ใบความรู้ที่4

หลักการแก้ปัญหา

**โครงสร้างภาษา  C**

ภาษา C เป็นโปรแกรมภาษาระดับสูง ถูกพัฒนาขึ้นในปี 1972 ที่ AT&T Bell Lab เราสามารถใช้ภาษา C มาเขียนเป็นคำสั่งต่างๆ ที่คอมพิวเตอร์สามารถเข้าใจได้ และกลุ่มของคำสั่งเหล่านี้ เราก็เรียกกันว่า อัลกอริธึม ได้มีผู้ให้คำจำกัดความของคำว่า อัลกอริธึม ว่าเป็น “A precise description of a step-by-step process that is guaranteed to terminate after a finite number of steps with a correct answer for every particular instance of an algorithmic problem that may occur.” สำหรับ Compiler ภาษา C ที่มีในปัจจุบัน มี 2 ค่ายใหญ่ๆ ที่มีผู้คนสนใจใช้กันมากได้แก่ Microsoft และ Borland

โครงสร้างภาษา  C  แบ่งออกได้เป็น  5  ส่วนดังต่อไปนี้

1.       พรีโพรเซสเซอร์  (Preprocessor  directives)

ส่วนนี้โปรแกรมต้องมีจะใช้สำหรับเรียกไฟล์ที่โปรแกรมต้องการในการทำงานและกำหนดค่าต่าง ๆ สำหรับไดเร็กทีฟ  ที่ใช้กันบ่อย ๆ ได้แก่

# include  เป็นการแจ้งให้คอมไพเลอร์อ่านไฟล์อื่นเข้ามาคอมไพล์ร่วมด้วย  รูปแบบการใช้จะทำโดยเขียน #include  แล้วตามด้วยชื่อไฟล์  ดังนี้

#include  “stdio.h”  หมายความว่า  อ่านไฟล์  stdio.h  เข้ามาด้วย

#include  “conio.h”  หมายความว่า  อ่านไฟล์  conio.h  เข้ามาด้วย

#define  เป็นการกำหนดค่านิพจน์ต่าง ๆ ให้กับชื่อของตัวแปร  โดยมีรูปแบบดังนี้

          #define  END  20;

          #define  A  5\*6+3;

2.       ส่วนการกำหนดค่า  (Global  declarations)

ส่วนนี้จะให้ในการประกาศตัวแปรหรือฟังก์ชันที่ต้องใช้ในโปรแกรม  โดยทุก ๆ ส่วนของโปรแกรมสามารถจะเรียกใช้ข้อมูลที่ประกาศไว้ในส่วนนี้ได้

3.       ส่วนฟังก์ชันหลัก  (The main()  function)

ส่วนนี้ทุกโปรแกรมจะต้องมี  ซึ่งจะประกอบไปด้วยประโยคคำสั่งต่าง ๆ ที่จะให้โปรแกรมทำงานโดยนำคำสั่งต่าง ๆ มาเรียงต่อกัน  และแต่ละประโยคคำสั่งจะจบด้วยเครื่องหมายเซมิโคลอน ;  โดยโปรแกรมหลักนี้จะเริ่มต้นด้วย  **main()**ตามด้วยเครื่องหมายปีกกาเปิด  {  และจบด้วยเครื่องหมายปีกกาปิด }

4.       การสร้างฟังก์ชันและการใช้ฟังก์ชัน  (Uses-defined function)

เป็นการเขียนคำสั่งและฟังก์ชันต่าง ๆ ขึ้นใช้ในโปรแกรม  โดยต้องอยู่ในเครื่องหมาย  {   }  และต้องสร้างฟังก์ชันหรือคำใหม่ที่ให้ทำงานตามที่เราต้องการให้กับโปรแกรมและสามารถเรียกใช้ได้

ตัวอย่างเช่น

#include  <stdio.h>

main()

{

          function();   /\*เรียกใช้ฟังก์ชันที่สร้างขึ้น\*/

}

function();             /\*สร้างฟังก์ชันใหม่  โดยให้ชื่อว่า function\*/

{

          return;         /\*คืนค่าที่เกิดจากการทำฟังก์ชัน\*/

}

5.       ส่วนอธิบายโปรแกรม  (Program  comment)

ส่วนนี้ใช้เขียนคอมเมนต์โปรแกรม  เพื่ออธิบายการทำงานต่าง ๆ ทำให้ผู้ศึกษาโปรแกรมในภายหลังทำความเข้าใจโปรแกรมได้ง่ายขึ้น  เมื่อคอมไพล์โปรแกรมส่วนนี้จะถูกข้ามไป

คอมเม้นท์ (Comment) มีอยู่สองแบบ

1. คอมเม้นท์แบบบรรทัดเดียวจบ (Single line comment)

            ข้อความที่ตามหลังเครื่องหมาย // ในบรรทัดนั้นทั้งบรรทัด จะไม่ถูกประมวลผลคอมพิวเตอร์จะเริ่มประมวลผลโค๊ดใหม่ในบรรทัดถัดไป

2. คอมเม้นท์แบบหลายบรรทัด (Multiple line comment)

              เริ่มต้นด้วยเครื่องหมาย  /\*  และจบด้วยเครื่องหมาย \*/  ข้อความที่อยู่ระหว่างเครื่องหมายทั้งสองนี้ จะถือว่าเป็น Comment ทั้งหมด

**ตัวอย่าง**

#include  <stdio.h>

main()

{

// this is a comment and will not be shown in output.

printf( "This isn\'t a comment" ); // this is a single line comment

/\*

     This is a multiple line comment.

     Comment #1

     Comment #2

\*/

}

**ผลลัพธ์**     This isn't a comment

**ดังนั้นรูปแบบของการเขียนโปรแกรม จึงเขียนออกมาในรูปแบบดังนี้**

**// ข้อความที่อยู่ข้างหลังเครื่องหมาย // จะเป็นคำอธิบายโปรแกรม**

**#include**

**main()**

**{**

**constant declarations; //การกำหนดค่าคงที่ต่างๆ**

**variable declarations; // การกำหนดตัวแปรต่างๆ**

**executable statements; // คำสั่งการทำงานของโปรแกรม**

**}**