หน่วยที่ 1
ความรู้เกี่ยวกับการเขียนโปรแกรมภาษา C++

สาระการเรียนรู้

1.1 ภาษาซีพลัสพลัสเบื้องต้น

1.2 โครงสร้างคำสั่ง

1.3 ตัวแปรและชนิดข้อมูล

1.4 เครื่องหมายดำเนินการ

1.5 อินพุตและเอาต์พุต

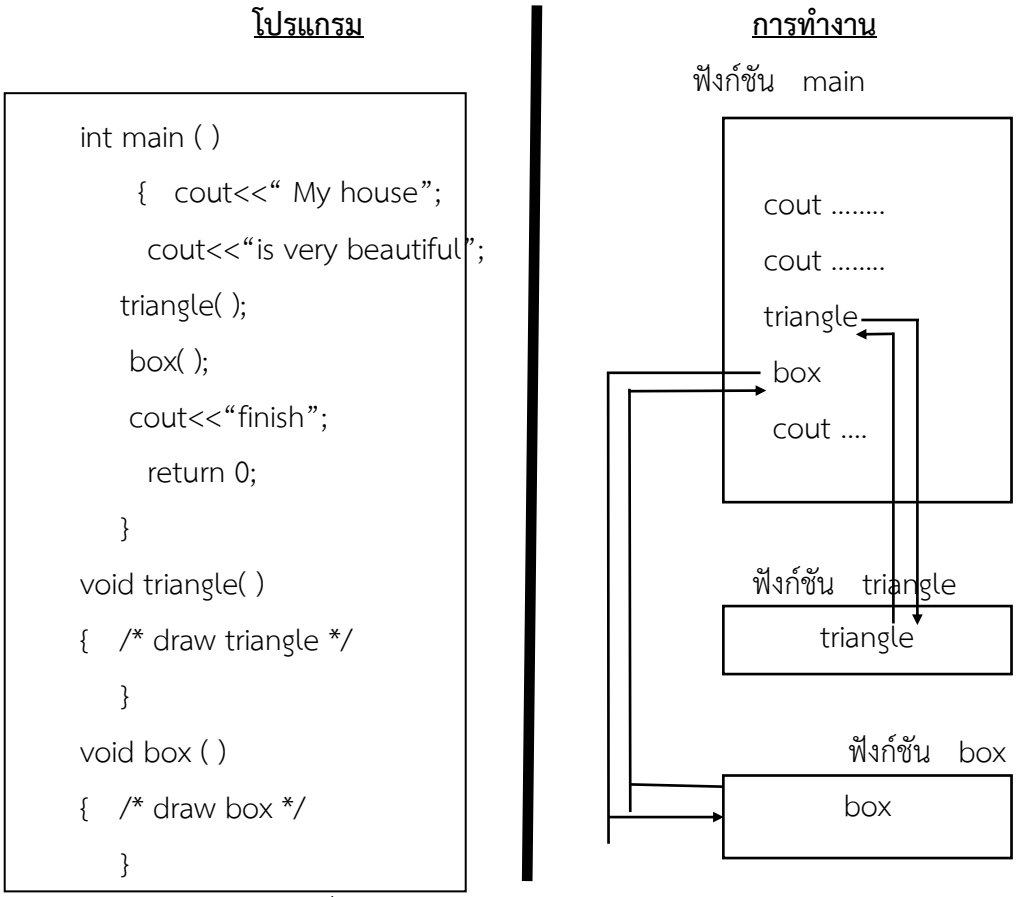
1.1 ภาษาซีพลัสพลัสเบื้องต้น (Structure of a C++ program)

ภาษาซีพลัสพลัส ถูกพัฒนาขึ้นมาเมื่อปี ค.ศ. 1983 ที่ Bell Laboratories โดย Bjarne Stroustrup ภาษานี้ถูกพัฒนาขึ้นมาจากภาษา C และมีการเพิ่มความสามารถของภาษา C++ด้วยการที่ทำให้สามารถเขียนเป็นโปรแกรมเชิงวัตถุ (Object-Oriented Programming) ซึ่งเพิ่มประสิทธิภาพในการพัฒนาโปรแกรมที่มีขนาดใหญ่และสามารถรองรับความซับซ้อนของการเขียนโปรแกรมของระบบงานในปัจจุบัน

ภาษาซีพลัสพลัส เป็นภาษาที่ทำงานได้อย่างกว้างขวาง เข้าใจง่าย เขียนง่าย ตลอดจนมีคำสั่งที่อำนวยความสะดวกให้กับโปรแกรมเมอร์ที่จะสามารถเรียกใช้ได้ตามที่ต้องการ ซึ่งโปรแกรมเมอร์ควรจะต้องศึกษาและทำความเข้าใจในกฎเกณฑ์เหล่านั้นให้ดีเสียก่อน ก็จะทำให้สามารถนำภาษาซีพลัสพลัสมาใช้งานได้อย่างมีประสิทธิภาพ กฎเกณฑ์ของเครื่องมือที่ใช้เขียนโปรแกรมภาษาซีพลัสพลัสของแต่ละผู้ผลิต จะมีข้อแตกต่างกันไปบ้างเล็กน้อย

1.1.1 รูปแบบการทำงานของภาษาซีพลัสพลัส

ในภาษาซีพลัสพลัสจะเขียนโปรแกรมโดยการเรียกใช้แต่ละชุดของโปรแกรมที่เรียกว่า **ฟังก์ชัน (Function)** หรือในโปรแกรมภาษาอื่นอาจจะเรียกว่า **โปรแกรมย่อย หรือชุดคำสั่งย่อย (Procedure)** นั่นเอง ฟังก์ชันเหล่านี้จะมีชื่ออะไรก็ได้ กี่ฟังก์ชันก็ได้ แต่อย่างน้อยต้องมี 1 ฟังก์ชันที่ชื่อ **main** เพื่อให้โปรแกรมเริ่มทำงานที่ฟังก์ชันนี้ ดังรูป



รูปที่ 1.1 แสดงรูปแบบการทำงานของภาษาซีพลัสพลัส

จากตัวอย่าง จะแสดงให้เห็นว่า ภาษาซีพลัสพลัสสามารถมีได้หลายฟังก์ชัน แต่จะมีฟังก์ชันหลัก คือ ฟังก์ชัน main ในการควบคุมการทำงานของโปรแกรมว่าให้ทำฟังก์ชันใดบ้าง แต่ถ้าโปรแกรมขนาดเล็กไม่มีการทำงานที่ซับซ้อน อาจจะไม่จำเป็นต้องมีฟังก์ชันอื่นๆ มีฟังก์ชัน main เพียงฟังก์ชันเดียวก็ได้

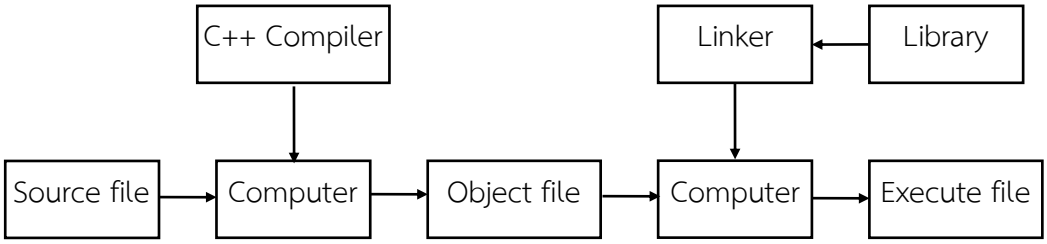
1.1.2 การคอมไพล์และลิงค์โปรแกรมในภาษาซีพลัสพลัส

การสร้างโปรแกรมที่สามารถใช้งานได้ขึ้นมาโปรแกรมหนึ่ง ในภาษาซีพลัสพลัสมีขั้นตอนดังนี้

1.1.2.1 สร้างตัวโปรแกรมที่เป็นตัวอักษร หรือเรียกว่า **ซอร์สไฟล์ (Source file)** โดยมีนามสกุลเป็น .cpp ขึ้นมาก่อน โดยใช้โปรแกรมที่สามารถเขียนไฟล์ที่เก็บอักขระ (Editor) ใดๆ ก็ได้ อักษรหรืออักขระใดๆ นั้น จะต้องอยู่ในรูปแบบของการโปรแกรมภาษา (ขั้นตอนนี้คือการสร้างโปรแกรมที่เป็นภาษามนุษย์นั่นเอง)

1.1.2.2 คอมไพล์เลอร์ของภาษาซีพลัสพลัส (C++ Compiler) จะทำการแปลงซอร์สไฟล์จากอักขระใดๆ ให้เป็นรหัสที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้เก็บไว้ในอีกไฟล์หนึ่งเรียกว่าไฟล์วัตถุประสงค์ (Object file) ที่มีนามสกุล .obj (ขั้นตอนนี้เรียกว่า **การคอมไพล์** เป็นการแปลงภาษามนุษย์เป็นภาษาเครื่องนั่นเอง)

1.1.2.3 ตัวเชื่อม (Linker) จะทำการตรวจสอบว่าในโปรแกรมที่เขียนขึ้นนั้น มีการเรียกใช้งานฟังก์ชันมาตรฐานใด จากห้องสมุดของภาษาซีพลัสพลัส (C++ Library) บ้างหรือไม่ ถ้ามี ตัวเชื่อมจะทำการรวมเอาฟังก์ชันเหล่านั้นเข้ากับไฟล์วัตถุประสงค์ แล้วจะได้ไฟล์ที่สามารถทำงานได้ โดยมีนามสกุลเป็น .exe (ขั้นตอนนี้เรียกว่า **การลิงค์** เป็นการรวมฟังก์ชันสำเร็จรูปเข้าไป แล้วสร้างไฟล์ที่ทำงานได้)



รูปที่ 1.2 แสดงขั้นตอนการคอมไพล์และลิงค์โปรแกรมภาษาซีพลัสพลัส

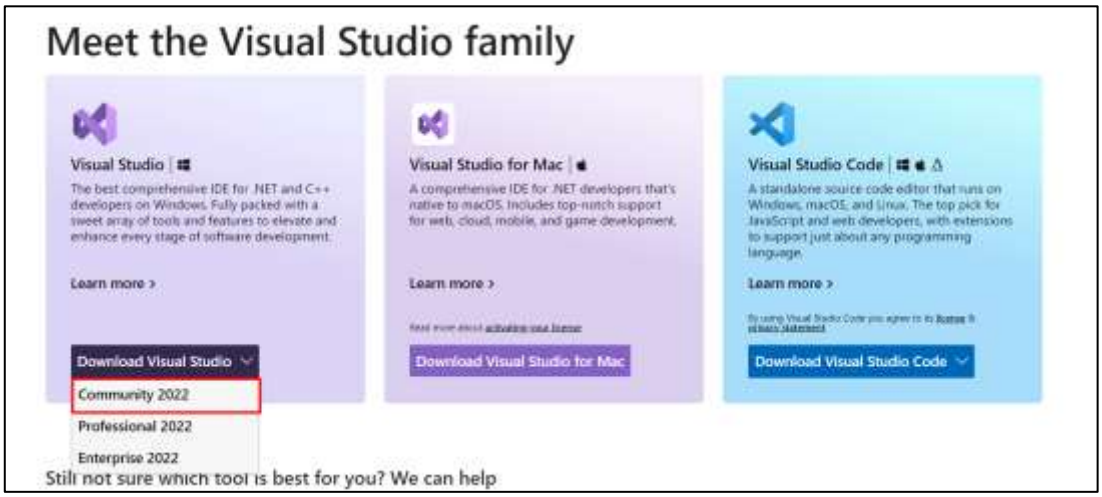
1.1.3 การสร้างโปรแกรมภาษาซีพลัสพลัสด้วย Microsoft visual studio

การเขียนโปรแกรมคอมพิวเตอร์ภาษาต่าง ๆ ปกติจะเขียนคำสั่งต่าง ๆ ด้วยโปรแกรมที่เรียกว่า editor เช่น notepad แล้วเปลี่ยนภาษาที่เขียนนั้นเป็นภาษาเครื่องโดยใช้ตัวแปลภาษา (compiler) ของภาษาที่ใช้เขียนโปรแกรมคอมพิวเตอร์ เช่น ตัวแปลภาษาของภาษาซีพลัสพลัส ซึ่งก็มีผู้ผลิตหลายราย ซึ่งต้องเป็นไปตามมาตรฐาน ANSI C++ และมีรายละเอียดเพิ่มเติมแตกต่างกันไปในปัจจุบัน เครื่องมือที่ช่วยในการพัฒนาโปรแกรม ภาษาอังกฤษเรียกว่า IDE (Integral Development Environment) ซึ่งเป็นโปรแกรมที่ออกแบบมาเพื่อช่วยในการเขียนโปรแกรมได้ง่ายขึ้น โดยไม่ต้องแยกใช้ editor เขียนโปรแกรม แล้วเรียกใช้

compiler ทำการคอมไพล์โปรแกรมอีก เช่น edit plus , visual studio , Bloodshed Dev-C++ และ code::blocks ในการเรียนการสอนรายวิชาการเขียนโปรแกรมคอมพิวเตอร์ ได้เลือกใช้ IDE ของ Microsoft visual studio ซึ่งเป็นซอฟต์แวร์ที่ download จากเว็บไซต์ https://visualstudio.microsoft.com/ ชื่อโปรแกรม Visual Studio Community 2022 ชื่อไฟล์ VisualStudioSetup.exe

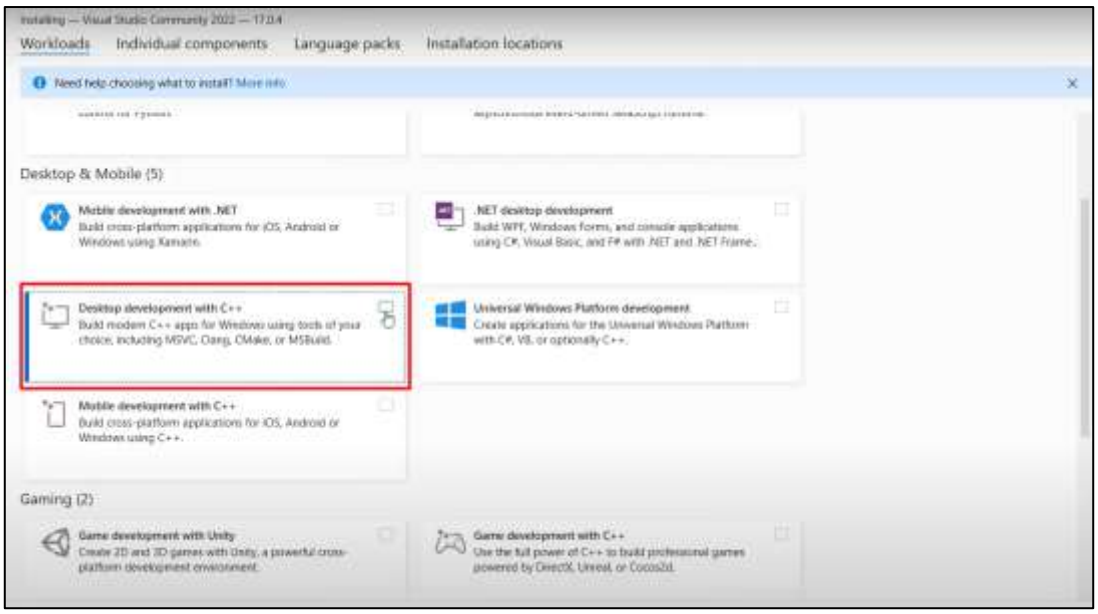
1.1.3.1 การดาวน์โหลดและติดตั้งโปรแกรม Visual Studio 2022

1) ไปที่เว็บไซต์ https://visualstudio.microsoft.com/ เลือกดาวน์โหลดโปรแกรม Visual Studio 2022 เวอร์ชัน Community ดังรูป 1-2



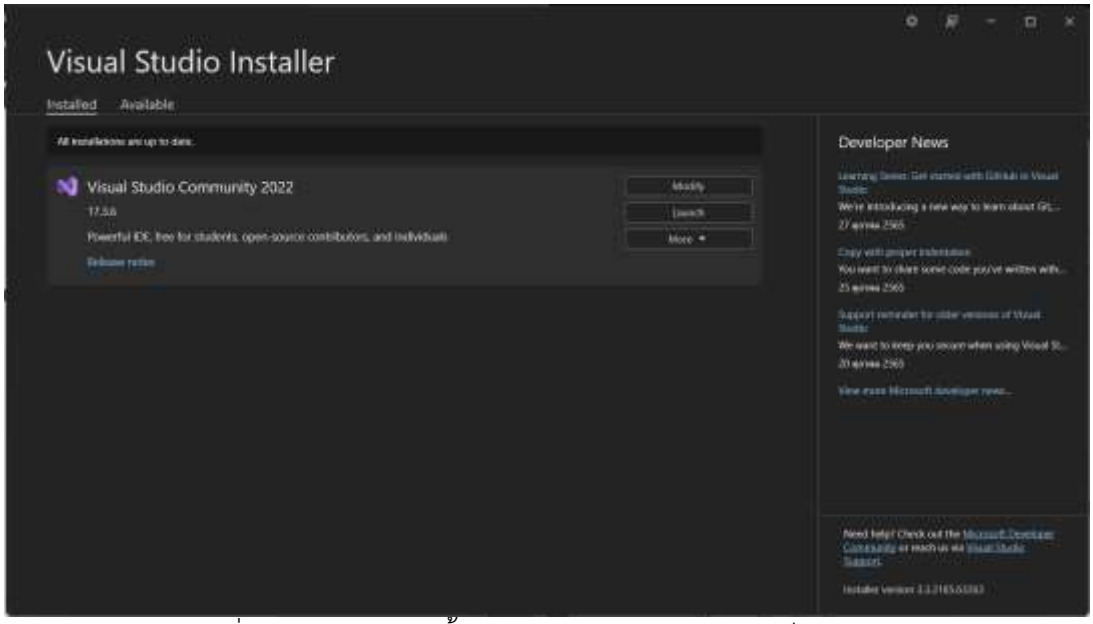
รูปที่ 1-2 แสดงการดาวน์โหลดตัวติดตั้งโปรแกรม visual studio 2022

2) ดับเบิลคลิกไฟล์ที่ดาวน์โหลดมา เพื่อเข้าสู่ขั้นตอนการติดตั้งโปรแกรม visual studio 2022 โดยเลือก Desktop development with C++ และกด install ดังรูปที่ 1-3



รูปที่ 1-3 แสดงหน้าจอเลือกติดตั้งประเภทโปรเจกต์

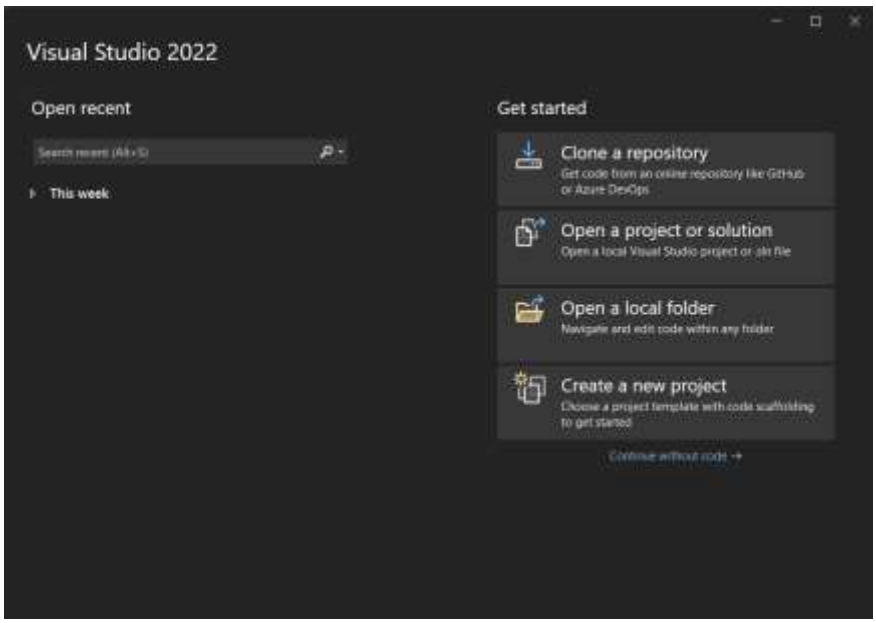
3) รอการดาวน์โหลดและติดตั้งรายการต่าง ๆ ตามที่เลือกจนเสร็จสมบูรณ์ถือว่าโปรแกรม visual studio 2022 พร้อมใช้งานแล้ว ดังรูปที่ 1-4



รูปที่ 1-4 แสดงการติดตั้งโปรแกรม Visual Studio เสร็จสมบูรณ์

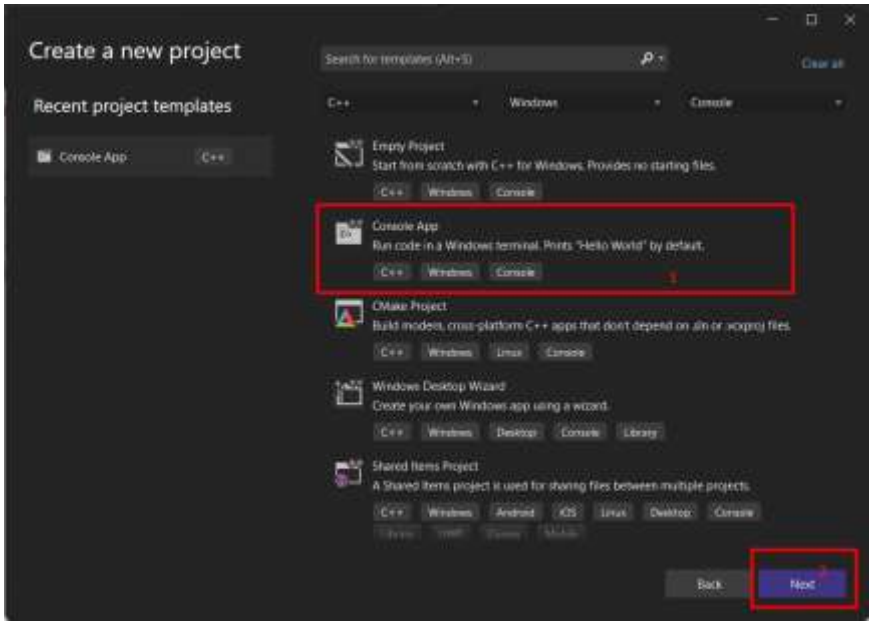
1.1.3.2 การสร้างโปรเจกต์ใน Visual Studio 2022

1) การเรียกใช้ Visual Studio 2022 ทำได้ทำนองเดียวกับการเรียกใช้โปรแกรมอื่น ๆ เช่น เรียกที่ Start->All Programs -> visual studio 2022 จะได้หน้าต่างโปรแกรม visual studio 2022 ดังรูปที่ 1-5



รูปที่ 1-5 หน้าต่างโปรแกรม visual studio 2022

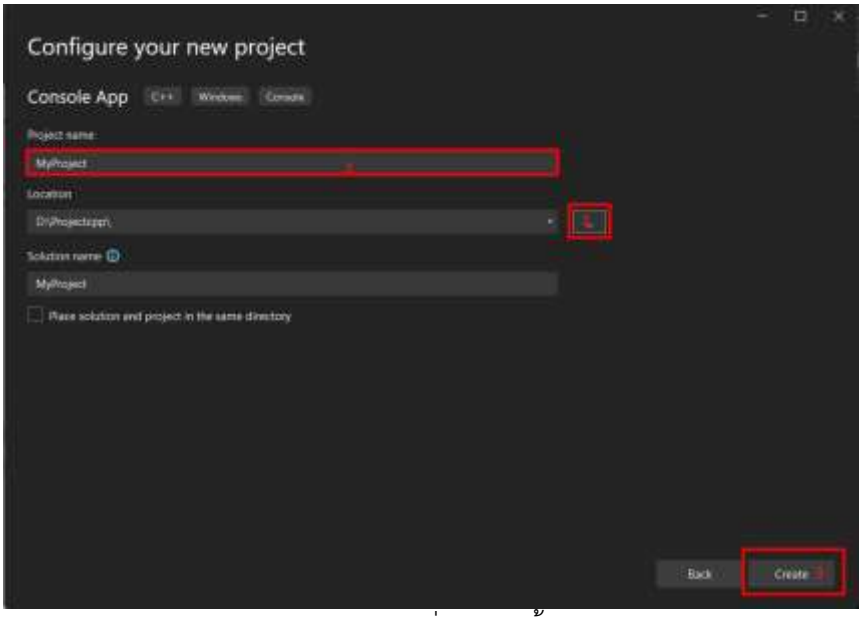
2) ที่หน้าจอเริ่มต้น ให้เลือก Create a new Project จากนั้นเลือกสร้างโปรเจกต์ประเภท Console App ดังรูปที่ 1-6



รูปที่ 1-6 แสดงการสร้างโปรเจกต์ Console App

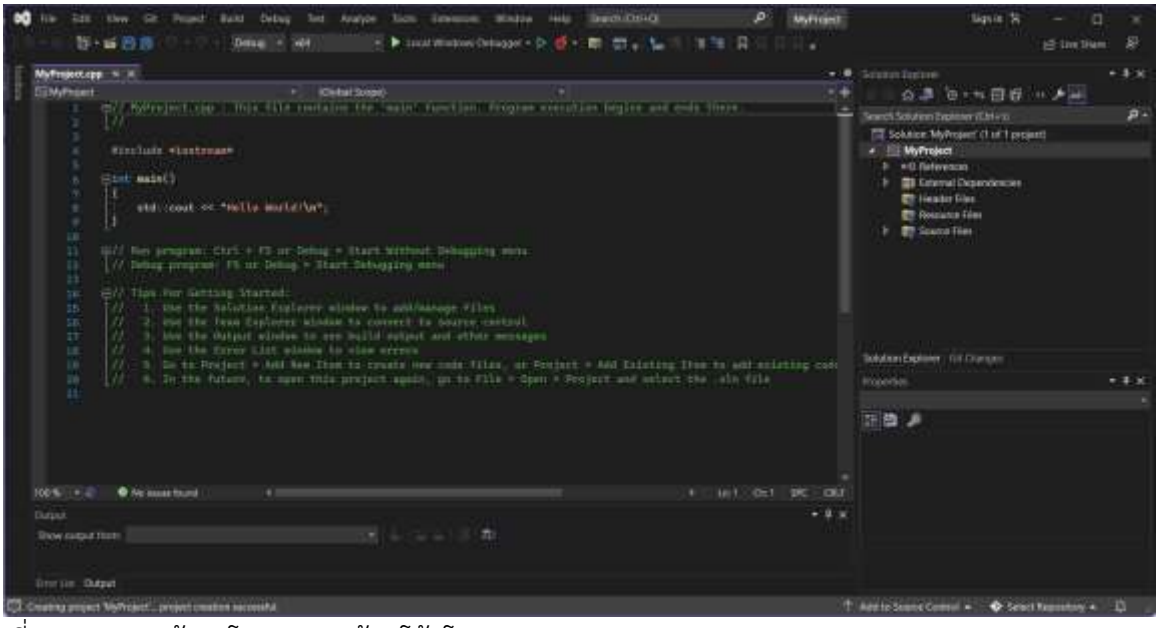
3) การกำหนดรายละเอียดให้กับโปรเจกต์ที่จะสร้างขึ้นมา ดังนี้

- ช่อง Project name หมายถึง ชื่อโปรเจกต์ ในกรณีนี้ตั้งชื่อว่า MyProject
- ช่อง Location หมายถึง พาธที่ใช้จัดเก็บโปรเจกต์ปัจจุบันโดยการคลิกที่ปุ่ม ... เลือกโฟลเดอร์ตามที่คุณต้องการ ในกรณีนี้เก็บโปรเจกต์ไว้ที่ D:\Projectcpp ดังรูปที่ 1-7



รูปที่ 1-7 แสดงการกำหนดรายละเอียดให้กับโปรเจกต์ที่จะสร้างขึ้นมา

4) เมื่อกำหนดรายละเอียดให้กับโปรเจกต์เสร็จแล้วจะปรากฏหน้าจอโปรแกรมพร้อมโค้ดโปรแกรมขึ้นมาดังรูปที่ 1-8



รูปที่ 1-8 แสดงหน้าจอโปรแกรมพร้อมโค้ดโปรแกรม

1.2 โครงสร้างของโปรแกรมที่เขียนด้วยภาษาซีพลัสพลัส

โครงสร้างของโปรแกรมที่เขียนด้วยภาษาซีพลัสพลัสแบ่งย่อยได้เป็น 3 ส่วนดังนี้

ส่วนเรียกใช้ไฟล์อื่นๆ หรือส่วนหัวของฟังก์ชัน เป็นส่วนที่บอกให้คอมไพล์เลอร์ไปดึงไฟล์อื่นที่กำหนดมาแปลร่วมด้วย ไฟล์เหล่านี้อาจจะเป็นไฟล์มาตราฐานที่มีให้แล้วในภาษาซีพลัสพลัส หรือเป็นไฟล์ที่เขียนขึ้นมาใหม่ก็ได้

ส่วนกำหนดชื่อในโปรแกรม หรือ ส่วนประกาศตัวแปร เป็นส่วนที่ใช้กำหนดค่าคงที่ ตัวแปร และค่าอื่น ๆ ที่ต้องการ

ส่วนคำสั่ง จะประกอบด้วยคำสั่งต่างๆ หรือฟังก์ชันอื่นๆ ที่ใช้ในการทำงานของโปรแกรม

<u>ตัวอย่าง</u> โครงสร้างโปรแกรมที่เขียนด้วยภาษาซีพลัสพลัส ดังรูปที่ 1.3

```
#include <iostream>
using namespace std;

char    ch;


int  main(void)
{
        ch = 'A';
        cout<<"Hello world";
        return 0;
}
```

ส่วนเรียกใช้ไฟล์อื่น

ส่วนกำหนดชื่อ

ส่วนคำสั่ง

รูปที่ 1.3 แสดงส่วนประกอบของโปรแกรมที่เขียนด้วยภาษาซีพลัสพลัสอย่างง่าย

ผลที่จากการรันโปรแกรม จะได้ข้อความบนหน้าจอว่า

Hello World

คำอธิบาย

- #include <iostream> คือบอกคอมไพล์เลอร์ให้นำไฟล์ iostream.h มารวมด้วย
- main คือชื่อของฟังก์ชัน โปรแกรมจะเริ่มทำงานที่นี่ และเมื่อจบฟังก์ชัน main หมายถึงจบโปรแกรมด้วย
- char ch คือการประกาศตัวแปร ch เป็นตัวแปรที่มีชนิดเป็นตัวอักษร A
- ch = 'A' คือการให้ค่าตัวแปร ch มีค่าเป็นตัวอักษร A
- cout<<"Hello World" คือการใช้ฟังก์ชัน printf พิมพ์ข้อความที่อยู่ในเครื่องหมาย " " ออกทางอุปกรณ์เอาต์พุตมาตรฐาน

**<u>ข้อสังเกต</u> ทุกประโยคภาษาซีพลัสพลัส (C++ Statement) จะต้องมีเครื่องหมาย ; ปิดท้าย

1.2.1 รูปแบบคำสั่งในภาษาซีพลัสพลัส

รูปแบบคำสั่งในภาษาซีพลัสพลัส มีกฎเกณฑ์ในการเขียนคำสั่ง ดังนี้

1.2.1.1 คำสั่งทุกคำสั่งต้องเขียนด้วยอักษรตัวเล็กเสมอ เช่น คำสั่ง cout, cin , for

1.2.1.2 ทุกคำสั่งจะใช้เครื่องหมาย ; (semi-colon) แสดงการจบของคำสั่ง เช่น cout<<"Hello World";

1.2.1.3 เครื่องหมาย { } (bracket) เครื่องหมายนี้ทำหน้าที่กำหนดขอบเขตของกลุ่มคำสั่งในฟังก์ชัน

1.2.2 การเขียนคำอธิบาย (Comment) ในภาษาซีพลัสพลัส

ภาษาซีพลัสพลัสนิยมการเขียนข้อความอธิบายการทำงานในส่วนต่างๆของโปรแกรมเพื่อให้เข้าใจและอ่านโปรแกรมง่ายขึ้น การเขียนอธิบายจะสามารถเขียนได้ 2 แบบ คือ

ในกรณีที่ข้อความอธิบายเป็นข้อความสั้นๆ ผู้เขียนโปรแกรม สามารถใช้เครื่องหมาย // ก่อนหน้าข้อ-ความ ดังนี้

// ข้อความที่ต้องการอธิบาย

ในกรณีที่ข้อความอธิบายเป็นข้อความยาวๆ สามารถใช้เครื่องหมาย /* และ */ คร่อมข้อความที่ต้องการอธิบาย ดังนี้

```
/* ...................................................
..... ข้อความที่ต้องการอธิบาย.....
.............................................................*/
```

การเขียนคำอธิบายจะเขียนไว้ที่ไหนในโปรแกรมก็ได้ โดยมากจะนิยมเขียนอธิบายขั้นตอนการทำงานก่อนเขียนคำสั่ง หรือ อธิบายความหมายของคำสั่งไว้ข้างคำสั่งนั้นก็ได้

1.3 ตัวแปรและชนิดข้อมูล (Variables and Data types)

ตัวแปร (Variable) หมายถึง ชื่อเรียกแทนพื้นที่เก็บข้อมูลในหน่วยความจำ มีชนิดของข้อมูลหรือชนิดของตัวแปรคือชนิดข้อมูลตัวอักษร (char) ชนิดข้อมูลเลขจำนวนเต็ม (int, long, unsigned,int, unsigned long int) ชนิดข้อมูลเลขทศนิยม (float, double) ชนิดข้อมูลบูลลีน (bool)

คำเฉพาะ (Identifiers) คือ ชื่อของหน่วยข้อมูลในโปรแกรม คำเฉพาะอาจเป็นคำที่คอมไพล์เลอร์สร้างให้ใช้อยู่ในไลบรารี่หรือเป็นคำที่สร้างขึ้นเพื่อใช้งาน โดยคำที่สร้างขึ้นมาเรียกว่าตัวแปร

1.3.1 กฎการตั้งชื่อคำเฉพาะหรือชื่อตัวแปร

1.3.1.1 ตัวอักษรตัวแรกต้องเป็นตัวอักษรหรือ _ (underline character) จะเป็นตัวเลขไม่ได้

1.3.1.2 ตัวถัดไปจะเป็นตัวอักษรหรือ _ หรือตัวเลขก็ได้

1.3.1.3 ห้ามใช้คำสงวน

1.3.1.4 ห้ามมีเครื่องหมายทางคณิตศาสตร์หรือเครื่องหมายวรรคตอนใด ๆ รวมทั้งช่องว่างระหว่างตัวอักษร

การกำหนดชื่อตัวแปร ตัวอักษรภาษาอังกฤษตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ภาษาซีพลัสพลัส ถือว่าเป็นคนละตัวกัน เช่น test และ Test เป็นตัวแปรคนละตัวกัน

ห้ามตั้งชื่อซ้ำกับคำสงวน ซึ่งได้แก่

asm, auto, bool, break, case, catch, char, class, const, const_cast, continue, default, delete, do, double, dynamic_cast, else, enum, explicit, extern, false, float, for, friend, goto, if, inline, int, long, mutable, namespace, new, operator, private, protected, public, register, reinterpret_cast, return, short, signed, sizeof, static, static_cast, struct, switch, template, this, throw, true, try, typedef, typeid, typename, union, unsigned, using, virtual, void, volatile, wchar_t

1.3.2 ชนิดของข้อมูลในภาษาซีพลัสพลัส

1.3.2.1 ตัวแปรแบบ char เป็นตัวแปรที่ใช้สำหรับเก็บข้อมูลที่เป็นตัวอักษรขนาด 1 ตัว โดยใช้เนื้อที่ในการเก็บ 1 ไบต์ ตัวอย่างตัวแปรชนิดนี้ เช่น ‘A’ , ‘b’ , ‘1’ , ‘?’

1.3.2.2 ตัวแปรแบบ int เป็นตัวแปรที่ใช้สำหรับการเก็บค่าตัวเลขที่เป็นจำนวนเต็มที่มีค่าระหว่าง -2147483648 ถึง 2147483649 ใช้เนื้อที่ในการเก็บ 4 ไบต์ ตัวอย่างตัวแปรชนิดนี้ เช่น 5 ,-10, 2000

1.3.2.3 ตัวแปรแบบ long เป็นตัวแปรที่เก็บค่าเป็นจำนวนเต็มที่มีจำนวนไบต์เป็น 2 เท่าของจำนวนเดิม (มักจะใช้เป็นคำนำหน้าตัวแปร เช่น long int)

1.3.2.4 ตัวแปรแบบ float เป็นตัวแปรที่ใช้เก็บข้อมูลที่เป็นเลขทศนิยม โดยจะเก็บอยู่ในรูป a.b x 10e ใช้พื้นที่ในการเก็บ 4 ไบต์ มีค่าระหว่าง 3.4E-38 ถึง 3.4E+38 หรือ แสดงเป็น เลขทศนิยมได้ไม่เกิน 6 ตำแหน่ง ตัวอย่างตัวแปรชนิดนี้ เช่น 10.625, -6.67

1.3.2.5 ตัวแปรแบบ double เป็นตัวแปรที่เก็บข้อมูลที่เป็นเลขทศนิยมเหมือนกับ float แต่จะใช้พื้นที่ในการเก็บมากกว่าเดิม 2 เท่า คือมีขนาด 8 ไบต์ มีค่าระหว่าง 1.7E-308 ถึง 1.7E+308

1.3.2.6 ตัวแปรแบบ unsigned แสดงว่าเป็นตัวแปรที่เก็บค่าเป็นจำนวนเต็มแบบไม่คิดเครื่องหมาย (เป็นบวกเท่านั้น) มักจะใช้เป็นคำนำหน้าตัวแปร ตัวอย่างการใช้งาน เช่น unsigned int

1.3.2.7 ตัวแปรแบบ bool เป็นตัวแปรเก็บข้อมูลเฉพาะจริง (true) หรือเท็จ (false) เท่านั้น

ชนิด	ขนาดความกว้าง	ช่วงของค่า
char	1 ไบต์	character (-128 ถึง 127)
int	4 ไบต์	-2147483648 ถึง 2147483647
short int	2 ไบต์	-32768 ถึง 32767
long int	4 ไบต์	-2147483648 ถึง 2147483647
float	4 ไบต์	3.4E-37 ถึง 3.4E+38 หรือ ทศนิยม 6 ตำแหน่ง
double	8 ไบต์	1.7E-307 ถึง 1.7E+308 หรือ ทศนิยม 12 ตำแหน่ง
unsigned int	4 ไบต์	4 ไบต์ 0 ถึง 4294967295
bool	1 ไบต์	true or false หรือ 1 กับ 0

ตารางที่ 1-2 แสดงชนิดและขนาดพื้นที่เก็บข้อมูล

1.3.3 การประกาศและการกำหนดค่าเริ่มต้นให้กับตัวแปร

การประกาศตัวแปร มีลักษณะดังนี้

ชนิดตัวแปร ชื่อตัวแปร , ชื่อตัวแปร , ชื่อตัวแปร,.....;

การกำหนดค่าเริ่มต้นให้กับตัวแปรมีลักษณะดังนี้

ชนิดตัวแปร ชื่อตัวแปร =ค่าของข้อมูล ;

ตัวอย่างของการประกาศตัวแปรได้แก่

ตัวอย่างที่ 1 int a ;

หมายความว่า ประกาศตัวแปร a เป็นตัวแปรที่ใช้สำหรับเก็บค่าที่เป็นเลขจำนวนเต็มที่มีค่าอยู่ระหว่าง -2147483648 ถึง 2147483649

ตัวอย่างที่ 2 int num1=8;

หมายความว่า ประกาศตัวแปร num1 เป็นตัวแปรที่เก็บค่าตัวเลขจำนวนเต็ม โดยให้ค่าเริ่มต้นเท่ากับ 8

ตัวอย่างที่ 3 float money,price ;

หมายความว่า money และ price เป็นตัวแปรที่ใช้สำหรับเก็บค่าที่เป็นเลขทศนิยม โดยจะให้ตำแหน่งทศนิยมได้ไม่เกิน 6 หลัก

ตัวอย่างที่ 4 char ch='A';

หมายความว่า ประกาศตัวแปร ch เป็นตัวแปรที่เก็บค่าตัวอักษรเพียง 1 ตัว คือ ตัวอักษร 'A'

ตัวอย่างที 5 unsigned long int test;

หมายความว่า ประกาศตัวแปร test เป็นตัวแปรที่ใช้สำหรับเก็บค่าที่เป็นเลขจำนวนเต็ม แบบยาวที่ไม่คิดเครื่องหมาย

ตัวอย่างเพิ่มเติม

```
char  a,b,c,d;          /* ตัวแปร a,b,c,d เป็นตัวแปรชนิด character */
unsigned  e;                /* ตัวแปร e เป็นตัวแปรชนิด unsigned int */
char  key = 'A';        /* ตัวแปร key เป็นตัวแปรชนิด character มีค่า 'A' */
```

นอกจากนี้ การกำหนดตัวแปรใน สามารถทำได้ 2 แบบ คือ

กำหนดไว้นอกกลุ่มคำสั่ง หรือฟังก์ชัน เรียกตัวแปรนี้ว่า Global Variable กำหนดไว้นอกฟังก์ชัน ใช้งานได้ทั้งโปรแกรม มีค่าเริ่มต้นเป็น 0 (กรณีไม่ได้กำหนดค่าเริ่มต้น)

กำหนดไว้ในกลุ่มคำสั่ง หรือฟังก์ชัน เรียกตัวแปรนี้ว่า Local Variable กำหนดไว้ภายในฟังก์ชัน ใช้งานได้ภายในฟังก์ชันนั้น และไม่ถูกกำหนดค่าเริ่มต้นโดยอัตโนมัติ

ตัวอย่างของกำหนดตัวแปรแบบ Global และ Local

```
บรรทัดที่ 1.   #include<iostream.h>
บรรทัดที่ 2.   #define PI 3.14159
บรรทัดที่ 3.   int  area;          /* global variable */
บรรทัดที่ 4.   int main()
บรรทัดที่ 5.   { float radius;     /* local variable */
บรรทัดที่ 6.
บรรทัดที่ 7.      .........
```

บรรทัดที่ 8. .....

บรรทัดที่ 9.

บรรทัดที่ 10. }

1.3.4 ค่าคงที่ (constants)

คือ ตัวแปรที่กำหนดค่าโดยไม่สามารถเปลี่ยนแปลงค่าของมันได้อีก การประกาศค่าคงที่สามารถทำได้โดย

การใช้คำสงวน const นำหน้าประเภท ดังนี้

const ชนิดตัวแปร ชื่อตัวแปร = ค่าที่ต้องการกำหนด;

เช่น const int width = 100;

const int zip = 92000;

กำหนดในส่วนของ Preprocessor directive โดยใช้ #define ทำหน้าที่ใช้กำหนดค่าคงที่ที่เป็นชื่อแทน คำ นิพจน์ คำสั่ง หรือคำสั่งหลายคำสั่ง มีรูปแบบการใช้งานดังนี้

#define ชื่อตัวแปร (ชื่อที่ใช้แทน) ค่าที่ต้องการกำหนด

เช่น #define TEN 10 กำหนด TEN แทนค่า 10

#define PI 3.141592654 กำหนด PI แทนค่า 3.141592654

1.4 เครื่องหมายดำเนินการ (Operators)

เครื่องหมายดำเนินการ (Operators) หรือ ตัวดำเนินการ นั้นกำหนดการการกระทำที่เกิดขึ้นกับตัวแปรและค่าคงที่ โดยที่นิพจน์ประกอบด้วยตัวแปร และค่าคงที่ และใช้ตัวดำเนินการคำนวณเพื่อให้ได้ค่า ตัวดำเนินการในภาษาซีพลัสพลัส มีจำนวนมากเมื่อเปรียบเทียบกับภาษาคอมพิวเตอร์อื่น ๆ ภาษาซีพลัสพลัสได้แบ่งตัวดำเนินการออกเป็น 3 ประเภท ได้แก่ ตัวดำเนินการทางคณิตศาสตร์ ตัวดำเนินการสัมพันธ์และตัวดำเนินการตรรกะ และ ตัวดำเนินการประกอบ

1.4.1 ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)

ตัวดำเนินการทางคณิตศาสตร์ เป็นตัวดำเนินการที่คำนวณทางด้านคณิตศาสตร์ให้มีค่าออกมามีชนิดสอดคล้องกับชนิดของตัวถูกกระทำ ประกอบด้วยตัวดำเนินการต่างๆ ดังนี้

เครื่องหมาย	ความหมาย	ตัวอย่าง
+	การบวก	A+B
-	การลบ	A-B
*	การคูณ	A*B
/	การหาร	A/B

เครื่องหมาย	ความหมาย	ตัวอย่าง
%	การหารเอาแต่เศษไว้ (Modulo)	5%3=1 เศษ 2 จะเก็บแต่เศษ 2 เอาไว้
--	การลดค่าลงครั้งละ 1	A -- จะเหมือนกับ A=A-1
++	การเพิ่มค่าขึ้นครั้งละ 1	A++ จะเหมือนกับ A=A+1

ตารางที่ 3.1 แสดงประเภทของตัวดำเนินการทางคณิตศาสตร์

1.4.2 ตัวดำเนินการเพิ่ม และลดค่า (Increment & Decrement)

ตัวดำเนินการเพิ่ม และลดค่า เป็นตัวดำเนินการในการเพิ่มหรือลดค่าของตัวแปร โดย

ตัวดำเนินการเพิ่มค่า ++ จะบวกหนึ่งเข้ากับตัวถูกดำเนินการ

ตัวดำเนินการลดค่า -- จะลบหนึ่งออกจากตัวถูกดำเนินการ

สาเหตุที่มีตัวดำเนินการนี้ เพราะการดำเนินการเพิ่มหรือลดค่าเกิดขึ้นเสมอ สำหรับตัวดำเนินการ ตัวดำเนินการเพิ่ม และลดค่ามีวิธีใช้งาน 2 แบบคือ

Prefix คือวางไว้หน้าตัวแปร เช่น ++n

Postfix คือวางไว้หลังตัวแปร เช่น n++

ในทั้งสองกรณี ผลลัพธ์คือการเพิ่มค่า 1 ให้กับ n แต่นิพจน์ ++n จะเพิ่มค่าก่อนที่จะนำค่าไปใช้ ขณะที่ n++ จะเพิ่มค่าหลังจากนำค่าไปใช้ ซึ่งหมายความว่าถ้ามีการนำค่าไปใช้ (ไม่เพียงหวังเฉพาะผลลัพธ์) ++n และ n++ จะแตกต่างกัน ตัวอย่างดังนี้

ตัวอย่าง ถ้า x มีค่า 5 และ y มีค่า 5

n = x++; จะเป็นการกำหนดค่า n ให้เท่ากับ x แล้วเพิ่มค่า 1 ให้กับ x

n = ++y; จะเป็นการเพิ่มค่า 1 ให้กับ y แล้วกำหนดค่า n ให้เท่ากับ y

ตัวดำเนินการนี้จะใช้ได้กับเฉพาะตัวแปรเท่านั้น

ตัวอย่าง 1.4 การใช้ตัวดำเนินการเพิ่มค่า

```
#include <iostream>
using namespace std;
int main()
{
    int x=2;
    int y=2;
    cout<<"x++ = " << x++<<endl ;
    cout<<"x = " << x<<endl ;
    cout<<"++y = " << ++y<<endl ;
```

```
        cout<<"y = " << y<<"\n" ;
        return 0;
    }
```

ผลลัพธ์ที่ได้

```
    x++ = 2
    x=3
    ++y=3
    y=3
```

1.4.3 ตัวดำเนินการสัมพันธ์ (Relational Operator)

ตัวดำเนินการสัมพันธ์ เป็นเครื่องหมายที่ใช้ในการเปรียบเทียบและตัดสินใจ ซึ่งผลของการเปรียบเทียบจะเป็นได้ 2 กรณีเท่านั้นคือ จริงและเท็จ ภาษาซีพลัสพลัสถือว่าค่าทางตรรกจริงและเท็จมีชนิดเป็น int ดังนั้นผลการกระทำทางตรรกจึงมีค่าเป็นจำนวนเต็ม และมีค่าได้เพียงสองค่าคือ 1 หรือตัวเลขใด ๆ แทนค่าความจริงเป็นจริง และ 0 แทนค่าความจริงเป็นเท็จ (0 เป็นเท็จ ส่วนตัวเลขอื่นๆ ทั้งหมดมีค่าเป็นจริง)

เครื่องหมายที่เป็นตัวดำเนินการสัมพันธ์มี 4 ตัวคือ

< (น้อยกว่า)

> (มากกว่า)

<= (น้อยกว่าเท่ากับ) และ

>= (มากกว่าเท่ากับ)

ตัวอย่างการใช้งาน เช่น

```
    a<3         /* ถ้า a มีค่าเป็น 5 ประโยคนี้จะได้ผลลัพธ์เป็น 0 (เท็จ) */
    a>b         /* ถ้า a=10  b=8 ประโยคนี้จะได้ผลลัพธ์เป็น 1 (จริง) */
    -1 >= (2.2*X+3.3)
    a<b<c       /* รูปแบบถูกต้อง แต่ลำดับการวางตัวแปรอาจทำให้สับสนได้ โดยจะทำ
การเปรียบเทียบค่า b และ c ก่อน จึงนำผลลัพธ์มาเปรียบเทียบกับ a */
```

ตัวอย่างการใช้งานผิดแบบ

```
    a =<    b       /* ผิดลำดับ   เครื่องหมาย = มาก่อน < */
    a <  =  b       /* ผิดรูปแบบ ห้ามเว้นวรรคระหว่าง < และ = */
```

1.4.4 ตัวดำเนินการเท่ากับ (Equality Operator)

ตัวดำเนินการเท่ากับ ใช้ในการเปรียบเทียบค่า 2 ค่า ว่ามีค่าเท่า หรือไม่เท่ากัน ผลลัพธ์ที่ได้จะมีเพียง 2 ค่าเช่นเดียวกับผลลัพธ์ของตัวดำเนินการสัมพันธ์คือ จริง และ เท็จ

เครื่องหมายที่เป็นตัวดำเนินการเท่ากับมี 2 ตัวคือ

== (เท่ากับ) != (ไม่เท่ากับ)

ตัวอย่างการใช้งาน เช่น

c == 'A' /* ถ้าตัวแปร c เก็บค่าอักขระ A ผลลัพธ์จะได้ค่าจริง */

k != -2 /* ถ้าตัวแปร k เก็บค่าตัวเลข –2 ผลลัพธ์จะได้ค่าเท็จ */

x+y == 2 * z – 5

ตัวอย่างการใช้งานผิดแบบ

a=b /* การเปรียบเทียบค่าตัวแปร a กับ b ว่ามีค่าเท่ากันหรือไม่กลายเป็นการให้ค่าตัวแปร a ด้วยค่าตัวแปร b */

a= =b -1 /* ใช้งานผิดรูปแบบ โดยมีช่องว่างระหว่างเครื่องหมาย = ทั้ง 2 ตัว */

1.4.5 ตัวดำเนินการตรรกะ (Logical Operator)

ตัวดำเนินการตรรกะ ใช้ในการเปรียบเทียบ และกระทำทางตรรกะกับค่าตัวเลข หรือค่าที่อยุ่ในตัวแปร ผลลัพธ์ที่ได้ จะมีเพียง 2 ค่าเช่นเดียวกับผลลัพธ์ของตัวดำเนินการสัมพันธ์คือ จริง และ เท็จ

เครื่องหมายที่เป็นตัวดำเนินการทางตรรกะ มี 3 ตัวคือ

! (นิเสธ)

&& (และ)

|| (หรือ)

**หมายเหตุ นิเสธ คือการกลับค่าความจริงของค่าตัวเลข หรือตัวแปรที่ตามหลังเครื่องหมาย

ตัวดำเนินการนิเสธ ต้องการตัวถูกกระทำการเพียง 1 ตัวเท่านั้น

ตัวอย่างการใช้งาน

!a /* ถ้าค่าความจริงของตัวแปร a มีค่าจริง ผลลัพธ์จากการนิเสธจะได้ค่าเป็นเท็จ */

!(x+7.7)

!(a<b || c<d) /* a มากระทำกับ b แล้วเอาผลลัพธ์มา || กับ ผลลัพธ์ระหว่างการกระทำระหว่าง c กับ d จากนั้นนำผลลัพธ์ที่ได้จากการ || มา ทำการนิเสธอีก */

ตัวอย่างการใช้งานผิดแบบ

a! /* ผิดลำดับเครื่องหมาย ! มาก่อนตัวแปร */

a! = b /* ผิดความต้องการ โดยความต้องการคือ กลายเป็นการใช้เครื่องหมาย != ไป */

1.4.5.1 ตัวดำเนินการ && และ || ต้องการตัวถูกกระทำ 2 ตัว

ตัวอย่างการใช้งาน

a&&b /* a และ b ถ้าค่าความจริงของ a เป็นจริง และ ค่าความจริง b เป็นจริง ผลลัพธ์ที่ได้จะมีค่าความจริงเป็นจริง */

a || b /* a หรือ b ถ้าค่าความจริงของ a เป็นเท็จ และ ค่าความจริง b เป็นเท็จ ผลลัพธ์ที่ได้จะมีค่าความจริงเป็นเท็จ */

!(a<b) && C /* นำ a มาเปรียบเทียบกับ b แล้วนำผลลัพธ์มานิเสธ จากนั้นมา && กับ C */

3 && (-2 * a + 7) /* หาค่าภายในวงเล็บก่อน แล้วนำค่าผลลัพธ์ที่ได้มา && กับ 3 */

ตัวอย่างการใช้งานผิดแบบ

a && /* ตัวกระทำไม่ครบ */

a| |b /* เว้นช่องว่างระหว่างเครื่องหมาย || */

a & b /* เครื่องหมาย && ไม่ครบ */

&b /* ตำแหน่งในหน่วยความจำของตัวแปร b */

1.4.5.2 การหาค่าของตัวดำเนินการตรรกะ

&& (และ)

นำค่าความจริง 2 ค่ามาเปรียบเทียบกัน ได้ผลของการเปรียบเทียบตามตารางที่ 3.4

P	Q	P && Q
0	0	0
0	1	0
1	0	0
1	1	1

ตารางที่ 3.4 แสดงผลของตัวดำเนินการ && ระหว่างตัวแปร P และ Q

|| (หรือ)

นำค่าความจริง 2 ค่ามาเปรียบเทียบกัน ได้ผลของการเปรียบเทียบตามตารางที่ 3.5

P	Q	P || Q
0	0	0
0	1	1
1	0	1
1	1	1

ตารางที่ 3.5 แสดงผลของตัวดำเนินการ || ระหว่างตัวแปร P และ Q

! (นิเสธ)

นำค่าความจริงมาเปรียบเทียบ ได้ผลของการเปรียบเทียบตามตารางที่ 3.6

P	!P
0	1
1	0

ตารางที่ 3.6 แสดงผลของตัวดำเนินการ ! กับตัวแปร P

1.4.6 ตัวดำเนินการประกอบ (Compound Operator)

ตัวดำเนินการประกอบ คือ ตัวดำเนินการที่เป็นรูปแบบย่อ ของตัวดำเนินการ+ตัวแปรที่ถูกดำเนินการ โดยตัวดำเนินการประกอบในภาษาซี มีทั้งหมด 10 ตัวดังนี้

+= -= /= %=

<<= >>= != ^=

การกำหนดรูปแบบของตัวดำเนินการประกอบให้กับตัวแปรมีรูปแบบดังนี้

Variable Operator = expression;

จะมีความหมายเท่ากับ

Variable = variable Operator expression;

ตัวอย่างเช่น

i = i + 1; ตัวดำเนินการประกอบคือ i += 1;

i = i – a; ตัวดำเนินการประกอบคือ i - = a;

i = i * (a + 1); ตัวดำเนินการประกอบคือ i *=a+1;

i = i / (a-b); ตัวดำเนินการประกอบคือ i /= a-b;

1.4.8 ลำดับการทำงานก่อน-หลังของตัวดำเนินการ

ลำดับการทำงานก่อน-หลังของตัวดำเนินการเป็นไปตามตารางที่ 3.7

ตัวดำเนินการ	ทิศทางการดำเนินการ
(), [],	ซ้ายไปขวา
++, --, +(ค่าบวก), -(ค่าลบ), sizeof	ขวาไปซ้าย
*, /, %	ซ้ายไปขวา
+, - (ตัวกระทำการทางคณิตศาสตร์)	ซ้ายไปขวา
<, <=, >, >=	ซ้ายไปขวา
==, !=	ซ้ายไปขวา
&&	ซ้ายไปขวา
||	ซ้ายไปขวา
=, +=, -=, /=, %=, !=, <<=, >>=	ขวาไปซ้าย

ตารางที่ 3.7 แสดงลำดับความสำคัญของตัวดำเนินการทั้งหมด เรียงลำดับจากมากไปน้อย

1.5 อินพุต และเอาต์พุต (Input and Output)

การสั่งให้เครื่องคอมพิวเตอร์ทำงานใดๆ และรอรับค่าผลลัพธ์จากเครื่อง จะต้องมีการติดต่อกับเครื่อง โดยการส่งผ่านค่าอินพุตไปยังเครื่องคอมพิวเตอร์ และรับค่าเอาต์พุตจากเครื่องคอมพิวเตอร์ดังนี้

อินพุต คือ การรับค่าข้อมูลของผู้ใช้เข้าไปในเครื่องคอมพิวเตอร์ อุปกรณ์อินพุตมาตรฐาน ได้แก่ คีย์บอร์ดหรือแป้นพิมพ์ สแกนเนอร์ หรือ เมาส์

เอาต์พุต คือ การแสดงผลข้อความ ข้อมูล หรือ ค่าตัวแปรใดๆ ออกมาแสดงให้กับผู้ใช้ทางอุปกรณ์แสดงผลเอาต์พุตต่างๆ อุปกรณ์

เอาต์พุตมาตรฐาน ได้แก่ จอภาพ ลำโพง หรือ เครื่องพิมพ์

เครื่องคอมพิวเตอร์ทั่วไป ไม่สามารถจดจำตัวอักษรในลักษณะของรูปร่างตัวอักษรได้ การจดจำรูปร่างตัวอักษรเพื่อนำไปใช้งานนั้น คอมพิวเตอร์จะจดจำในรูปรหัสแทนตัวอักษร เมื่อต้องการแสดงตัวอักษร คอมพิวเตอร์ก็จะนำค่ารหัสเหล่านี้ไปเปิดตารางภาพตัวอักษร แล้วนำภาพตัวอักษรตามรหัสนั้นไปแสดงต่อไป

สตริง (String) คือกลุ่มข้อมูลที่ประกอบไปด้วยตัวอักขระที่เขียนเรียงกันไป ใช้ตัวแปรชนิด อักขระในการเก็บภาพข้อมูลต่อหนึ่งหน่วยตัวอักษร โดยจะมีเครื่องหมาย “ ” อัญประกาศ (Double quote) ล้อมรอบอยู่เสมอ เช่น Hello World! ในภาษาซีพลัสพลัสต้องเขียนเป็น “Hello World!”

1.5.1 การแสดงผลข้อมูล

ในการแสดงผลข้อมูลในภาษาซีพลัสพลัสจะมีคำสั่งมาตรฐานในการแสดงผลข้อมูล หรือค่าตัวแปรออกมาทางจอภาพ คำสั่งนั้นคือ cout (อ่านว่า ซี-เอาต์)

ตัวอย่างเช่น เมื่อผู้เขียนโปรแกรมต้องการจะแสดงข้อความ “Hello World” บนหน้าจอคอมพิวเตอร์ ผู้เขียนโปรแกรมสามารถใช้คำสั่ง cout ได้ดังนี้

```
#include <iostream>
using namespace std;
int main()
{           cout<<”Hello World!\n”;
            return 0;
}
```

จะได้ผลลัพธ์ ทางหน้าจอคือ

```
Hello World!
```

พร้อมกับการขึ้นบรรทัดใหม่ของเคอร์เซอร์ (cursor)

จากตัวอย่างข้างต้น สามารถอธิบายการทำงานในแต่ละบรรทัดของโปรแกรมได้ดังนี้

#include <iostream>

จะเป็นการแจ้ง C++ preprocessor ก่อนการคอมไพล์ ว่าในโปรแกรมนี้จะมีการแสดงข้อมูล (data) บนหน้าจอ หรือ รับข้อมูลจากทางคีบอร์ด ซึ่งในตัวอย่างนี้จะหมายถึงการแสดงข้อมูลทางหน้าจอเพราะคำสั่ง cout

int main()

ฟังก์ชั่นหลักของโปรแกรม โดยจะมีการรีเทิร์น (return) ค่าเป็นแบบจำนวนเต็ม (integer).

cout<<”Hello World!\n”;

ทั้งบรรทัดนี้จะถูกเรียกว่า statement ซึ่งประกอบด้วย

cout เป็นคำสั่งที่ใช้เมื่อต้องการแสดงข้อมูลทางหน้าจอ

<< จะถูกเรียกว่า output operator หรือ insertion operator เมื่อโปรแกรมถูกทำงาน ตัวอักษรหรือค่าที่อยู่ต่อจากเครื่องหมายนี้จะถูกแสดง ซึ่งในที่นี้คือประโยค Hello World! ในการแสดงตัวอักษรหรือประโยคในภาษาซีพลัสพลัสนั้น จะต้องอยู่ภายในเครื่องหมายอัญประกาศ (Double Quote)

; เมื่อจบ statement หนึ่งๆ จะต้องถูกปิดด้วย เครื่องหมาย semi-colon

\n จะเห็นว่า \n จะไม่ถูกแสดงบนหน้าจอ เนื่องจาก \ (backslash) จะเป็นตัวอักษรพิเศษ เมื่อรวมกับตัวอักษรที่ด้านหลัง แล้วจะทำให้มีความหมายต่างๆ ซึ่งในที่นี้ จะหมายความว่า เป็นการสั่งให้เคอร์เซอร์ขึ้นบรรทัดใหม่ เราเรียก รหัสพิเศษนี้ว่า Escape Sequence

นอกจาก \n แล้ว ยังมีรหัสพิเศษอื่นๆอีก ดังนี้คือ

Escape Sequence	ค่า	หน้าที่
\a	0x07	เสียงดังออกลำโพงหนึ่งครั้ง
\b	0x08	เลื่อน cursor ไปลบตัวอักษรทางซ้ายมือหนึ่งตัวอักษร
\f	0x0c	ขึ้นหน้าใหม่
\n	0x0a	ขึ้นบรรทัดใหม่
\r	0x0d	เลื่อน cursor ไปทางซ้ายมือสุดของบรรทัด
\t	0x09	เลื่อนเคอร์เซอร์ ไป 1 tab ในแนวนอน
\\	0x5c	เครื่องหมาย \
\′	0x2c	เครื่องหมาย ′
\″	0x22	เครื่องหมาย ″
\?	0x3f	เครื่องหมาย ?

ตารางที่ 2.1 แสดงตัว Escape Sequence ต่างๆ ที่มีใช้ในภาษาซีพลัสพลัส

```
return 0
```

เป็นการจบการทำงานของฟังก์ชั่น main ซึ่งเป็นการบอกว่าต้องการจะออกจากฟังก์ชั่น

ตัวอย่างด้านล่างนี้ เป็นตัวอย่างของการใช้งาน cout

ตัวอย่างที่ 1

```
#include <iostream.h>
int main()
{       cout<<"Hello";
      cout<<"World!\n";
        return 0;
}
```

ผลลัพธ์

```
Hello World!
```

ตัวอย่างที่ 2

```
#include <iostream.h>
int main()
{       cout<<"Hello\n";
      cout<<"World!\n";
        return 0;
```

```
}
```

ผลลัพธ์

```
Hello
World!
```

ตัวอย่างที่ 3

```
#include <iostream.h>
int main()
{       cout<<"The volume of the box is "<< 2*3*4;
        return 0;
}
```

```
The volume of the box is 24
```

ผลลัพธ์

ตัวอย่างที่ 4

```
#include <iostream.h>
int main()
{
      cout<<"The volume of the box is "
          << 2*3*4;
        return 0;
}
```

ผลลัพธ์

```
The volume of the box is 24
```

จากตัวอย่างที่ 4 จะเป็นกรณีที่ ในบรรทัดของคำสั่ง cout มีความยาวมาก ทำให้ไม่สามารถเขียนให้จบได้ภายใน 1 บรรทัด ผู้เขียนโปรแกรมสามารถขึ้นบรรทัดใหม่ได้ โดยขึ้นเครื่องหมาย << ในบรรทัดใหม่ แล้วใส่เครื่องหมาย ; ปิดท้ายในบรรทัดสุดท้าย

ตัวอย่างที่ 5

```
#include <iostream.h>
int main()
{       char x ='M';
      char name[10]="Maliwan";
      cout<<"The first character is "<<x<<endl;
      cout<<"My name is "<<name<<"\n";
        return 0;
```

```
}
```

ผลลัพธ์

```
The first character is M
My name is Maliwan
```

จากตัวอย่างที่ 5 จะมีคำสั่งเพิ่มขึ้นคือ `endl` จะมีความหมายกับ `\n` คือให้ขึ้นบรรทัดใหม่

1.5.2 การรับค่าข้อมูล

ในการรับค่าข้อมูลในภาษาซีพลัสพลัส จะมีคำสั่งมาตรฐานในการรับค่าข้อมูล ผ่านทางอุปกรณ์อินพุตมาตรฐาน(ซึ่งในที่นี้คือ แป้นพิมพ์) คำสั่งนั้นคือ cin (อ่านว่า ซี-อิน)

ตัวอย่างเช่น เมื่อผู้เขียนโปรแกรม ต้องการเขียนโปรแกรมให้คำนวณปริมาตรของกล่องสี่เหลี่ยมใดๆ ซึ่งมีสูตรในการคำนวณคือ กว้าง*ยาว*สูง ผู้เขียนโปรแกรมจะต้องรับค่าความกว้าง,ความยาว และความสูงของกล่องมาคำนวณ ดังนี้

```
#include <iostream.h>
int main()
{       int height,width,length;
      cout<<"Please enter height of box\n";
      cin>> height;
      cout<<"Please enter width of box\n";
      cin>> width;
      cout<<"Please enter length of box\n";
      cin>> length;
      cout<<"\n";
      cout<<"The volume of this box is "<<height*width*length<<"\n";
        return 0;
}
```

ผลลัพธ์

```
Please enter height of box
12
Please enter width of box
12
Please enter length of box
10

The volume of this box is 1440
```

จากตัวอย่างดังกล่าว ได้มีการกำหนดตัวแปร height,width,length เป็นแบบจำนวนเต็ม เพื่อรับค่าความสูง, ความกว้าง และความยาว ตามลำดับ มาใช้การคำนวณ นอกจากนี้ผู้เขียนโปรแกรมสามารถรับค่าของความสูง, ความกว้าง และความยาว ภายในครั้งเดียวได้ แทนที่จะรับค่าทีละบรรทัด ดังตัวอย่าง

```
#include <iostream.h>
int main()
{       int height,width,length;
      cout<<"Please enter height, width and length of box\n";
      cin>> height >> width >> length;
      cout<<"\n";
      cout<<"The volume of this box is "<<height*width*length<<"\n";
        return 0;
}
```

ผลลัพธ์

```
Please enter height, width and length of box
12 12 10

The volume of this box is 1440
```

จากตัวอย่างดังกล่าว จะสังเกตเห็นว่า หลังคำสั่ง cin จะตามด้วยเครื่องหมาย >> เสมอ เราเรียกเครื่องหมายนี้ว่า input operator หรือ extraction operator

ตัวอย่างของการใช้ cin

```
#include <iostream.h>
int main()
{       char name[15];
        int age;
      cout<<"Please enter your name\n";
      cin>> name;
      cout<<"Please enter your age\n";
      cin>> age;
      cout<<"Your name is "<<name<< "and your age is "<< age<< "\n";
        return 0;
}
```

ผลลัพธ์

```
Please enter your name
Thana
Please enter your age
20
Your name is Thana and your age is 20
```