

บทที่ 1

เรื่อง การใช้งาน Microsoft Visual Studio 2010 Professional และโครงสร้างภาษา C++

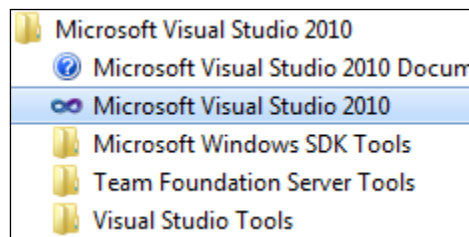
วัตถุประสงค์

- เพื่อให้ นักศึกษา รู้จักและ การใช้งาน Microsoft Visual Studio 2010 Professional
- เพื่อให้ นักศึกษา รู้จักประวัติ และ ขั้นตอนการพัฒนาโปรแกรมของ ภาษา C++
- เพื่อให้ นักศึกษา รู้จัก โครงสร้างโปรแกรม ภาษา C++

การใช้งาน Microsoft Visual Studio 2010 Professional

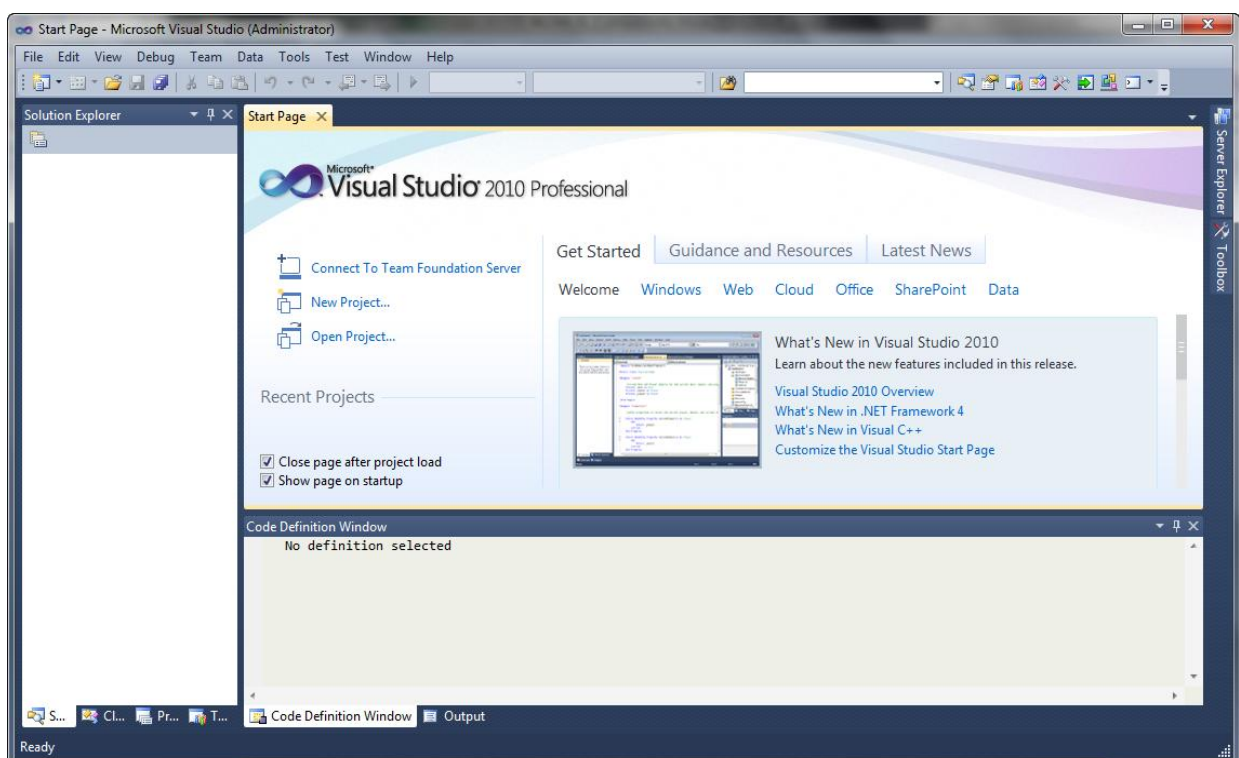
การเรียกใช้โปรแกรม Microsoft Visual Studio 2010 Professional

หลังจากที่ได้ทำการติดตั้งโปรแกรม Microsoft Visual 2010 Professional เป็นที่เรียบร้อยแล้ว การเรียกใช้งานโปรแกรม จะสามารถเรียกใช้ได้ โดยการคลิกปุ่ม Start → All Programs → Microsoft Visual Studio 2010 → Microsoft Visual Studio 2010 ดังรูปที่ 1-1




รูปที่ 1-1 แสดงการเรียกใช้งานโปรแกรม Microsoft Visual Studio 2010

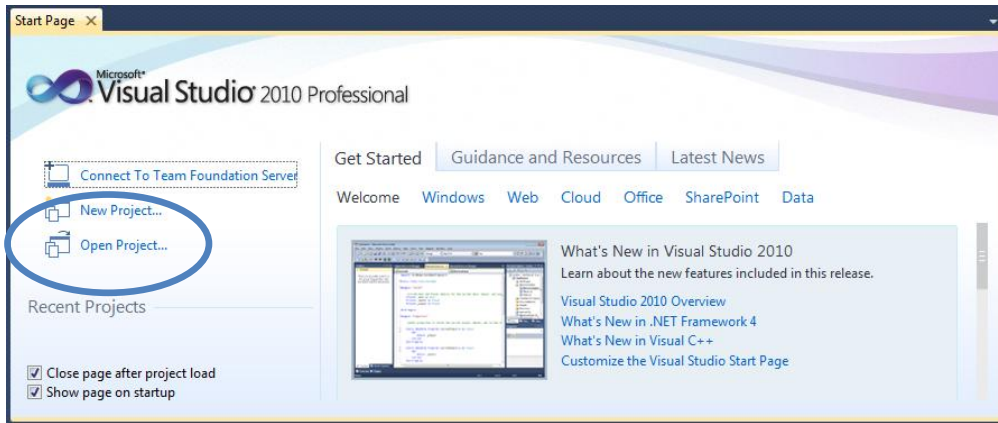
และหลังจากได้เรียกโปรแกรมเสร็จแล้ว ก็จะมีปรากฏหน้าจอโปรแกรมดังรูปที่ 1-2 ซึ่งพร้อมที่จะเริ่มทำงาน



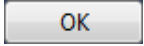
รูปที่ 1-2 แสดงหน้าจอแรกของโปรแกรม Microsoft Visual Studio 2010 Professional

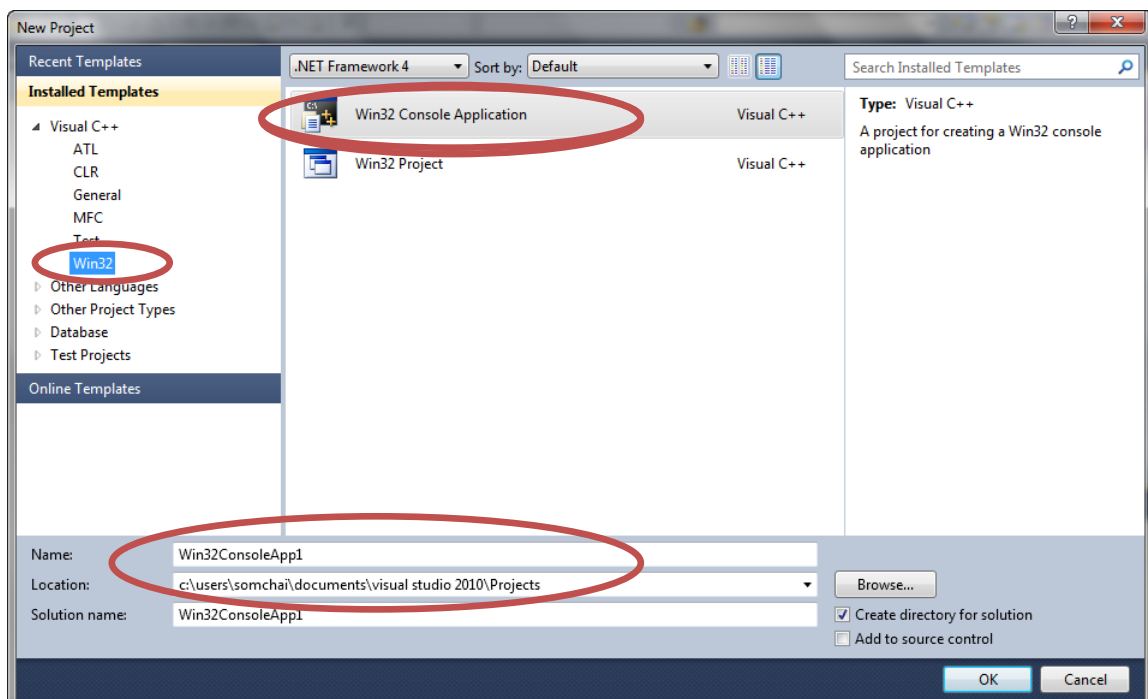
การสร้าง Project File

หลังจากปรากฏหน้าจอโปรแกรม Microsoft Visual Studio 2010 จะมีแถบชื่อ Start Page ซึ่งจะสามารถทำงานได้ 2 อย่างคือ New Project เพื่อสร้าง Project ใหม่ และ Open Project เพื่อเปิด Project ที่ได้สร้างไว้แล้ว แต่ถ้าหากหน้าแถบ Start Page ไม่ปรากฏ ก็สามารถเรียกให้แสดงจากเมนู View → Start Page หรือปุ่ม  บน Tool Bar ก็จะมีปรากฏหน้าจอ Start Page ดังรูปที่ 1-3



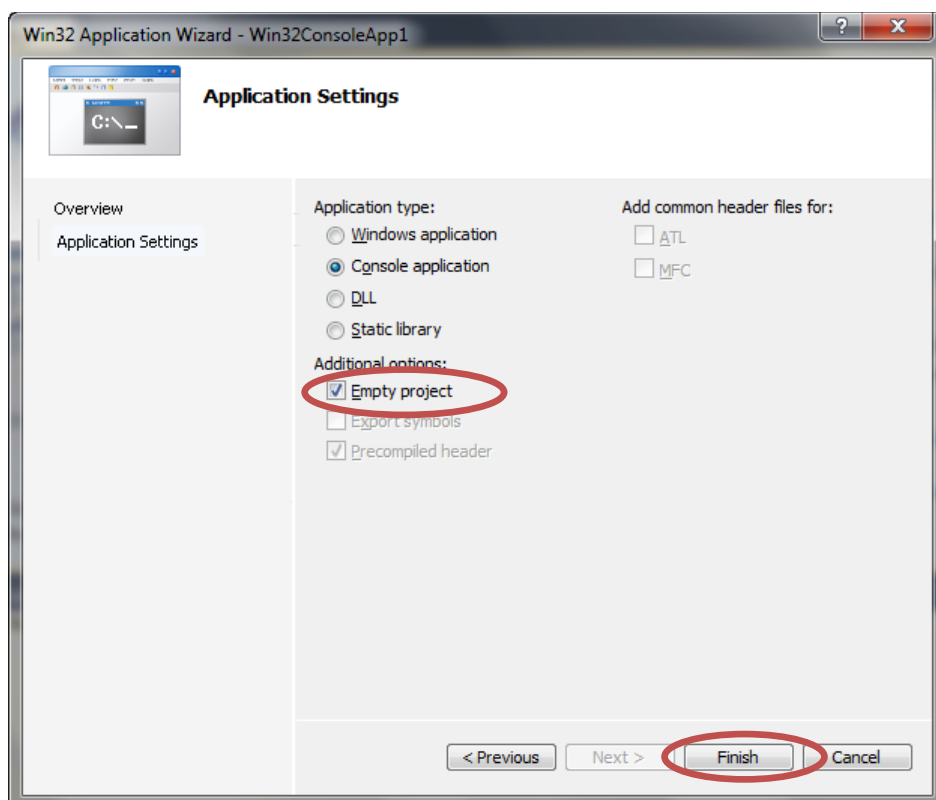
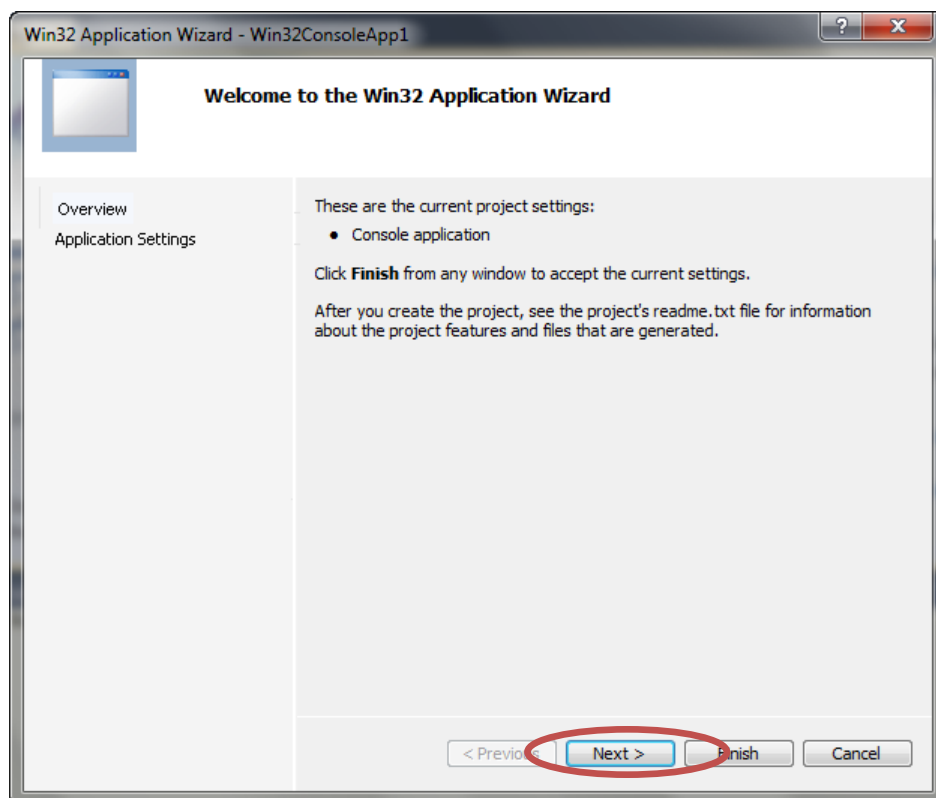
รูปที่ 1-3 แสดงหน้าจอ Start Page

โดยจะเริ่มจากการสร้าง Project ใหม่ ให้คลิกเลือกที่ New Project ตามรูปที่ 1-3 จากนั้นจะปรากฏหน้าต่างของ New Project ตามรูปที่ 1-4 ซึ่งจะแสดงค่า default เป็น Visual C++ และจะแสดงชนิด Project ให้เลือก โดยเลือกชนิดเป็น Win32 แล้วคลิกเลือก Win32 Console Application แล้วพิมพ์ชื่อ Project ให้ใส่ชื่อว่า Win32ConsoleApp1 หรือชื่อตามที่ต้องการ(ไม่ควรตั้งชื่อเป็นภาษาไทย) สำหรับตำแหน่งการจัดเก็บจะอยู่ที่ “C:\Documents and Settings\(\ตามชื่อผู้)\My Documents\Visual Studio 2010\Projects” กรณีเป็น WindowsXP แต่หากเป็น “C:\Users\(\ตามชื่อผู้)\My Documents\Visual Studio 2010\Projects” กรณีเป็น Windows7 จากนั้นให้คลิกปุ่ม  แต่หากไม่ใช่ภาษาที่ต้องการทำงาน ก็ให้ทำการเลือกภาษาก่อน โดยเลือกที่ Other Languages



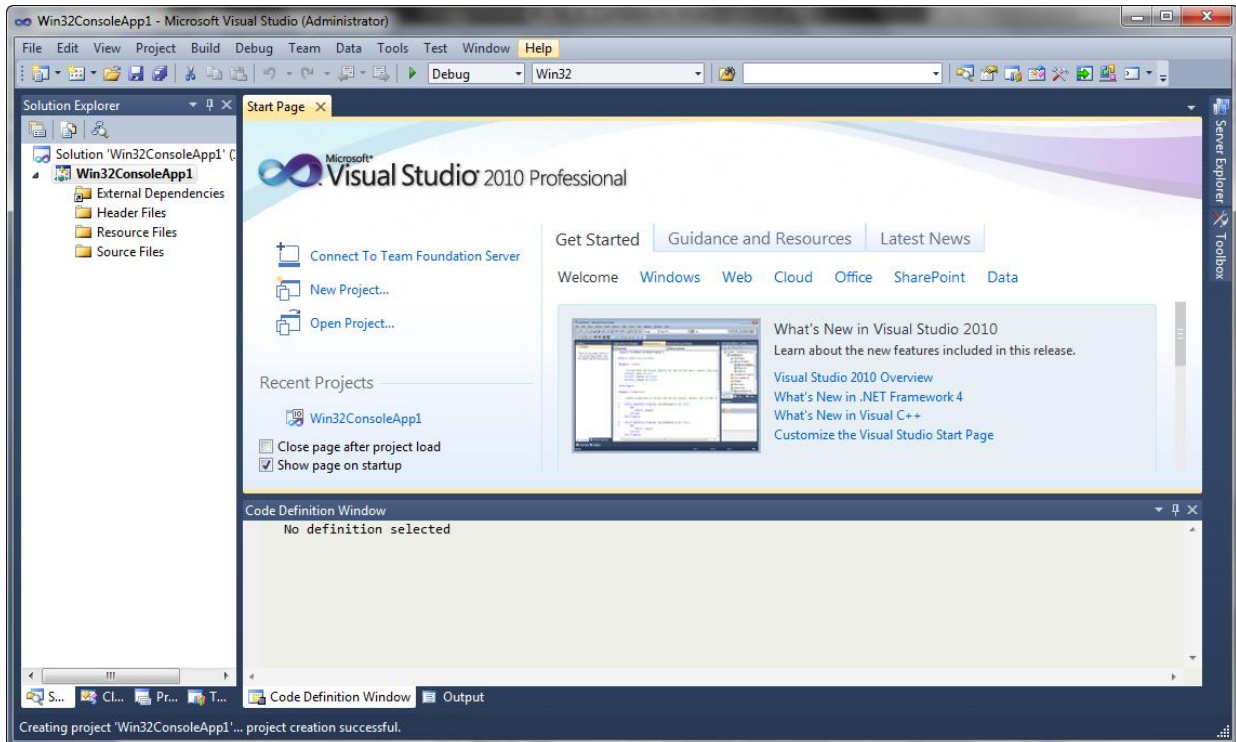
รูปที่ 1-4 แสดงหน้าจอการสร้าง Project ใหม่

เมื่อกดปุ่ม OK จะปรากฏหน้าต่าง Wizard ดังรูปที่ 1-5 จากนั้นให้กดปุ่ม **Next >** ก็จะแสดงหน้าจอรายละเอียดของ Project ให้ทำการคลิกเลือกที่ Application Type เป็นแบบ Console application และคลิกเลือก check box เป็น Empty Project แล้วกดปุ่ม **Finish** ก็จะขึ้นหน้าจอ ดังรูปที่ 1-6 ซึ่งจะมีการสร้าง Folder ให้ตามชื่อ Project ที่พร้อมจะเริ่มเขียนโปรแกรม



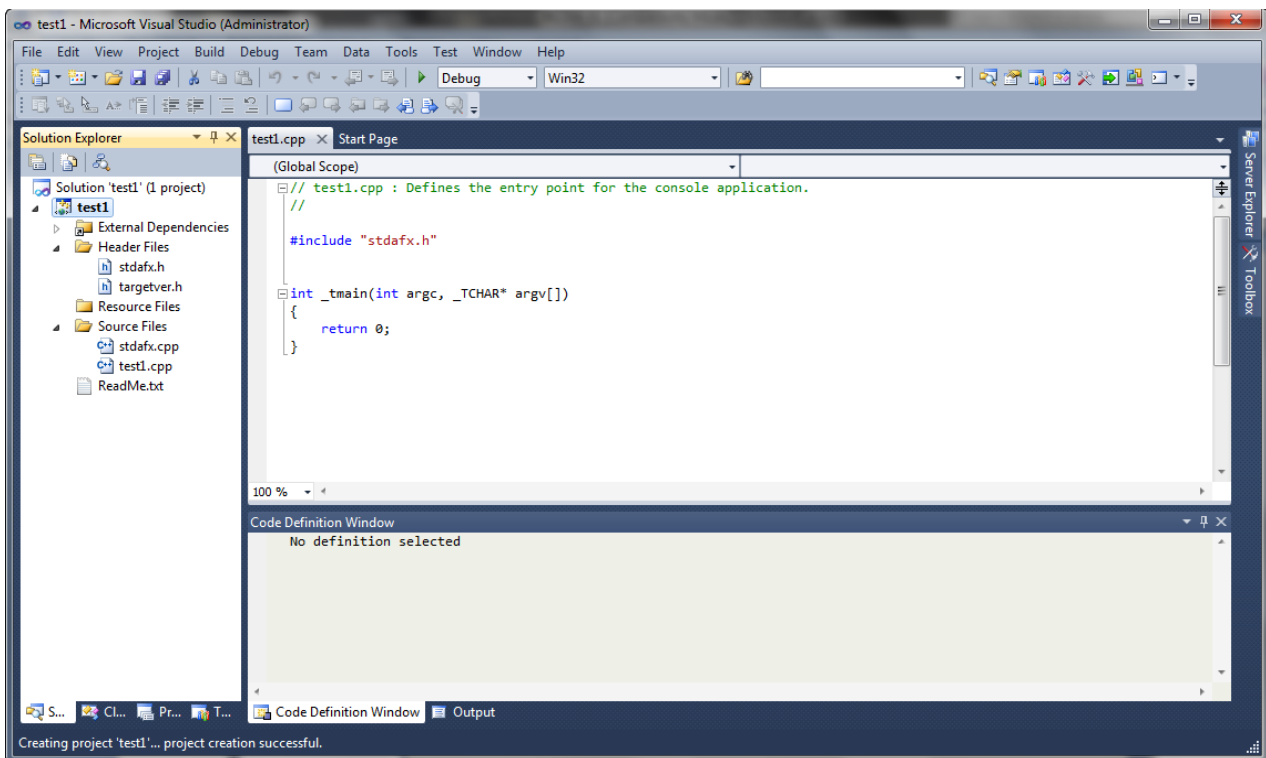
รูปที่ 1-5 แสดงหน้าจอรายละเอียดของ Win32 Application Wizard

หน้าจอ ดังรูปที่ 1-6 เมื่อสร้าง Project เสร็จแล้ว จะปรากฏรายละเอียดของ Solution และ Project อยู่ด้านซ้ายมือในหัวข้อ Solution Explorer ที่พร้อมจะเริ่มเขียนโปรแกรมภาษา C++ ได้

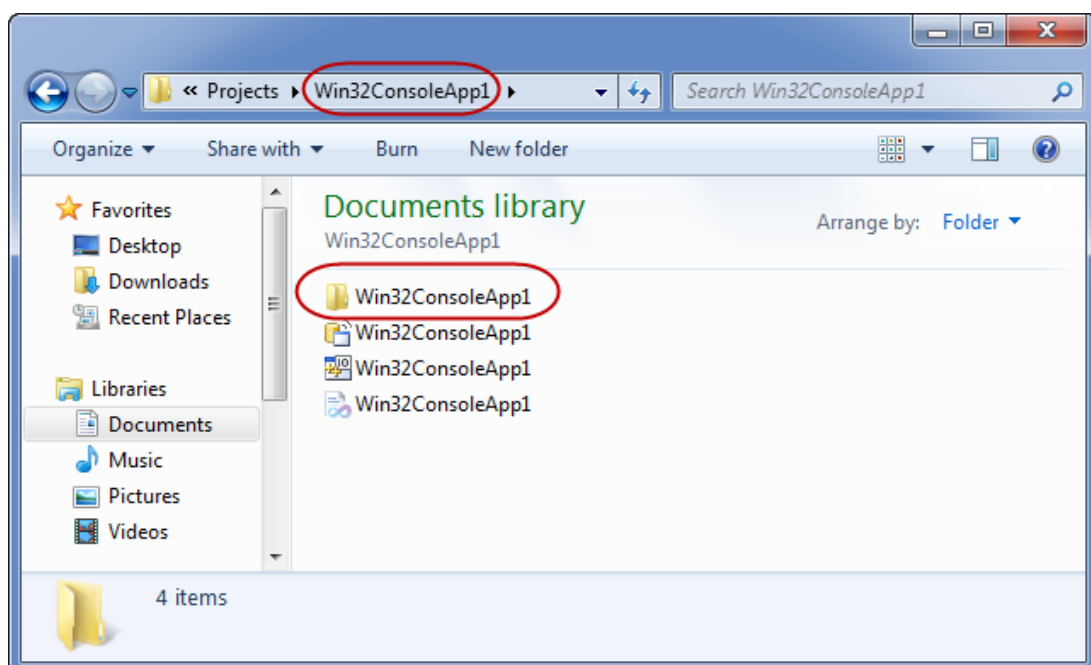


รูปที่ 1-6 แสดงหน้าจอหลังได้สร้าง Project file และเลือก Empty Project

แต่ถ้าหากในรูปที่ 1-5 กดปุ่ม Finish โดยไม่ได้เลือก Empty Project ไว้จะปรากฏหน้าจอเป็นดังรูปที่ 1-7 ซึ่งจะมีการเพิ่มไฟล์ต่างๆ ให้เองตามลักษณะของ .Net ซึ่งไม่ใช่ลักษณะรูปแบบที่จะใช้เรียนในวิชานี้

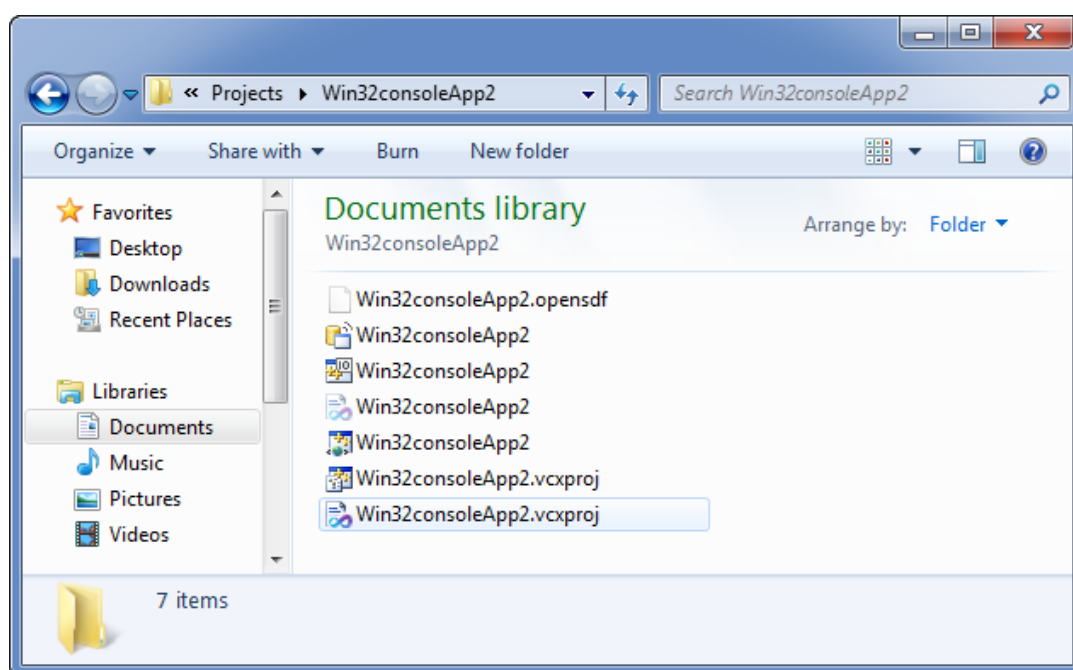


รูปที่ 1-7 แสดงหน้าจอหลังจากได้สร้าง Project ใหม่ และไม่เลือก Empty Project



รูปที่ 1-8 แสดงโครงสร้าง Folder หลังจากได้สร้าง Project ใหม่ และเลือก ☒ Create directory for solution

จากรูปจะเห็นว่าจะมีการสร้าง Folder หลักชื่อ Win32ConsoleApp1 ขึ้นตามชื่อ Project แล้วภายในก็
จะมีการสร้าง Folder ย่อยขึ้นชื่อ Win32ConsoleApp1 ตามชื่อของ Solution ที่กำหนดให้ และภายในจะมีไฟล์
ที่เกี่ยวข้องกับ Project อยู่



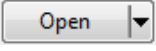
รูปที่ 1-9 แสดงโครงสร้าง Folder หลังจากได้สร้าง Project ใหม่ และไม่เลือก ☐ Create directory for solution

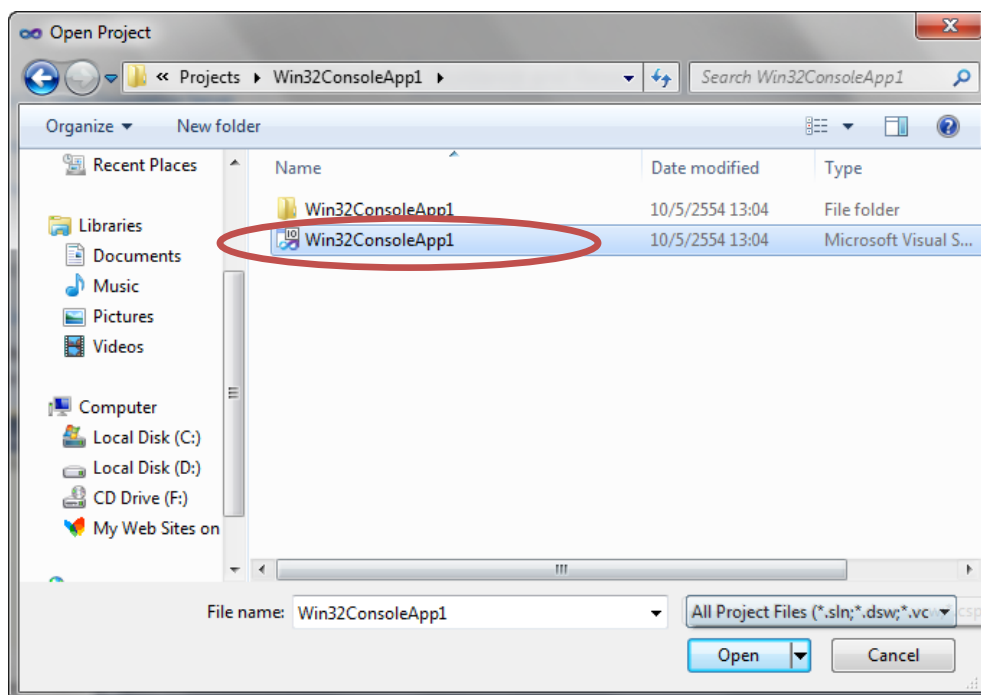
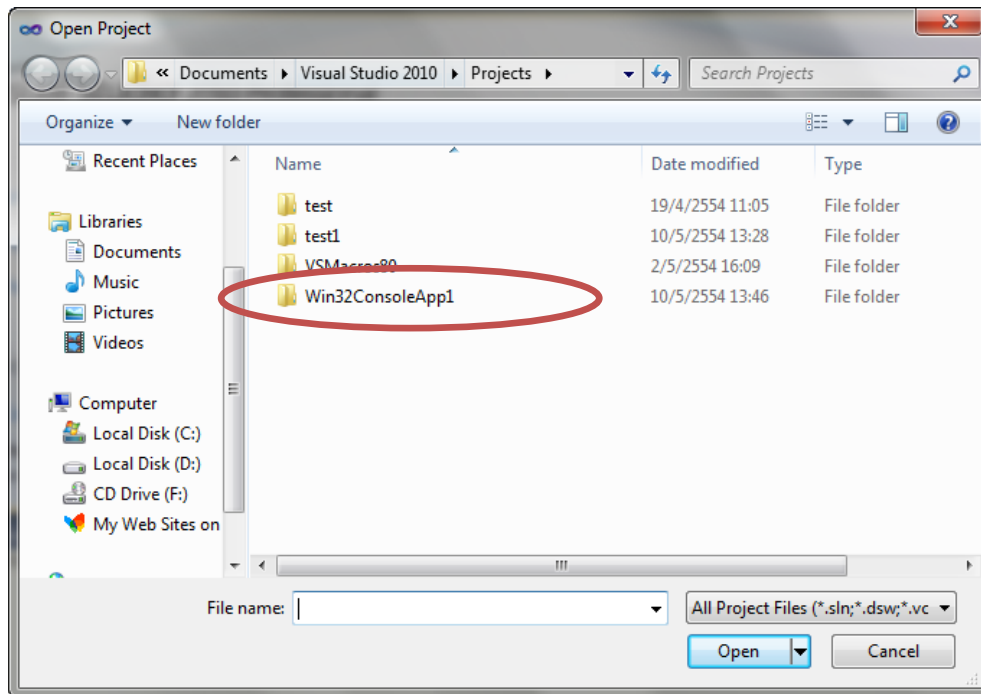
จากรูปจะเห็นว่าจะมีการสร้าง Folder หลักชื่อ Win32ConsoleApp2 ขึ้นตามชื่อ Project แล้วภายในก็
จะเป็นไฟล์ข้อมูลที่เกี่ยวข้องกับ Project อยู่ทั้งหมดรวมกัน ไม่มีการแบ่ง Folder ย่อย

หมายเหตุ การเลือก Project ผิดจะไม่สามารถ compile หรือ run ได้

การเปิดและปิด Project File

สำหรับการปิด Project File กรณีที่ไม่ได้ใช้งานแล้ว แต่ไม่ปิดโปรแกรม Microsoft Visual Studio 2010 จะเป็นการปิด Solution โดยทำการเลือกที่เมนู File → Close Solution

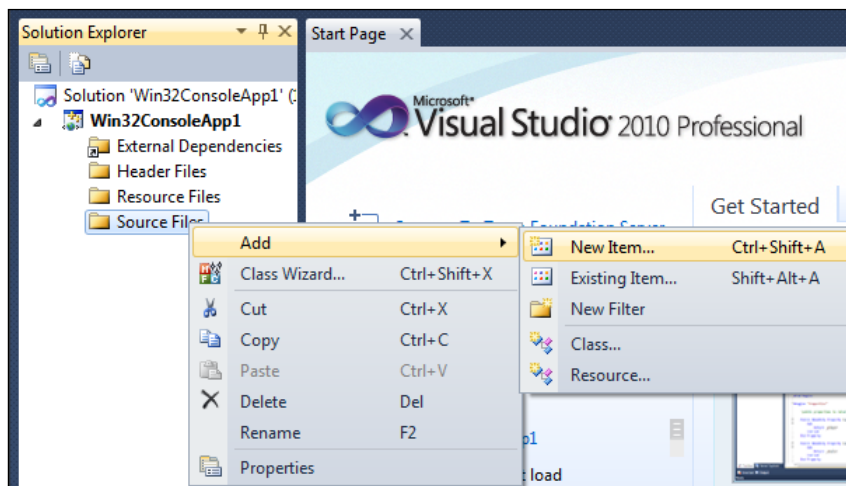
ส่วนการเปิด Project ที่ได้สร้างแล้ว ให้คลิกเลือกที่ Open Project จะปรากฏหน้าจอ Open Project ให้เลือกไปที่ชื่อ Folder ของ Project ที่ต้องการเปิด ดังรูปที่ 1-10 ในที่นี้ให้เลือกที่ Win32ConsoleApp1 จะปรากฏไฟล์ Solution มาให้เลือกที่มีชื่อเดียวกับ Project เมื่อคลิกเลือกที่ชื่อไฟล์นั้นแล้ว ให้กดปุ่ม  ก็จะเป็นการเปิดไฟล์ Project จะได้หน้าจอเหมือนกับตอนที่สร้าง Project ใหม่



รูปที่ 1-10 แสดงหน้าจอการเปิด Project file ที่สร้างไว้แล้ว

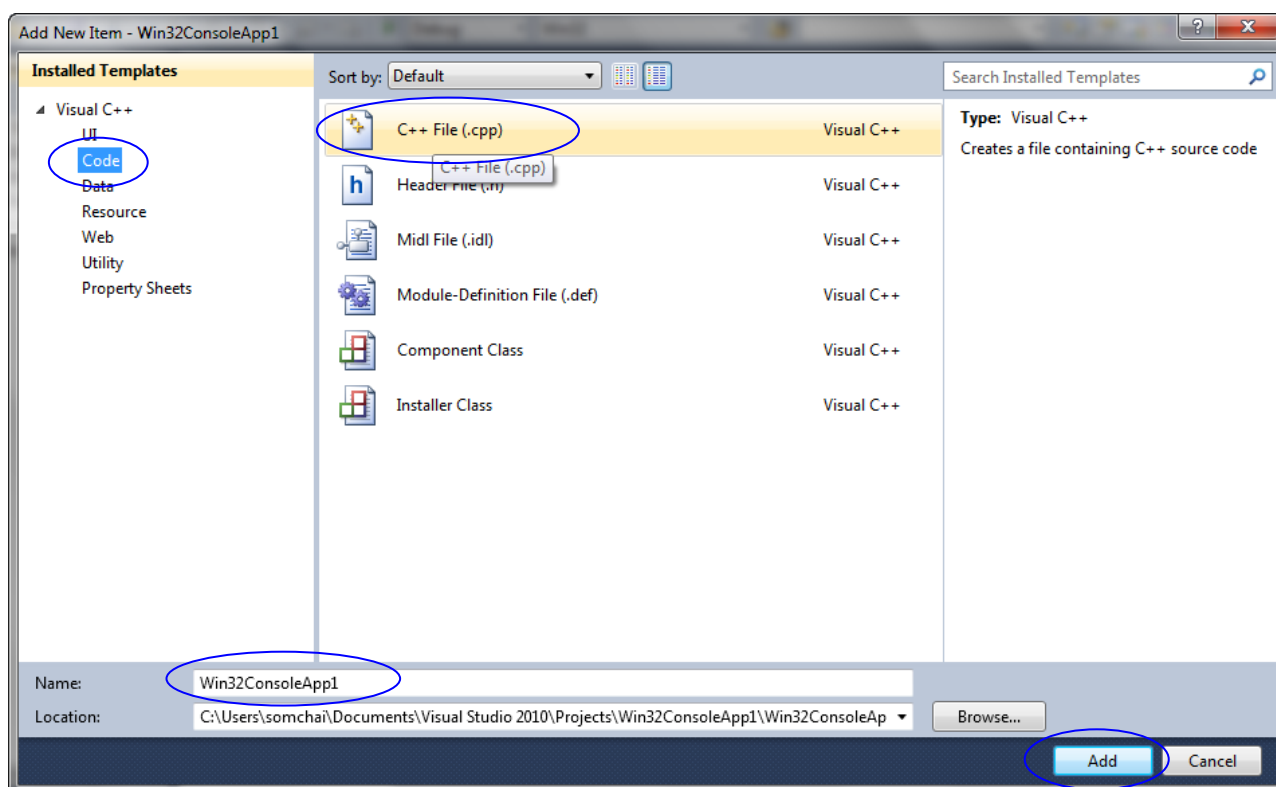
การสร้างไฟล์โปรแกรมภาษา C++

เมื่อสร้าง Project ได้แล้ว ก็จะเริ่มสร้างหรือนำไฟล์ภาษา C++ ที่มีนามสกุล .cpp เข้ามาใน Project โดยเริ่มจากการสร้างไฟล์ .cpp ใหม่ ให้คลิกเมาส์ปุ่มขวาที่ Folder ชื่อ Source files ในส่วน Solution Explorer ก็ จะปรากฏเมนูย่อยขึ้น ดังรูปที่ 1-11 ให้ทำการเลือกไปที่เมนู Add → New Item...

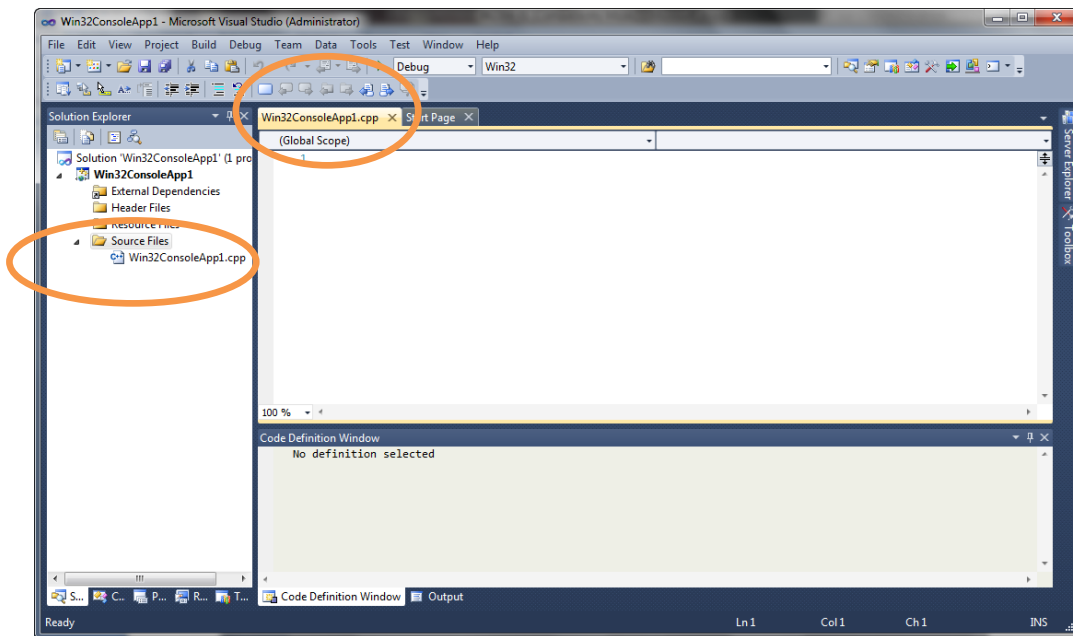


รูปที่ 1-11 แสดงหน้าจอสำหรับสร้างหรือไฟล์ภาษา cpp ใน Project

ก็จะปรากฏหน้าจอให้สร้างไฟล์ใหม่ ตามรูปที่ 1-12 จากนั้นทำการคลิกเลือกกลุ่ม Code และคลิกเลือกที่ C++ File(.cpp) เพื่อสร้างไฟล์ภาษา C++ โดยใส่ชื่อไฟล์ว่า Win32ConsoleApp1 หรือชื่ออื่นๆ ตามต้องการ (ไม่จำเป็นต้องชื่อเดียวกับ Project) เมื่อเสร็จแล้วก็ทำการกดปุ่ม **Add** ก็จะปรากฏพื้นที่ว่างทางด้านขวามือตามชื่อที่กำหนด ซึ่งแสดงว่าพร้อมที่จะให้เขียนโปรแกรมได้ ดังรูปที่ 1-13



รูปที่ 1-12 แสดงหน้าจอรายละเอียดการสร้างไฟล์ภาษา C++

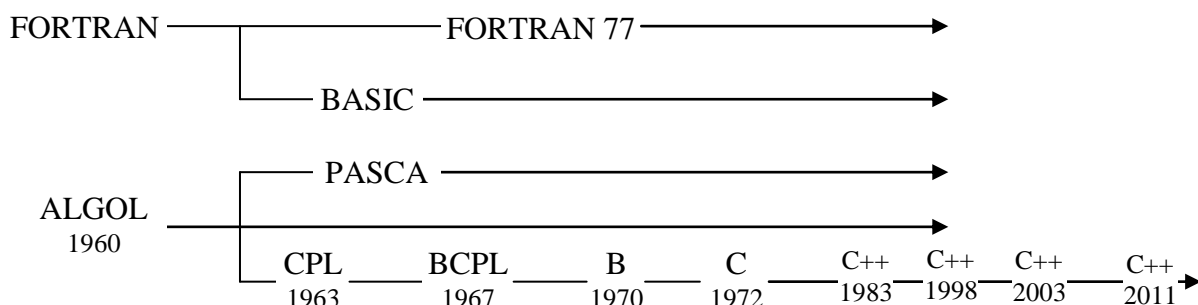


รูปที่ 1-13 แสดงหน้าจอหลังจากได้สร้างไฟล์ cpp เสร็จแล้ว และพร้อมที่จะรับการเขียนโปรแกรม C++

ประวัติภาษา C++

ภาษา C++ เริ่มเกิดขึ้น ในช่วงปี พ.ศ. 2523 (ค.ศ. 1980) ซึ่งพัฒนาจากภาษา C (ภาษา C มีต้นกำเนิดมาจาก 2 ภาษา คือ ภาษา BCPL และ ภาษา B) โดยนาย Bjarne Stroustrup ที่ห้องวิจัย Bell Laboratories บริษัท AT&T ในช่วงนั้นเป็นการเพิ่มคลาสให้กับภาษา C จึงเรียกว่า C with classes และใช้เฉพาะภายในหน่วยงานของบริษัท AT&T เท่านั้น เพื่อช่วยสนับสนุนการทำ Data Hiding และ Encapsulation ซึ่งเป็นคุณสมบัติประการหนึ่งของการเขียนโปรแกรมเชิงวัตถุ(OOP) หลังจากนั้นนายเบียร์นี สตาร์สตรูป พัฒนาภาษา C with classes ร่วมกับภาษา Simula ด้วยการใช้แนวคิดเกี่ยวกับคลาส และคุณสมบัติของการเขียนโปรแกรมเชิงวัตถุเข้าด้วยกัน

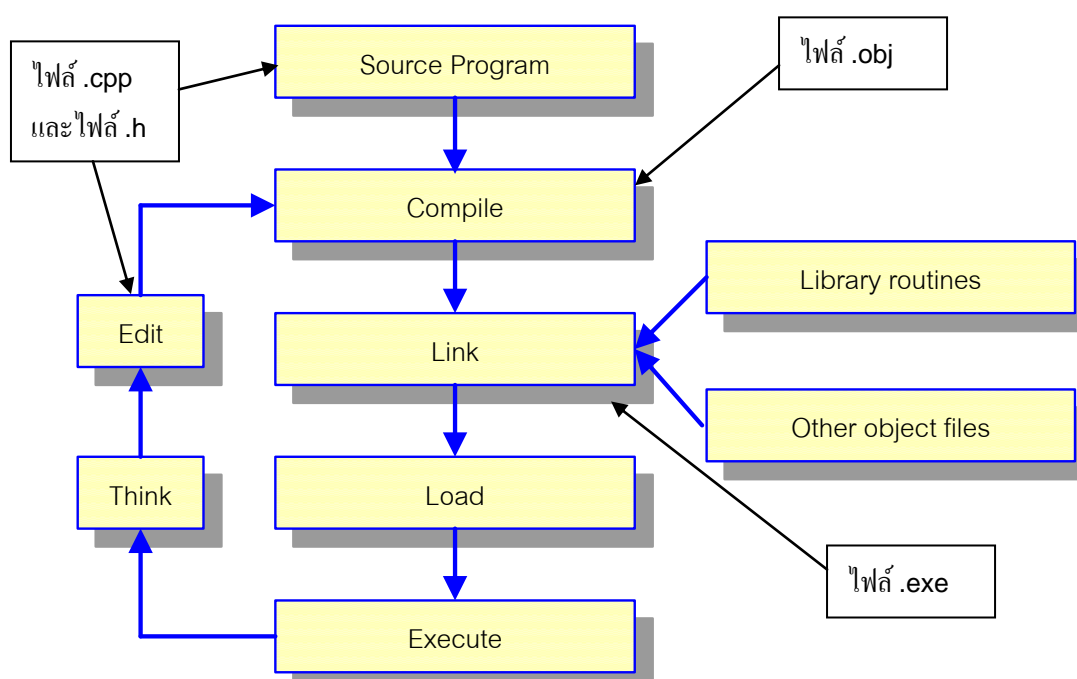
ในปี พ.ศ. 2526 (ค.ศ. 1983) ได้ตั้งชื่อใหม่เป็นภาษา C++ ต่อจากนั้นได้มีการพัฒนามาตลอด โดยการเพิ่มคุณสมบัติเด่นเข้ามา เช่น การเพิ่มการตรวจดักจับข้อผิดพลาด(Exception) การเพิ่มเทมเพลต(Template) เป็นต้น จากนั้นก็ได้มีการพัฒนาเพื่อทำให้ภาษา C++ เป็นมาตรฐานสากลและมาตรฐานอเมริกา(ISO/ANSI) มีชื่อเต็มคือ ISO/IEC 14882:1998 หรือเรียกย่อๆว่า C++ 98 ซึ่งเป็นมาตรฐานสากลแรกที่เผยแพร่สู่สาธารณชนอย่างเป็นทางการในปี พ.ศ. 2541 (ค.ศ. 1998) และเป็นที่นิยมไปทั่วโลกประมาณหนึ่งในสามของนักเขียนโปรแกรมคอมพิวเตอร์ทั้งหมด ต่อมาในปี 2003 ได้มีการกำหนดมาตรฐานเป็น ISO/IEC 14882:2003 โดยต่อมาในปี 2007 ได้กำหนดเป็น C++ Technical Report 1 (TR1) เป็นมาตรฐาน ISO/IEC TR 19768:2007 ที่มีการเพิ่มเติมไลบรารี แต่ไม่ถือเป็นมาตรฐานอ้างอิง และปัจจุบันมาตรฐานของภาษา C++ ได้กำหนดขึ้นใหม่ตาม ISO ในเดือนกันยายน 2011 คือ ISO/IEC 14882:2011 (เป็นที่รู้จักกันคือ C++11)



รูปที่ 1-14 แสดงประวัติการพัฒนาภาษา C++

ขั้นตอนการพัฒนาโปรแกรมด้วยภาษา C++

1. Editing เป็นขั้นตอนการสร้างและแก้ไขไฟล์ซอร์สโค้ดโปรแกรม ที่เป็นไฟล์ข้อความทั้งไฟล์นามสกุล .cpp และ ไฟล์นามสกุล .h สามารถมีได้มากกว่า 1 ไฟล์
2. Compiling เป็นขั้นตอนการแปลงไฟล์ซอร์สโค้ดโปรแกรม(cpp)ให้เป็นไฟล์ออบเจกต์(object) ด้วยตัวโปรแกรมคอมไพล์(Compiler) ซึ่งจะได้ไฟล์ออบเจกต์(.obj)
3. Linking เป็นขั้นตอนถัดมา โดยโปรแกรมเชื่อมโยง(Linker) เพื่อทำการเชื่อมโยงไฟล์ออบเจกต์ที่ได้มากับไฟล์ไลบรารีของระบบ ให้กลายเป็นไฟล์ที่สามารถเรียกใช้งานได้(exe)
4. Loading and executing เป็นขั้นตอนเมื่อมีการเรียกใช้งานไฟล์ exe โดยจะเกิดการโหลดไฟล์ exe ไว้ในหน่วยความจำ แล้วจึงทำงานตามคำสั่งในโปรแกรม
5. Viewing the behavior of the program เป็นการแสดงผลการทำงานของโปรแกรม



รูปที่ 1-15 แสดงขั้นตอนการพัฒนาโปรแกรมด้วยภาษา C++

สามารถดูข้อมูลเพิ่มเติมได้ที่ <http://en.wikipedia.org/wiki/C++>

โครงสร้างโปรแกรมของภาษา C++

โปรแกรมภาษา C++ มีโครงสร้างการเขียนโปรแกรมเหมือนภาษา C ซึ่งภาษา C++ แต่สามารถเขียนโปรแกรมทั้งในแบบโครงสร้างและแบบเชิงวัตถุได้ โดยมีลักษณะรูปแบบเริ่มต้นเป็นลักษณะฟังก์ชัน และต้องมีฟังก์ชันหลักหรือฟังก์ชันแรกชื่อ main ที่สะกดด้วยตัวอักษรเล็กหมด ซึ่งโครงสร้างของโปรแกรมจะประกอบด้วย 2 ส่วนหลักคือ ส่วนการประกาศการใช้ฟังก์ชันและตัวแปร และตัวฟังก์ชัน main

สำหรับโปรแกรมภาษา C++ จะมีฟังก์ชันเดียวหรือหลายฟังก์ชันก็ได้ แต่ต้องมีฟังก์ชันชื่อ main เพียงฟังก์ชันเดียวเท่านั้น โดยโปรแกรมจะเริ่มทำงานจากฟังก์ชัน main ก่อน ส่วนฟังก์ชันอื่นๆ จะกำหนดไว้ก่อนหรือหลังฟังก์ชัน main ก็ได้ แต่แบบที่นิยมจะกำหนดไว้หลังฟังก์ชัน main

ส่วนการประกาศการใช้ฟังก์ชันและตัวแปร

จะประกอบด้วย

- การใช้ฟังก์ชันในไฟล์ไลบรารีมาตรฐาน C++ หรือไฟล์เฮดเดอร์ต่างๆ เรียกส่วนนี้ว่า Preprocessor และจะต้องมีไฟล์เฮดเดอร์ iostream เสมอ
- การใช้ตัวแปรโกลบอล หรือการประกาศรูปแบบฟังก์ชันย่อย ส่วนนี้จะมีหรือไม่มีก็ได้ ขึ้นอยู่กับโปรแกรมนั้นๆ
- การใช้เนมสเปซ(namespace) ถ้าฟังก์ชันนั้นถูกสร้างในเนมสเปซ จะต้องประกาศการใช้เนมสเปซนั้นก่อน จึงจะใช้ฟังก์ชันนั้นได้ เช่น ไลบรารีมาตรฐาน C++ ถูกสร้างอยู่ในเนมสเปซ std ฉะนั้นก่อนใช้งานจะต้องประกาศเนมสเปซ std ด้วยคำสั่ง using namespace std หรือกำหนดชื่อ namespace ด้วยเมื่อมีการเรียกใช้คำสั่ง เช่น

```
std::cout << "Hello World.";
```

ฟังก์ชัน main

เป็นฟังก์ชันหลักที่ต้องมีอยู่เสมอในโปรแกรม ฟังก์ชัน main เป็นทางเข้าของโปรแกรมและมีหน้าที่ทำงานตามคำสั่งที่เขียน หรือไปเรียกฟังก์ชันอื่นๆ และแสดงผลลัพธ์มาที่หน้าจอ และสุดท้ายจะคืนค่าเป็น 0 เพื่อบอกว่าโปรแกรมนี้ได้ทำงานเสร็จแล้ว โปรแกรมก็จะถูกปิดลงอัตโนมัติ

Preprocessor directive	
Using directive	
Global Function or Variable declarations	
<pre>int main() { statements; return(0); }</pre>	<pre>void main() { statements; }</pre>

รูปที่ 1-16 แสดงโครงสร้างของโปรแกรมภาษา C++

แต่ละฟังก์ชันจะประกอบด้วยชื่อฟังก์ชันและตัวฟังก์ชันจะเริ่มต้นด้วยเครื่องหมาย { และจนถึงเครื่องหมาย } โดยทุกคำสั่ง(statement) จะต้องอยู่ภายในเครื่องหมายวงเล็บปีกกา คือ { และ } ของแต่ละฟังก์ชัน และในแต่ละคำสั่งนั้นจะต้องถูกปิดด้วยเครื่องหมายเซมิโคลอน (;)

การทดลองที่ 1_1 ให้นักศึกษาพิมพ์โปรแกรมตามรายละเอียดด้านล่างนี้

```
1 #include <iostream>
2 void main()
3 {
4     cout << "Hello World.";
5 }
```

บันทึกผลการทดลอง

การทดลองที่ 1_2 ให้นักศึกษาพิมพ์โปรแกรมตามรายละเอียดด้านล่างนี้

```
1 #include <iostream>
2 void main()
3 {
4     std::cout << "Hello World.";
5 }
```

บันทึกผลการทดลอง

การทดลองที่ 1_3 ให้นักศึกษาพิมพ์โปรแกรมตามรายละเอียดด้านล่างนี้

```
1 #include <iostream>
2 using namespace std;
3 void main()
4 {
5     cout << "Hello World.";
6 }
```

บันทึกผลการทดลอง

คำถาม

สังเกตผลลัพธ์ที่ได้จากการทดลองที่ 1_2 และ 1_3 มีแตกต่างหรือเหมือนกันอย่างไร

เพราะอะไรการทดลองที่ 1_2 และ 1_3 จึงไม่เกิด Syntax Error และสามารถแสดงผลลัพธ์ได้

การทดลองที่ 1_4 ให้นักศึกษาพิมพ์โปรแกรมตามรายละเอียดด้านล่างนี้

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << "Hello World.";
6     return(0);
7 }
```

บันทึกผลการทดลอง

คำถาม

ผลลัพธ์ที่ได้จากการทดลองที่ 1_3 และ 1_4 มีความแตกต่างหรือเหมือนกันอย่างไร

จากการทดลองที่ 1_3 และ 1_4 มีคำสั่งโปรแกรมอะไรที่แตกต่างกัน

การทดลองที่ 1_5 ให้นักศึกษาพิมพ์โปรแกรมตามรายละเอียดด้านล่างนี้

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << "Welcome to ";
6     cout << "C++ language.";
7     return (0);
8 }
```

บันทึกผลการทดลอง

คำถาม

ถ้าหากต้องการผลลัพธ์ที่ปรากฏบนหน้าจอในหนึ่งบรรทัดจำเป็นต้องใช้คำสั่ง cout เพียงคำสั่งเดียวหรือไม่

การทดลองที่ 1_6 โปรแกรมแสดงข้อความ “This is structure of C++ language.” ที่หน้าจอ

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << "This is structure of C++ language." << endl;
6     return (0);
7 }
```

บันทึกผลการทดลอง

การทดลองที่ 1_7 โปรแกรมแสดงข้อความ “This is structure of C++ language.” ที่หน้าจอ

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << "This is structure of " << endl;
6     cout << "C++ language." << endl;
7     return (0);
8 }
```

บันทึกผลการทดลอง

คำถาม

ผลลัพธ์ที่ได้จากการทดลองที่ 1_6 และ 1_7 แตกต่างหรือเหมือนกันอย่างไร

ถ้าหากคำสั่ง cout ในบรรทัดที่ 5 ไม่มีการใช้ endl ปิดท้ายคำสั่ง ผลลัพธ์ที่ได้จะเป็นอย่างไร

คำสั่ง endl มีหน้าที่ทำอะไร

หากต้องการให้ผลลัพธ์ในการทดลองที่ 1_7 มาจากการใช้คำสั่ง cout เพียงคำสั่งเดียว จะต้องเขียนคำสั่งอย่างไร

แบบฝึกหัดท้ายบท

1. ให้นักศึกษาวิเคราะห์และตรวจสอบโปรแกรมข้างล่างว่ามีข้อผิดพลาด(syntax error) หรือไม่ ถ้าไม่จะได้ผลลัพธ์อย่างไร

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << "I love C++" << endl << " very much." << endl;
6     return 0;
7 }
```

Compile

Build

บันทึกผลการทดลอง

2. ให้นักศึกษาวิเคราะห์และตรวจสอบโปรแกรมข้างล่างว่ามีข้อผิดพลาด(syntax error) หรือไม่ ถ้าไม่จะได้ผลลัพธ์อย่างไร

```
1 #include <iostream>
2 int Main()
3 {
4     std::cout << "I like Information Technology \n." << ;
5     std::cout << "I want to Programmer." << endl;
6     return 0;
7 }
8
```

Compile

Build

บันทึกผลการทดลอง

3. ให้นักศึกษาเขียนโปรแกรมภาษา C++ เพื่อแสดงข้อความที่ประกอบด้วยรหัสประจำตัว ชื่อ และนามสกุลของนักศึกษาบนหน้าจอ เมื่อรันโปรแกรมปรากฏผลลัพธ์ดังนี้

```
Student Code : 55-060216-9999-9
Student Name : Somchai Cheingpongpan
Division : Information Technology
Year : 1
Room : RB
Sex : Male
Age : 18
```

4. ให้นักศึกษาเขียนโปรแกรมภาษา C++ เพื่อแสดงตัวอักษร A ขนาด 7x6 เมื่อรันโปรแกรมปรากฏผลลัพธ์ดังนี้

```
  #
 # #
#   #
#####
#   #
```

