

# วิชา การพัฒนาโปรแกรม (ภาษาซี)

รหัสวิชา ว31293

หน่วยการเรียนรู้ที่ 1 ความรู้เบื้องต้นเกี่ยวกับภาษาซี

## สาระการเรียนรู้

1. กำเนิดภาษาซี
2. คุณสมบัติที่โดดเด่นของภาษาซี
3. โครงสร้างภาษาซี
4. โปรแกรม Dev - C++

## ผลการเรียนรู้

สามารถอธิบายประวัติความเป็นมา ลักษณะการทำงานภาษาซี และทำการติดตั้ง โปรแกรม เพื่อการใช้งานต่อไปได้

## จุดประสงค์การเรียนรู้

1. บอกประวัติความเป็นมา และลักษณะของภาษา C ได้
2. สามารถค้นหาและดาวน์โหลดโปรแกรมภาษา C ได้
3. สามารถใช้งานโปรแกรมภาษา C ได้



## ใบความรู้ที่ 1 เรื่อง เริ่มต้นกับภาษาซี

จัดทำโดย นางพรพนารัตน์ ชมภูนุช

### กำเนิดภาษาซี



ภาษา C เป็นภาษาคอมพิวเตอร์ที่ถูกค้นคิดขึ้นโดย Denis Ritchie ในปี ค.ศ. 1972 โดยใช้ระบบปฏิบัติการของยูนิกซ์ (UNIX) นับจากนั้นมาก็ได้รับความนิยมเพิ่มขึ้นจนถึงปัจจุบัน ภาษา C สามารถติดต่อในระดับฮาร์ดแวร์ได้ดีกว่าภาษาระดับสูงอื่น ๆ ภาษา C เป็นภาษาคอมพิวเตอร์ที่แปลภาษาด้วยคอมไพเลอร์ (Compiler) ทำหน้าที่ในการคอมไพล์ (Compile) หรือแปลงคำสั่งทั้งหมดในโปรแกรมให้เป็นภาษาเครื่อง (Machine Language) เพื่อให้เครื่องคอมพิวเตอร์นำคำสั่งเหล่านั้นไปทำงานต่อไป

ภาษา C เป็นภาษาที่มีความใกล้เคียงกับภาษาระดับต่ำ (Low-Level Language) จึงทำให้นักพัฒนาโปรแกรมสามารถที่จะกำหนดรายละเอียดของโปรแกรมให้เข้าถึงการทำงานในส่วนต่าง ๆ ของคอมพิวเตอร์ให้มากที่สุดเพื่อให้เกิดความเร็วในการทำงานสูงสุด และในขณะเดียวกันภาษา C ก็ยังมีความเป็นภาษาระดับสูง (High-Level Language) ทำให้ผู้พัฒนาสามารถที่จะพัฒนาโปรแกรมได้โดยเน้นไปที่การแก้ปัญหาที่ต้องการได้อย่างอิสระโดยไม่ต้องคำนึงถึงฮาร์ดแวร์ใด ๆ

ภาษาซีเป็นภาษาโปรแกรมคอมพิวเตอร์ที่ได้รับความนิยมและมีการพัฒนาอย่างต่อเนื่อง ต่อมา Mr. Bjarne Stroustrup นักวิจัยและพัฒนาของศูนย์วิจัยเบล (Bell Laboratories) ได้พัฒนาภาษา C++ (ซีพลัสพลัส) ขึ้นมา โดยที่ภาษา C++ มีความสามารถในการทำงานได้ทุกอย่างเหมือนกับภาษาซี ซึ่งมีรูปแบบและโครงสร้างของภาษาใกล้เคียงกัน แต่ภาษา C++ ใช้หลักการออกแบบโปรแกรมเชิงวัตถุ (Object Oriented Design) ในขณะที่ภาษาซีใช้หลักการออกแบบโปรแกรมแบบโมดูลาร์ (Modular Design)

### คุณสมบัติที่โดดเด่นของภาษาซี

ภาษาซีเป็นภาษาที่โดดเด่นกว่าภาษาระดับสูงทั่วไปในหลาย ๆ ด้านด้วยกัน ซึ่งประกอบด้วย

1. เป็นภาษาที่ไม่ขึ้นกับฮาร์ดแวร์และระบบปฏิบัติการ
2. เป็นภาษาที่มีความยืดหยุ่นสูงมาก
3. มีประสิทธิภาพสูง เมื่อเปรียบเทียบกับภาษาระดับสูงอื่น ๆ จะเห็นว่าชุดคำสั่งมีความ

กะทัดรัด และกระชับ รวมถึงประมวลผลได้อย่างรวดเร็ว

4. มีความสามารถในการโปรแกรมโมดูล คือสามารถแบ่งโมดูลเพื่อแยกคอมไพล์ได้ และสามารถทำการลิงค์เชื่อมโยงภาษาเข้าด้วยกันได้
5. มีตัวแปรชนิดพอยเตอร์ ที่สามารถเข้าถึงหรือชี้ไปยังตำแหน่งที่อยู่ของหน่วยความจำที่จัดเก็บข้อมูลโดยตรง
6. ภาษาซีมองตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่แตกต่างกัน (Case Sensitive)

## โครงสร้างภาษาซี

ภายในโปรแกรมภาษา C จะแบ่งออกเป็น 5 ส่วน ดังนี้

1. ส่วนหัวของโปรแกรม (Preprocessor Directive)
2. ส่วนของการกำหนดค่าภายนอก (Global Declarations)
3. ส่วนของฟังก์ชันหลัก (The main function)
4. ส่วนของฟังก์ชันย่อยที่ผู้ใช้สร้างขึ้นเอง (User-define function)
5. ส่วนอธิบายโปรแกรม (Program Comment)

### 1. ส่วนหัวของโปรแกรม

ส่วนหัวของโปรแกรมนี้ เรียกว่า 프리โพรเซสเซอร์ไดเรกทีฟ (Preprocessor Directives) เป็นคำสั่ง เริ่มต้นในการกำหนดไฟล์ที่จะนำมาประมวลผลหรือ ที่เรียกว่า **เฮดเดอร์ไฟล์ (Header File)** เพื่อเป็นการกำหนดค่าเริ่มต้นให้กับโปรแกรม Compiler โดยเริ่มต้นด้วยการใช้เครื่องหมายไดเรกทีฟ (Directive) นั่นก็คือเครื่องหมาย # ตามด้วยคำสั่ง include และชื่อของเฮดเดอร์ไฟล์ แต่ไม่ต้องลงท้ายด้วยเครื่องหมาย ; ซึ่งเป็นไฟล์ที่เก็บไลบรารีมาตรฐานของภาษา C ซึ่งจะถูกลิงเข้ามารวมกับโปรแกรมในขณะที่กำลังทำการคอมไพล์ เหมือนคำสั่งอื่นทั่วไป คำสั่งฟรีโพรเซสเซอร์ไดเรกทีฟ ที่ใช้ในการประกาศ เฮดเดอร์ไฟล์ที่ส่วนใหญ่ใช้กันทั่วไปคือ คำสั่ง #include โดยมีรูปแบบการใช้คำสั่งดังนี้

#### รูปแบบที่ 1

```
#include <ชื่อเฮดเดอร์ไฟล์>
```

#### รูปแบบที่ 2

```
#include "ชื่อเฮดเดอร์ไฟล์"
```

#### ตัวอย่าง

```
#include <stdio.h>
```

```
#include "stdio.h"
```

## 2. การประกาศตัวแปรภายนอก (Global Variables)

เป็นส่วนที่ใช้ประกาศตัวแปรหรือค่าต่าง ๆ ที่ให้ใช้ได้ทั้งโปรแกรม ซึ่งใช้ได้ทั้งโปรแกรมซึ่งในส่วนไม่จำเป็นต้องมีก็ได้

## 3. ส่วนฟังก์ชันหลัก

เป็นส่วนที่ผู้ใช้ต้องเขียนขึ้นเองโดยนำเอาคำสั่งหรือฟังก์ชันมาตรฐานต่าง ๆ มาเรียบเรียงกันขึ้นเป็นโปรแกรม เพื่อสั่งให้คอมพิวเตอร์รับข้อมูลจากอุปกรณ์อินพุตประมวลผลแล้วก็ให้ผลลัพธ์ตามที่เราต้องการ ส่วนมาจะเขียนขึ้นต้นด้วย main() ซึ่งหมายถึงการเรียกใช้ฟังก์ชัน main() นั้นเอง และตามด้วยโปรแกรมหลักที่ประกอบด้วยคำสั่งต่าง ๆ ภายในเครื่องหมายวงเล็บปีกกา { } ซึ่งภายในนี้จะประกอบด้วยส่วนต่าง ๆ ดังนี้

### 3.1 ส่วนตัวแปรแบบภายใน (Local Variables)

เป็นส่วนที่ใช้สำหรับประกาศตัวแปรที่จะใช้ในเฉพาะฟังก์ชันของตนเอง ฟังก์ชันอื่นไม่สามารถเข้าถึงหรือใช้ได้ ซึ่งจะต้องทำการประกาศตัวแปรก่อนการใช้งานเสมอ และจะต้องประกาศไว้ในส่วนนี้เท่านั้น

### 3.2 ตัวโปรแกรม (Statements)

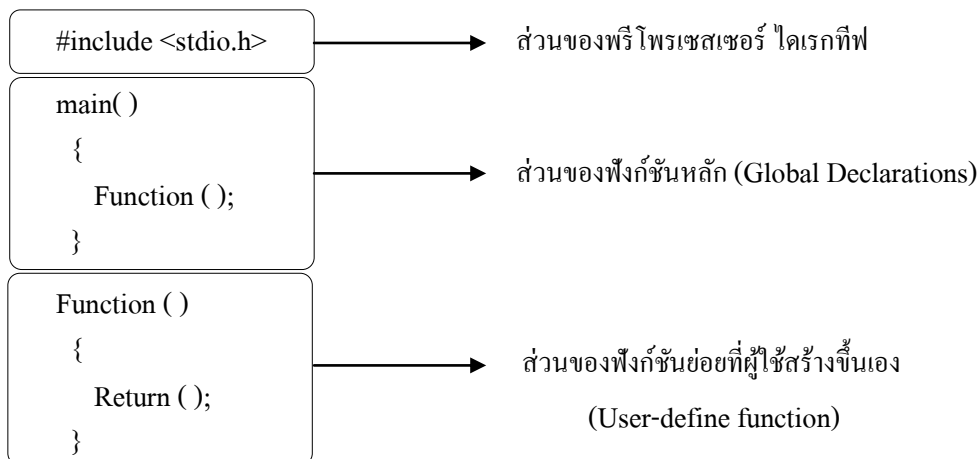
เป็นส่วนที่อยู่ถัดลงมาจากส่วนตัวแปรภายใน ซึ่งประกอบไปด้วยคำสั่งต่าง ๆ ของภาษา C และคำสั่งต่าง ๆ จะใช้เครื่องหมาย ; เพื่อเป็นการบอกให้รู้ว่าจบคำสั่งหนึ่ง ๆ แล้ว ส่วนใหญ่ คำสั่งต่าง ๆ ของภาษา C เขียนด้วยตัวพิมพ์เล็ก เนื่องจากภาษา C จะแยกความแตกต่างของตัวพิมพ์เล็กและพิมพ์ใหญ่ หรือ Case Sensitive นั้นเอง นอกจากนี้ภาษา C ยังไม่สนใจกับการขึ้นบรรทัดใหม่ เพราะฉะนั้นพิมพ์คำสั่งหลายคำสั่งในบรรทัดเดียวกันได้ โดยไม่มีเครื่องหมาย ; เป็นตัวจบคำสั่ง

### 3.3 ส่วนคำอธิบายโปรแกรม (Comment Lines)

ส่วนนี้ใช้ในการอธิบายโปรแกรมอาจพิมพ์ข้อความ สูตรหรือ อื่น ๆ ที่เกี่ยวข้องกับโปรแกรม ส่วนของโปรแกรมส่วนนี้จะมีหรือไม่มีก็ได้ ถ้าต้องเขียนให้เริ่มต้นด้วยเครื่องหมาย /\* ตามด้วยข้อความที่ต้องการ แล้วปิดท้ายด้วย \*/ (เครื่องหมาย / และ \* ต้องเขียนติดกันเสมอ)

#### 4. ส่วนของฟังก์ชันย่อยที่ผู้ใช้สร้างขึ้นเอง (User-define function)

เป็นการเขียนคำสั่งและฟังก์ชันต่าง ๆ ขึ้นมาใช้ในโปรแกรม โดยต้องอยู่ในเครื่องหมาย {} และการสร้างฟังก์ชันหรือคำใหม่ที่ทำให้การทำงานตามที่เรากำลังต้องการให้กับโปรแกรมและสามารถเรียกใช้ได้ ภายในโปรแกรม



รูปแสดง โครงสร้างการเขียนโปรแกรมแบบมี ส่วนของฟังก์ชันย่อยที่ผู้ใช้สร้างขึ้นเอง (User-define function)

#### 5. ส่วนอธิบายโปรแกรม (Program Comment)

เป็นส่วนที่ใช้สำหรับแสดงข้อความ เพื่ออธิบายสิ่งที่ต้องการในโปรแกรม ซึ่งจะใช้เครื่องหมาย `/*` และ `*/` ปิดหัวและปิดท้ายของข้อความที่ต้องการ หรือหากต้องการเขียนคำอธิบายเพียงบรรทัดเดียวให้ใช้เครื่องหมาย `//` กำกับไว้ข้างหน้าบรรทัดนั้น เมื่อมีการคอมไพล์โปรแกรม ข้อความในส่วนนี้จะถูกมองข้ามไป

**รูปแบบที่ 1** ใช้กำกับการอธิบายเพียงบรรทัดเดียว

```
// ข้อความอธิบายโปรแกรมหรือคำสั่งที่ต้องการละเว้นจากการตรวจสอบ
```

**รูปแบบที่ 2** ใช้กำกับการอธิบายหลายบรรทัด

```
/*
ข้อความอธิบายโปรแกรมหรือคำสั่งที่ต้องการละเว้นจากการตรวจสอบ
...
...
*/
```

## ศึกษาสังเกต

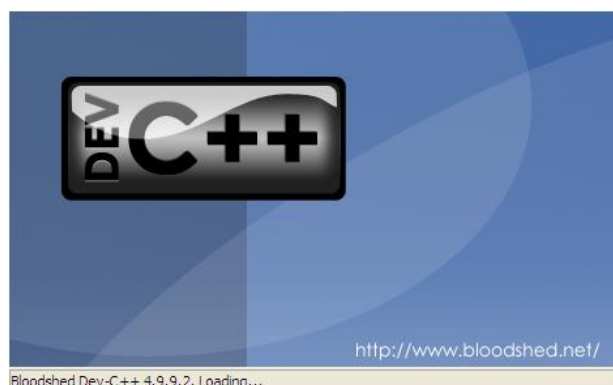
กฎเกณฑ์ที่ต้องรู้ในการเริ่มต้นฝึกหัดเขียนโปรแกรมภาษาซี

1. ที่ส่วนหัวโปรแกรม จะต้องกำหนดตัวประมวลผลก่อนเสมอ
2. ชุดคำสั่งในภาษาซี จะใช้อักษรพิมพ์เล็กทั้งหมด
3. ตัวแปรที่ใช้งาน ต้องถูกประกาศชนิดข้อมูลไว้เสมอ
4. ภายในโปรแกรม จะต้องมียังน้อย 1 ฟังก์ชันเสมอ ซึ่งก็คือ ฟังก์ชัน `main()` นั่นเอง

## โปรแกรม Dev-C++

### 1. แนะนำโปรแกรม Dev-C++

การเขียนโปรแกรมคอมพิวเตอร์ภาษาต่าง ๆ ปกติเขียนคำสั่งต่าง ๆ ด้วยโปรแกรมที่เรียกว่า editor ซึ่งอาจใช้โปรแกรมพวก word Processor เขียนโดยไม่ใช้รูปแบบพิเศษต่าง ๆ หรือใช้โปรแกรมที่มีรูปแบบพิเศษน้อย ๆ เช่น notepad เขียน แล้วเปลี่ยนภาษาที่เขียนนั้นเป็นภาษาเครื่องโดยใช้ compiler ของภาษาคอมพิวเตอร์ที่ใช้ เช่นคอมไพเลอร์ของภาษาซี ซึ่งก็มีผู้ผลิตหลายราย ซึ่งต้องเป็นไปตามมาตรฐาน ANSI C และมีรายละเอียดเพิ่มเติมแตกต่างกันไป

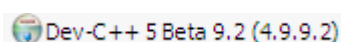


รูปภาพ แสดงตัวอย่างหน้าจอโปรแกรม Turbo C

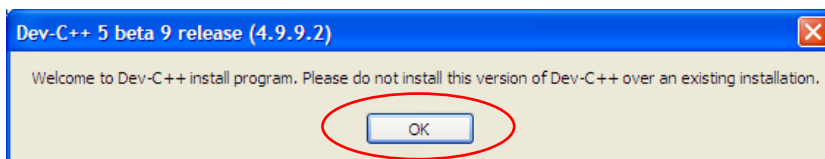
### 2. วิธีการติดตั้งโปรแกรม Dev-C++

เว็บไซต์ที่มีโปรแกรม Dev-C++ ไว้สำหรับการดาวน์โหลดเพื่อทำการติดตั้ง คือ [www.bloodshed.net](http://www.bloodshed.net) สามารถดาวน์โหลดโปรแกรม Dev-C++ ในเวอร์ชันต่าง ๆ ได้ และในขั้นตอนการติดตั้งโปรแกรม Dev-C++ เวอร์ชัน 4.9 มีขั้นตอนดังต่อไปนี้

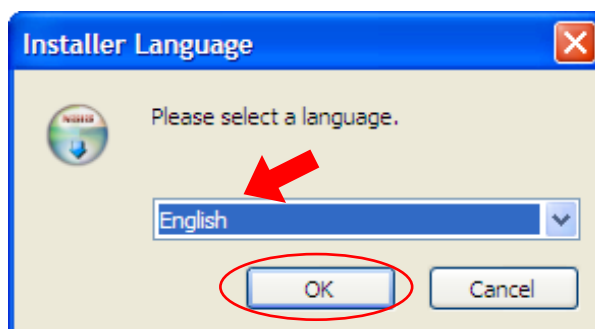
- 1) เมื่อเปิดไฟล์เครื่องโปรแกรม Dev-C++ ให้ทำการดับเบิลคลิกที่ไอคอนดังรูป



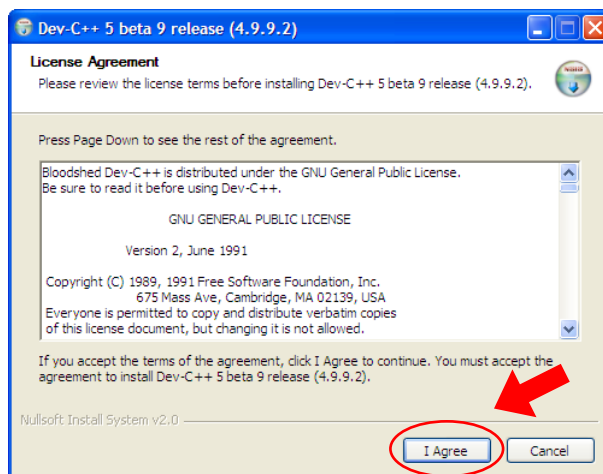
2) จะปรากฏหน้าต่างแสดงสถานการณ์ติดตั้งโปรแกรม แล้วทำการคลิกที่ปุ่ม OK ดังรูป



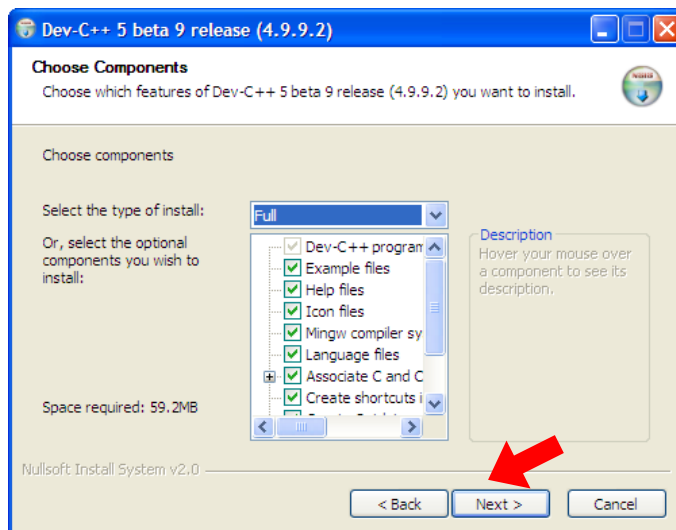
3) จะปรากฏหน้าต่างการติดตั้ง ในการเลือกภาษาที่ต้องการ ในที่นี้คือ English แล้วทำการคลิกที่ปุ่ม OK ดังรูป



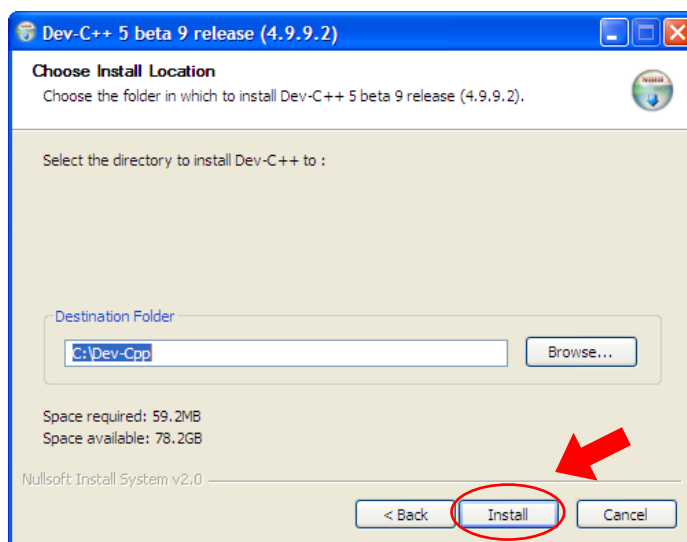
4) จะปรากฏหน้าต่างแสดงเงื่อนไขของการติดตั้ง ถ้ายอมรับเพื่อทำการติดตั้ง คลิกที่ปุ่ม I Agree ดังรูป



5) จะปรากฏหน้าต่างให้เลือก Components ในการทำงานของโปรแกรม ถ้าต้องการติดตั้ง Components ทั้งหมด ก็คลิกที่ปุ่ม Next ดังรูป

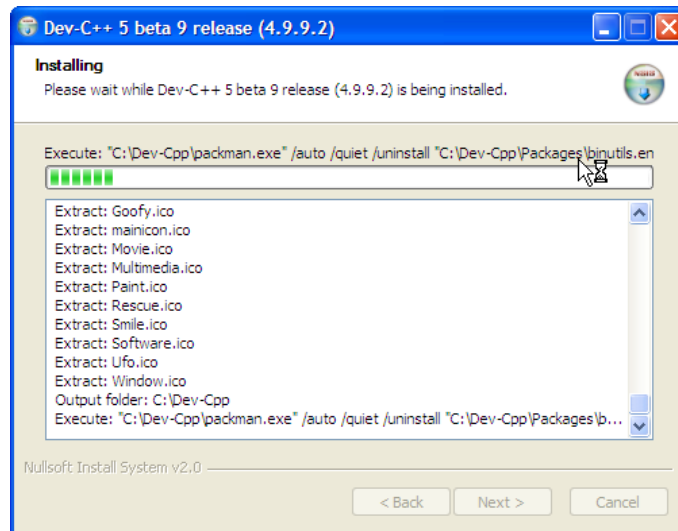


6) จะปรากฏหน้าต่างให้เลือกตำแหน่งในการติดตั้งโปรแกรม นั่นก็คือ หากต้องการกำหนดตำแหน่งการติดตั้งโปรแกรมในหน่วยความจำภายในเครื่องใหม่ หรือชื่อโฟลเดอร์ใหม่ ก็สามารถทำการเปลี่ยนไดเรกทอรีได้ ด้วยการคลิกที่ปุ่ม Browse หรือถ้าไม่ต้องการเปลี่ยนก็คลิกที่ปุ่ม Install ดังรูป



7) จะปรากฏหน้าต่างแสดงสถานการณ์ติดตั้งโปรแกรม ลงในหน่วยความจำภายในเครื่อง ดังรูป





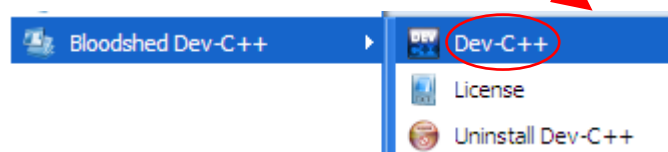
8) จะปรากฏหน้าต่าง ที่แสดงว่าได้ติดตั้ง โปรแกรมเสร็จแล้ว ให้คลิกที่ปุ่ม Finish ดังรูป



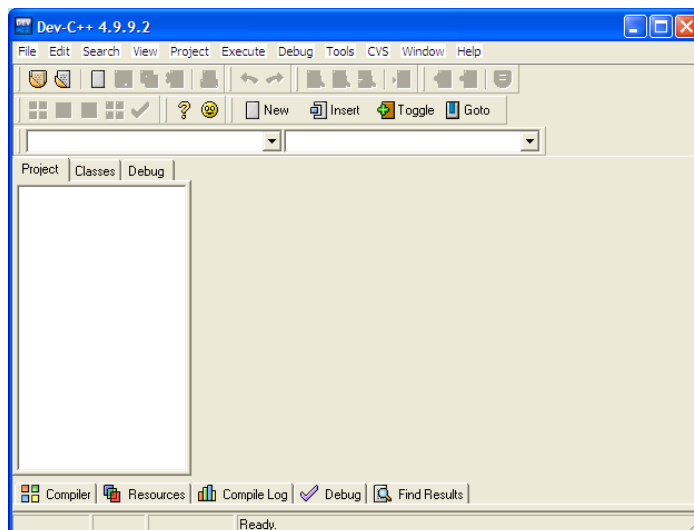
### 3. การเรียกใช้โปรแกรม Dev-C++

การเรียกใช้ Dev-C++ ทำได้ทำนองเดียวกับการเรียกใช้โปรแกรมอื่น ๆ

1) คลิกที่ปุ่ม Start > All Programs > Bloodshed Dev-C++ คลิกที่ Dev-C++ ดังรูป



2) จะได้นหน้าต่างโปรแกรม Dev-C++ ดังรูป



#### 4. ส่วนประกอบที่สำคัญของโปรแกรม Dev-C++

ส่วนประกอบที่สำคัญของโปรแกรม Dev-C++ ในการใช้งานเบื้องต้น มีดังนี้

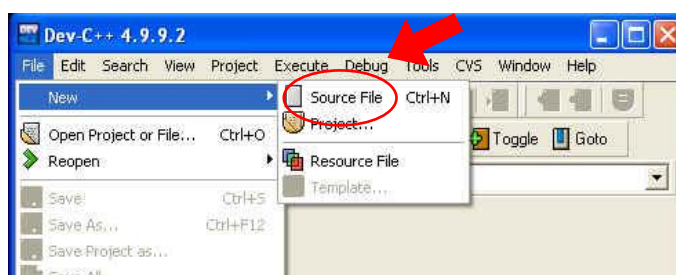
- 1) ส่วนของเมนู
- 2) ส่วนของ Editor Screen

##### 1) ส่วนของเมนู

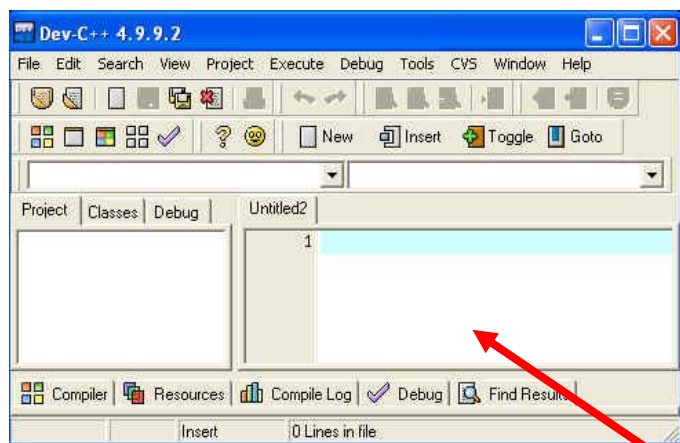
เป็นคำสั่งต่าง ๆ ที่จำเป็นสำหรับการใช้งานโปรแกรม เช่น การสร้างไฟล์ใหม่ การคอมไพล์ การรันโปรแกรม นอกจากนี้ในด้านล่างของเมนูยังมี แถบเครื่องมือ (Tool Bar) ที่รวบรวมคำสั่งต่าง ๆ ที่ใช้งานบ่อย ๆ ในรูปของไอคอนให้ผู้ใช้เลือกใช้งานได้ง่ายอีกด้วย ดังตัวอย่างการใช้งานเมนูที่สำคัญ ๆ ดังนี้

##### 1.1) เมนู File

- คำสั่ง **New** เมื่อเปิดโปรแกรมขึ้นมาครั้งแรก จะต้องเลือก เมนู File และเลือกคำสั่ง New เพื่อสร้างไฟล์ใหม่ โดยเลือกรายการย่อย Source File ดังรูป

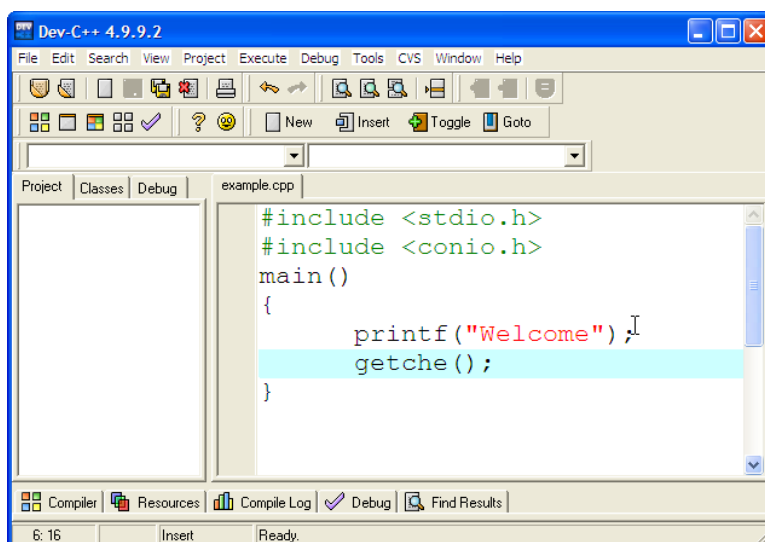


เมื่อเลือก คำสั่ง Source File (มีคีย์ลัด Ctrl+N) ใช้สำหรับสร้างไฟล์เพื่อเขียนคำสั่งในโปรแกรมที่จะสร้างขึ้น จะได้นหน้าต่างดังรูป

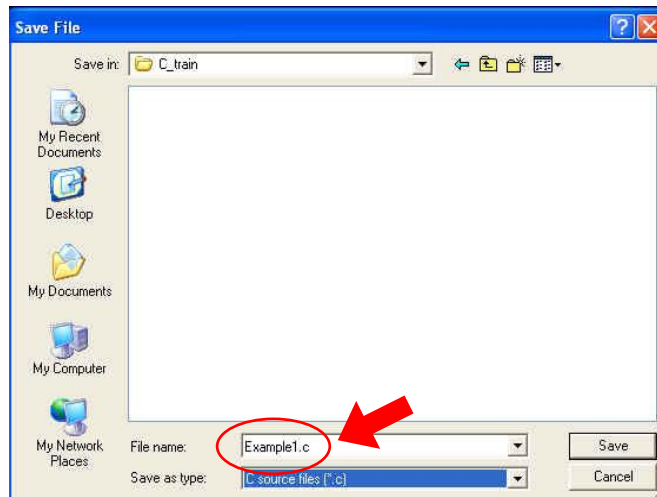


เขียนคำสั่ง ในช่องนี้

เมื่อเขียนคำสั่ง หรือ รหัสโปรแกรม จะได้นหน้าต่างโปรแกรมในลักษณะ ดังรูป



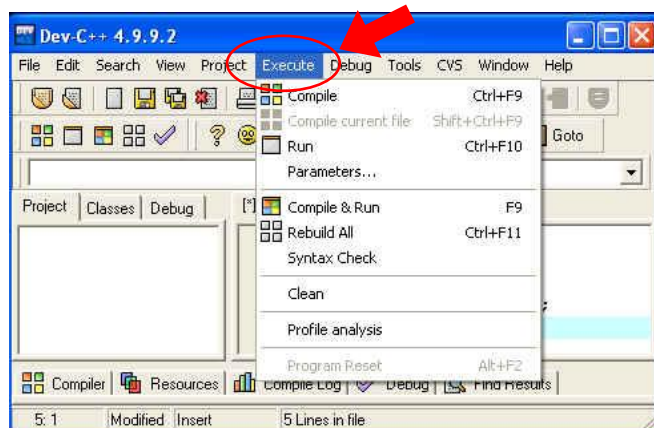
- คำสั่ง **Save** เมื่อเขียน โปรแกรมเสร็จหรือต้องการบันทึกไว้เพื่อป้องกันการเสียหาย การบันทึกครั้งแรกใช้คำสั่ง File Save As... หรือถ้าใช้คำสั่ง File Save กับไฟล์ที่ไม่มีการบันทึกมาก่อน โปรแกรมจะเปิดหน้าต่าง ของ คำสั่ง File Save As... ดังรูป



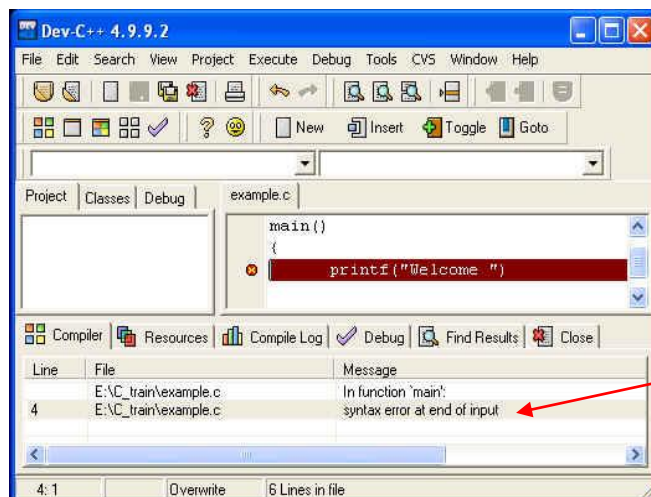
ในช่อง Save in: เลือกโฟลเดอร์ที่ต้องการเก็บไฟล์ ในช่อง Save as type: เลือก C source files(\*.c) ในช่อง File name: ตั้งชื่อตามต้องการ เพื่อความมั่นใจให้ใส่ส่วนขยายของไฟล์เป็น .c จะได้ไฟล์ source code ของโปรแกรมที่จะเป็นภาษา C ถ้าไม่กำหนดให้ถูกต้อง จะมีส่วนขยายเป็น .cpp ซึ่งเป็นไฟล์ ของ Source code ในภาษา C++ ซึ่งมีรายละเอียดบางประการต่างไป จึงต้องระวังในเรื่องนี้ คำสั่งอื่น ๆ ในเมนู File ก็เป็นลักษณะทำนองเดียวกันกับโปรแกรมอื่น ๆ จึงไม่ขออธิบายในที่นี้

## 1.2) เมนู Execute

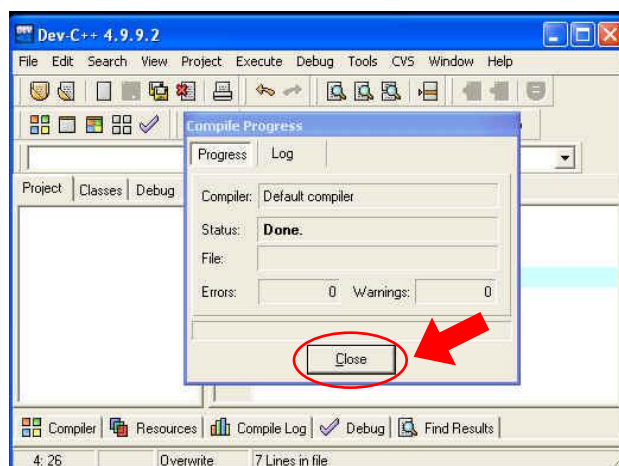
เมนูนี้ใช้สำหรับเพื่อเปลี่ยนคำสั่งให้เป็นภาษาเครื่องและทดลองสั่งให้โปรแกรมโดยไฟล์ที่จะใช้ในเมนูนี้ จะต้องเป็นไฟล์ที่บันทึกมาก่อนแล้ว เมื่อเปิดเมนู Execute จะได้หน้าต่าง ดังรูป



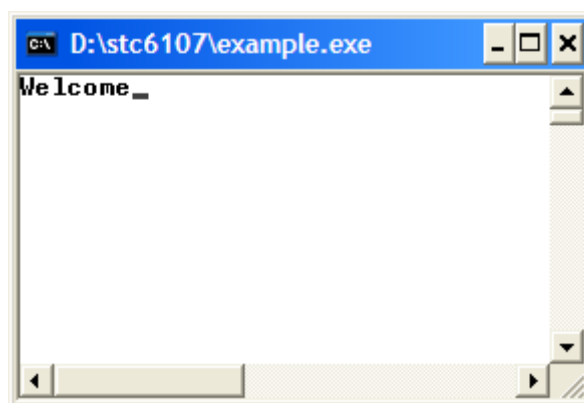
- คำสั่ง **Compile** เป็นคำสั่งที่ต้องใช้ประจำ คำสั่ง Compile เป็นการไปสั่งให้ Compiler เปลี่ยนคำสั่งที่เราสร้างไปเป็น ภาษาเครื่อง แล้วตามด้วยสั่งให้ Linker ชุดเดียวกันทำให้ภาษาเครื่องที่ได้รวมกับคำสั่งอื่น ๆ ที่จำเป็น เพื่อให้ได้โปรแกรมที่สามารถทำงานได้ ถ้ามีปัญหาต่าง ๆ ที่ทำให้ไม่ได้โปรแกรมที่สามารถทำงานได้ เช่น เขียนคำสั่งผิด จะแสดงความผิดพลาด ลักษณะดังรูป



ถ้าโปรแกรมที่เขียนสามารถ Compile ได้สำเร็จ แม้จะมีข้อผิดพลาดแต่เป็นข้อผิดพลาดที่ไม่มีปัญหาต่อการ Compile จะได้นหน้าต่าง ดังรูป ปิดหน้าต่าง Compile Progress เพื่อทำงานต่อไป

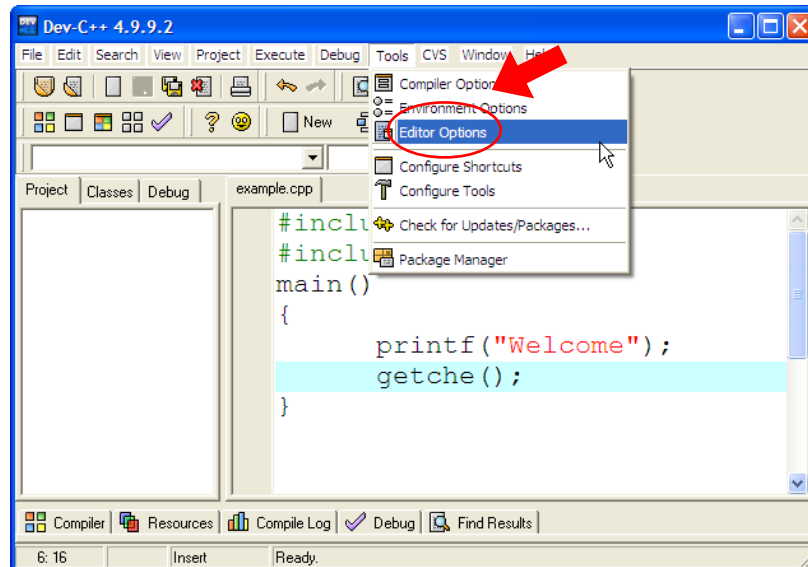


- คำสั่ง **Run** เป็นคำสั่งให้โปรแกรมที่ Compile แล้วทำงานเพื่อทดสอบดูผลการทำงานของโปรแกรม โปรแกรมที่จะใช้คำสั่งนี้ได้ต้องผ่านการ ใช้คำสั่ง Execute Compile มาแล้ว จะได้ผลลัพธ์ดังรูป

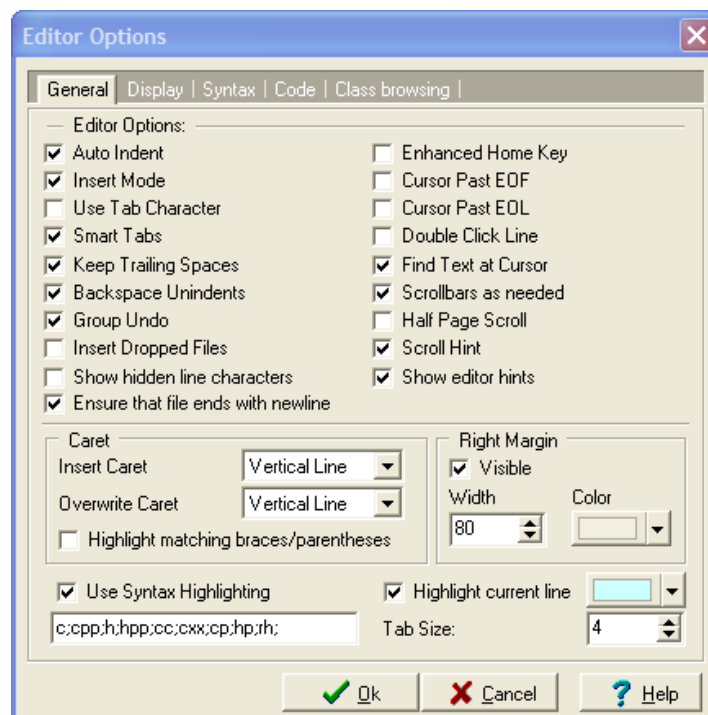


### 1.3) เมนู Tools

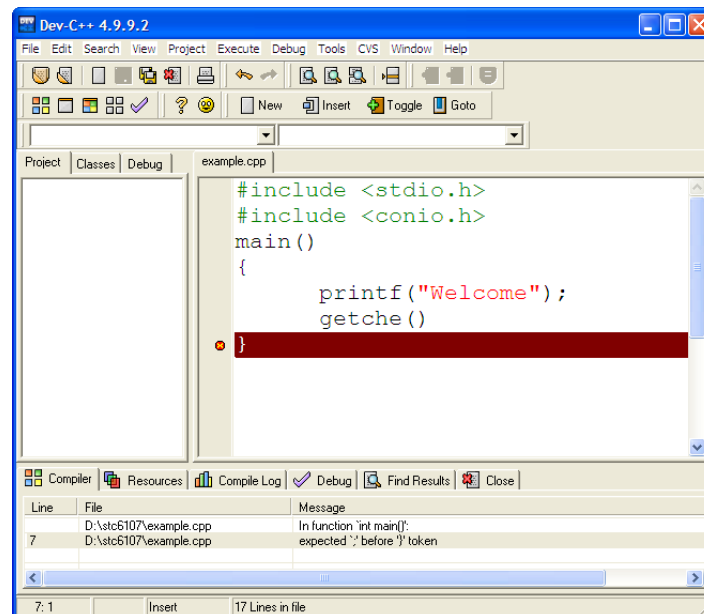
เมนูนี้มีคำสั่งหลายคำสั่ง ดังรูป



คำสั่งต่าง ๆ ในเมนูนี้ ขอแนะนำคำสั่งเดียวเพื่อความสะดวกในการแก้ไขข้อผิดพลาด เมื่อมีการเขียนคำสั่งผิด คือ คำสั่ง Editor Options เมื่อใช้คำสั่งนี้ ให้เลือก tab Display แล้วเลือกคำสั่ง Line Numbers ดังรูป



เมื่อ compile โปรแกรมแล้วมีข้อผิดพลาด โปรแกรมจะแจ้งว่าผิดพลาดที่บรรทัดใด ผู้เขียนโปรแกรมจะสามารถไปทำการได้ง่าย ๆ ไม่ต้องนับบรรทัด เพราะจะมีหมายเลขบรรทัดปรากฏอยู่ ดังรูป



```
#include <stdio.h>
#include <conio.h>
main()
{
    printf("Welcome");
    getch()
}
```

Line	File	Message
7	D:\stc6107\example.cpp	In function 'int main()': expected ';' before '}' token