อโครงงาน แล้วจึงนำชื่อไปใช้จริงได้
ในชื่อหน้า เช่น ชื่อของตัวแปรคือ student count loop

4.8. จงบอกและอธิบายการเว้นวรรคตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ
เมื่อเว้นวรรคตอนเพื่อเขียนโปรแกรมให้ดูง่ายขึ้น
โปรแกรมอย่างเช่น student count loop

printf("Hello World");

4.9. จงบอกและอธิบายการใช้งานเครื่องหมายวงเล็บตามมาตรฐาน Codex พร้อมยกตัวอย่างประกอบ

- เมื่อใช้เครื่องหมายวงเล็บต้องใส่ชื่อตัวแปรให้ชัดเจน
- เมื่อใช้เครื่องหมายวงเล็บต้องใส่ชื่อตัวแปรให้ชัดเจน
ตัวอย่าง Tap 1 คือชื่อของตัวแปร if (C Max < Count) {
Tap Main = Count;

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. บันทึกโค้ดโปรแกรมหาดังกล่าว

ผังงาน	โค้ดโปรแกรม
<pre> graph TD Start([start]) --> Input[/Please Enter Value:/] Input --> Decl[/int q1, float q2, char q3;/] Decl --> Print[/printf q1, q2, q3/] Print --> Stop([stop]) </pre>	<pre> #include <stdio.h> #include <conio.h> int main() { int q1; float q2; char q3; printf("Please enter value: "); scanf("%d %f %c", &q1, &q2, &q3); printf("%d %f %c", q1, q2, q3); 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 lines\n", N);
    for (i = 0; i < N; i++) {
        printf("[%d] Hello World\n", i);

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

5.2.3. บันทึกผลลัพธ์การทำงานของโค้ดโปรแกรมนี้

ผลลัพธ์การทำงานของโปรแกรม

```
-- I show 'Hello world' to screen 7 lines)
[0] Hello World
[1] Hello World
[2] Hello World
[3] Hello World
[4] Hello World
[5] Hello World
[6] Hello World
*****
*****
*****
*****
6. สรุปผลการปฏิบัติงาน
**
```

สามารถรัน Code online ที่ Codex ได้

7. คำถามทางการทดลอง

7.1. จงบอกจุดผิดพลาดที่ทำให้มักเกิดสิ่งผิดปกติภายในโค้ดโปรแกรมบ้อยที่สุด พร้อมแนวทางการแก้ไขปัญหาที่เกิดขึ้น

จุดที่ผิดพลาดบ่อยที่สุดคือ () ของเลขเปิด-ปิด { }

อีกที่คือ - ปิด

จุดที่ควรระวังเป็นพิเศษ (คือช่องว่าง "C" ของเลขเปิด-ปิด { })

จงเก็บไว้ให้ดี

7.2. จงบอกประโยชน์ของมาตรฐานการเขียนโค้ดโปรแกรมแบบ Codex พร้อมยกตัวอย่างประกอบ

ช่วยให้ Code อ่านง่ายขึ้น ทำให้ได้งานง่ายขึ้น

Code อ่านง่าย-ข้อผิดพลาดน้อย-ได้งานง่ายขึ้น

สามารถทำงานร่วมกันกับคนอื่น