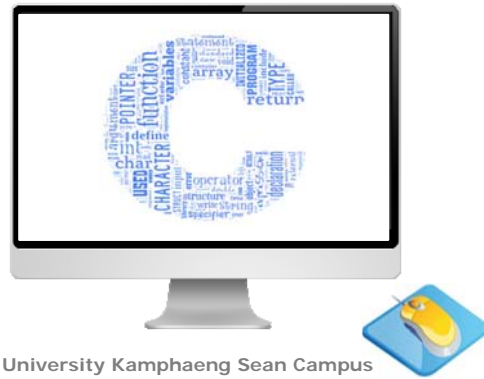


Getting started with C programming



Computer Engineering, Kasetsart University Kamphaeng Sean Campus

Outline

- ขั้นตอนในการเขียนโปรแกรม
- กระบวนการคอมไพล์ (Compile) โปรแกรม ในภาษาซี
- Integrated development environment (IDE)
- เริ่มต้นการใช้โปรแกรม Code::Blocks
- โครงสร้างโปรแกรมภาษา C
- รูปแบบคำสั่งในภาษาซี



1. ขั้นตอนในการเขียนโปรแกรม

- วิเคราะห์ปัญหา (Analysis)
 - ระบุปัญหาที่ต้องการจะแก้ไข
 - แบ่งปัญหานั้นออกเป็นปัญหาย่อยๆ
 - ค้นหาแนวคิดและวิธีการที่จะแก้ปัญหานั้นๆ
- พัฒนาโปรแกรม (Implementation)
 - เขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ โดยอาศัยแนวคิดการแก้ปัญหาที่คิดไว้ก่อนหน้านี้
- ตรวจสอบและแก้ไขจุดผิดพลาด (Testing & Debugging)
 - คอมไพล์และทดลองรันโปรแกรม เพื่อตรวจสอบการทำงาน
 - หากโปรแกรมทำงานไม่ถูกต้อง หาจุดผิดให้พบ
 - แก้ไขจุดผิดพลาดและทดลองใหม่ จนกว่าโปรแกรมจะทำงานได้ถูกต้อง

ผลลัพธ์ที่ได้ เรียกว่า ซอร์สไฟล์ (Source file) หรือ ซอร์สโค้ด (Source code) โดยมีนามสกุลเป็น .c



1. ขั้นตอนในการเขียนโปรแกรม

ตัวอย่าง : การคำนวณพื้นที่วงกลม

Input:
circle radius

Process:
calculate circle area πr^2

Output:
Circle area

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

```
int main()
{
    float pi = 3.1416;
    int radius = 5;
    float area;

    area = pi*radius*radius;
    printf("circle area is %f", area);

    return 0;
}
```

- Compile
- Error checking
 - Syntax error
 - Runtime error
 - Logic error

วิเคราะห์ปัญหา

พัฒนาโปรแกรม

ตรวจสอบ และ
แก้ไขจุดผิดพลาด



1. ขั้นตอนในการเขียนโปรแกรม

ข้อผิดพลาดของโปรแกรม

■ Syntax error/ Compile-time error

- คือ ข้อผิดพลาดจากการใช้ไวยากรณ์ภาษาที่ผิด หรือ อาจเกิดจากการสะกดคำผิด
- ตรวจพบได้ในขณะที่ทำการคอมไพล์ (compile) โปรแกรม

```
Printf("Hello world");  
printf("Hello world");
```



5

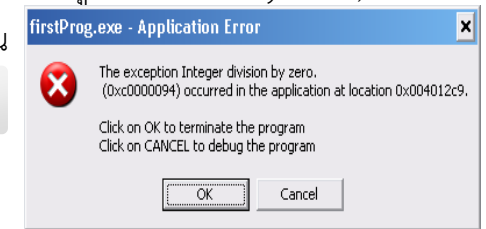
1. ขั้นตอนในการเขียนโปรแกรม

ข้อผิดพลาดของโปรแกรม

■ Run-time error

- คือ ข้อผิดพลาดที่เกิดจากสภาวะแวดล้อมในขณะที่ทำงานส่งผลทำให้การทำงานผิดพลาด
- ไม่สามารถตรวจพบได้ในขณะที่ทำการคอมไพล์โปรแกรม
- เช่น การคำนวณที่มีตัวหารมีค่าเป็นศูนย์ (Divided by zero),
ป้อนข้อมูลผิดประเภท เป็นต้น

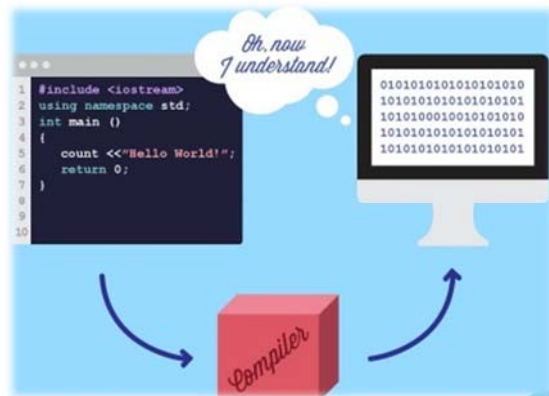
```
x = 10/(3-6/2);
```



6

2. กระบวนการคอมไพล์โปรแกรมในภาษาซี

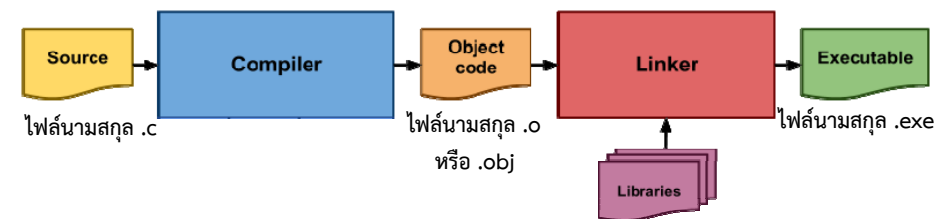
- ตัวแปลภาษาคอมไพเลอร์ (Compiler) จะอ่านโปรแกรมทั้งหมด (Source code) และแปลงไปเป็นภาษาเครื่อง (machine language)



7

2. กระบวนการคอมไพล์โปรแกรมในภาษาซี

- กระบวนการการแปลภาษา (Compiler Pipeline) อธิบายได้ดังนี้



- คอมไพเลอร์ของภาษาซี (C Compiler) ทำการแปลงซอร์สไฟล์ (ไฟล์นามสกุล .c) จากอักขระใดๆ ให้เป็นรหัสที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้ (ไฟล์นามสกุล .o)
- ตัวเชื่อม (Linker) จะทำการรวมฟังก์ชันมาตรฐานจากห้องสมุดของภาษาซี (C Library) เข้ากับไฟล์นามสกุล .o แล้วสร้างไฟล์ที่ทำงานได้ (ไฟล์นามสกุล .exe)



8

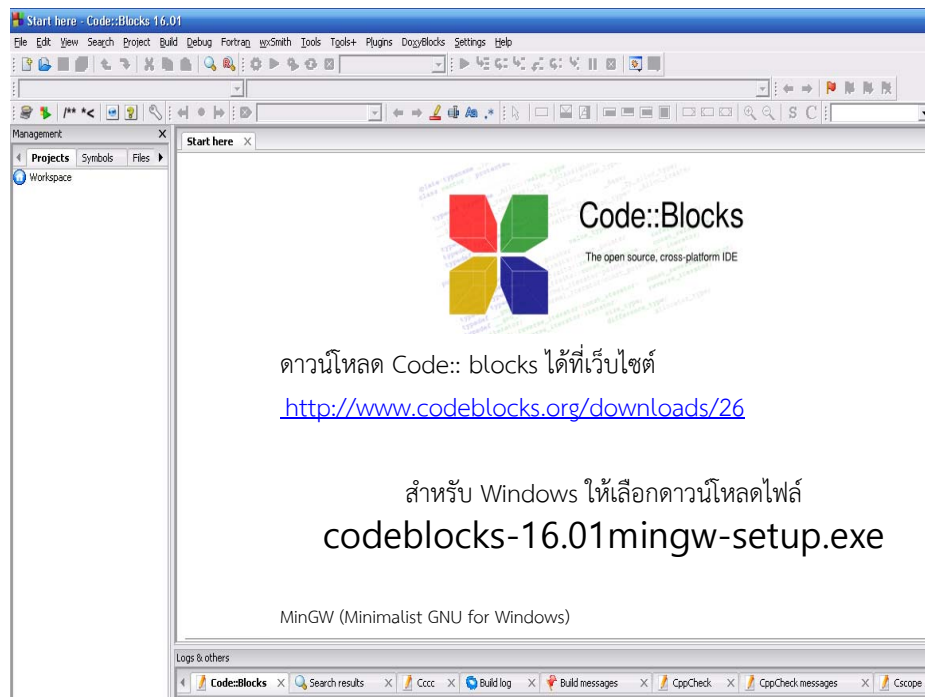
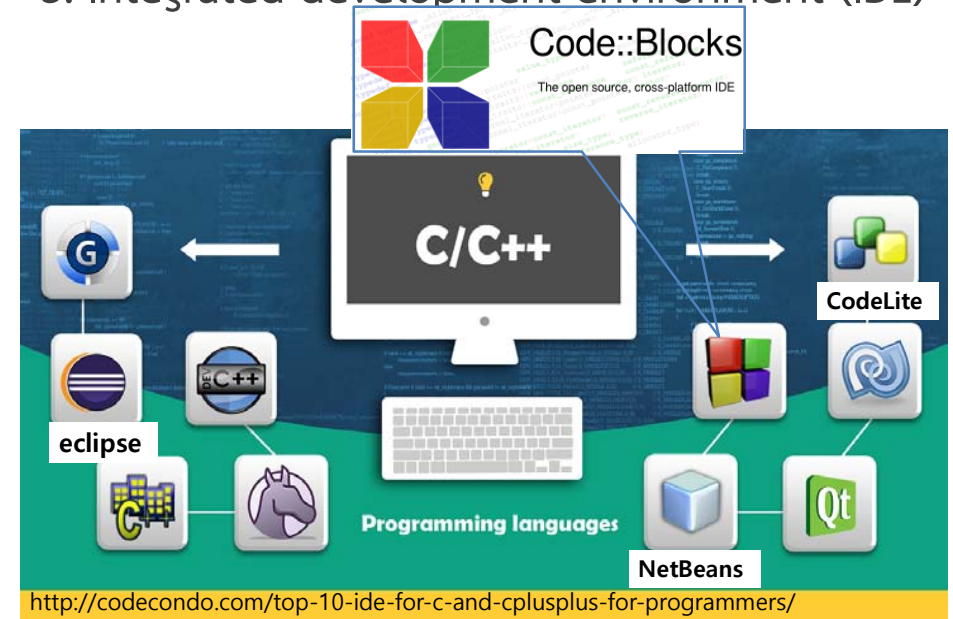
3. Integrated development environment (IDE)

- IDE คือ โปรแกรมประยุกต์ซึ่งอำนวยความสะดวกให้นักเขียนโปรแกรมคอมพิวเตอร์ในการพัฒนาซอฟต์แวร์ ประกอบด้วย
 - เครื่องมือสำหรับการเขียนโปรแกรม (Source code editor)
 - ตัวแปลภาษา (compiler/ interpreter)
 - และ เครื่องมืออำนวยความสะดวกในการพัฒนาโปรแกรม เพิ่มเติม เช่น โปรแกรมตรวจแก้จุดบกพร่อง (debugger)

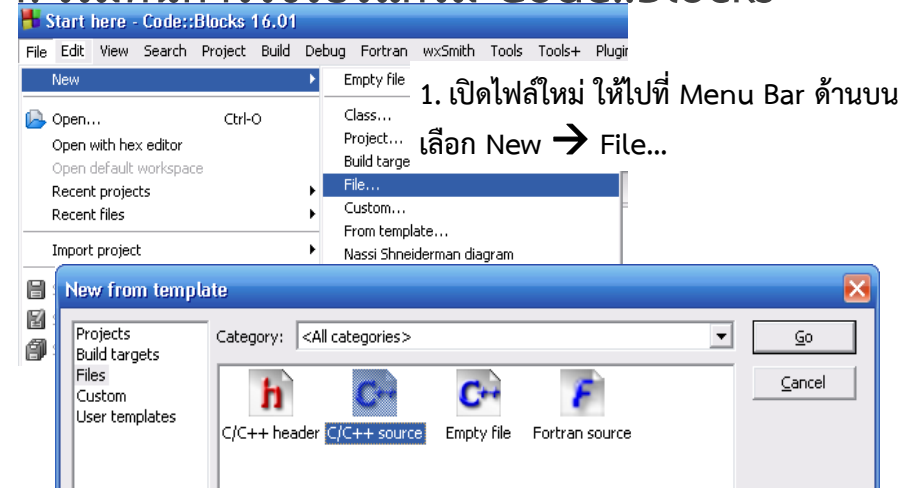


9

3. Integrated development environment (IDE)



4. เริ่มต้นการใช้โปรแกรม Code::Blocks

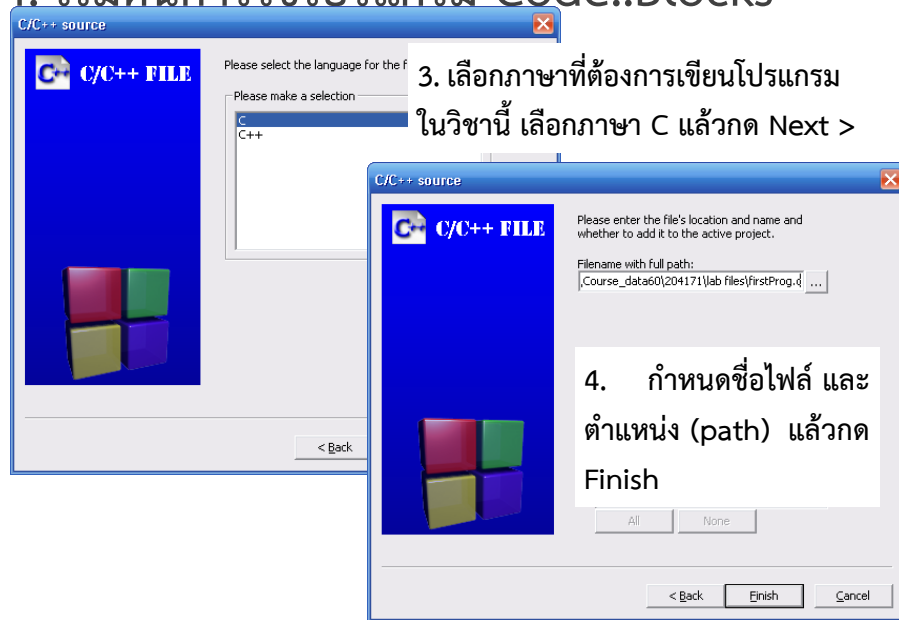


- เปิดไฟล์ใหม่ ให้ไปที่ Menu Bar ด้านบน เลือก New → File...
- เลือกประเภทของไฟล์ ให้เลือกไปที่ C/C++ source แล้วกด Go



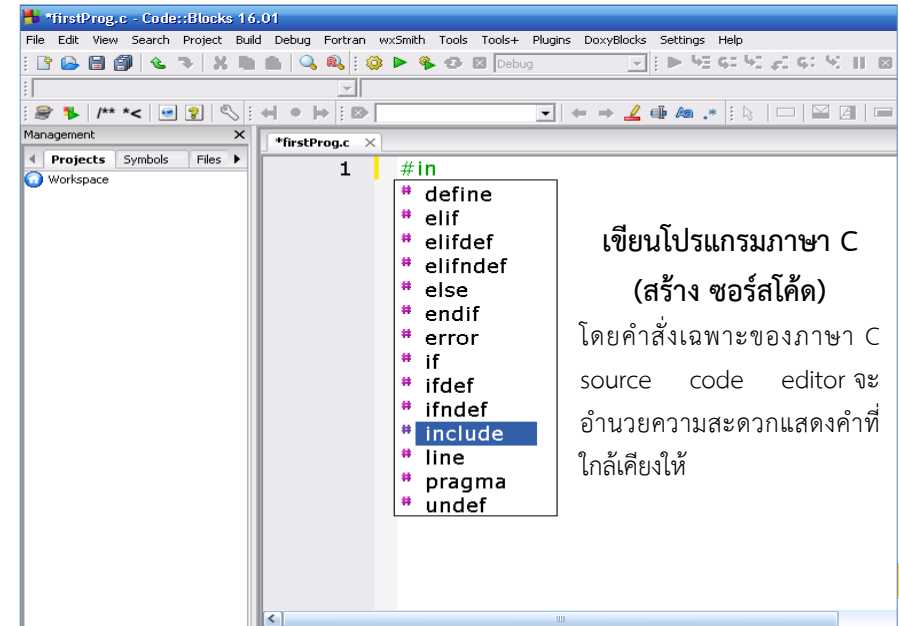
12

4. เริ่มต้นการใช้โปรแกรม Code::Blocks



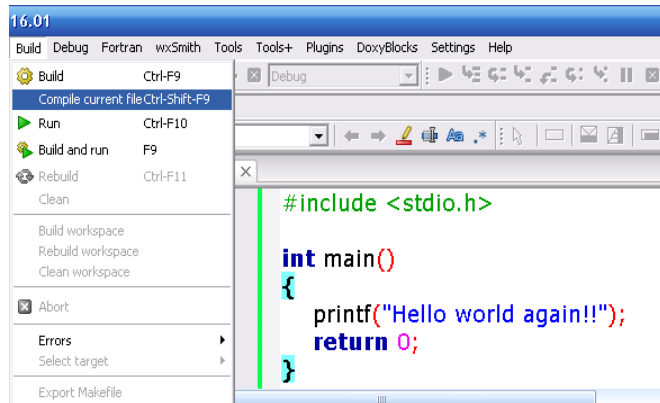
3. เลือกภาษาที่ต้องการเขียนโปรแกรม
ในวิชานี้ เลือกภาษา C แล้วกด Next >

4. เริ่มต้นการใช้โปรแกรม Code::Blocks



เขียนโปรแกรมภาษา C
(สร้าง ซอร์สโค้ด)
โดยคำสั่งเฉพาะของภาษา C
source code editor จะ
อำนวยความสะดวกแสดงคำที่
ใกล้เคียงให้

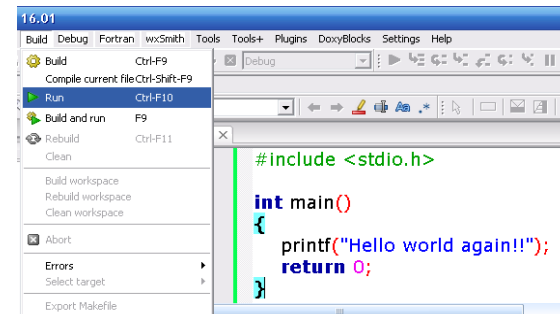
4. เริ่มต้นการใช้โปรแกรม Code::Blocks



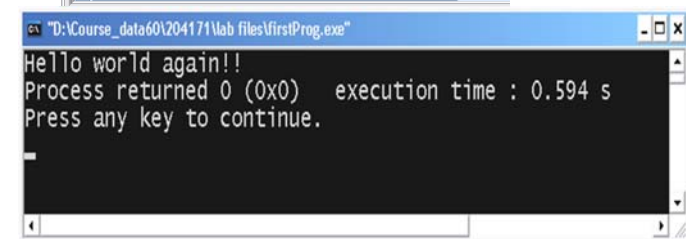
■ Compile โปรแกรม
ที่แถบ Menu bar ด้านบน
ให้เลือก Build →
Compile current file
หรือกดคีย์ลัด
Ctrl + Shift + F9

ตรวจสอบข้อผิดพลาด (Error) ของโปรแกรม
ถ้ามีข้อผิดพลาดจะไม่สามารถรันโปรแกรมได้

4. เริ่มต้นการใช้โปรแกรม Code::Blocks



■ Run โปรแกรม
หลังจากคอมไพล์สำเร็จ เราจะ
สั่งให้โปรแกรมทำงาน โดยกด
เลือก Build → Run
หรือกดคีย์ลัด Ctrl + F10



5. โครงสร้างโปรแกรมภาษา C

1 2
จะต้องมีเสมอ

โปรแกรมเมอร์เป็นผู้เขียนขึ้นมาตาม
ลักษณะของโปรแกรม

1 ส่วนเรียกใช้ library (ไฟล์นามสกุล .h)
ไฟล์เหล่านี้จะเป็นไฟล์มาตรฐานที่มี
ให้แล้วในภาษาซี (standard library)
หรือเป็นไฟล์ที่เขียนขึ้นมาใหม่ก็ได้

- stdio.h
- string.h

```
1 #include <stdio.h>
2 int main()
{
    float pi = 3.1416;
    int radius = 5;
    float area;

    area = pi*radius*radius;
    printf("circle area is %f", area);

3 return 0;
}
```



17

5. โครงสร้างโปรแกรมภาษา C

2 main function

- โปรแกรมภาษา C จะต้องประกอบด้วยฟังก์ชันชื่อ main() เสมอ และจะต้องมีเพียงหนึ่งฟังก์ชันต่อหนึ่งโปรแกรมเท่านั้น (ไม่ว่าจะประกอบด้วยไฟล์กี่ไฟล์ก็ตาม)
- โปรแกรมภาษา C จะเริ่มต้นทำงานในฟังก์ชัน main() เสมอ

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

```
int main() {
    float pi = 3.1416;
    printf("PI %f", pi);
    return 0;
}

int Main(){
}
```



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

```
int main() {
    float pi = 3.1416;
    printf("PI %f", pi);
    return 0;
}

int main(){
}
```



18

6. รูปแบบคำสั่งในภาษาซี

รูปแบบคำสั่งในภาษาซี มีกฎเกณฑ์ในการเขียนคำสั่ง ดังนี้

- คำสั่งทุกคำสั่งต้องเขียนด้วย **อักษรตัวเล็ก** เสมอ เช่น คำสั่ง printf , scanf
- ทุกคำสั่งจะใช้เครื่องหมาย semi-colon ‘;’ แสดงการจบของคำสั่ง
- การเขียนคำสั่ง จะเขียนได้แบบอิสระ (Free Format) คือ สามารถเขียน
หลายๆคำสั่งต่อกันได้ เช่น

```
printf("Hello");   printf("Goodbye");   a = 95;
```

หมายเหตุ แต่เพื่อความเป็นระเบียบและอ่านง่าย ควรจะเขียน 1 คำสั่งต่อ 1 บรรทัด



19

Your first program in C

- เขียนโปรแกรมแสดงข้อความ “Hello world!” ทางจอภาพ โดยทำตามขั้นตอนที่อธิบายในหัวข้อที่ 4 พร้อมทั้งศึกษาโครงสร้างของโปรแกรม ที่อธิบายในหัวข้อที่ 5
- เปิด folder ตำแหน่งที่บันทึกไฟล์ที่สร้างขึ้น ตรวจสอบว่ามีไฟล์ นามสกุล .c, .o และ .exe หรือไม่

Name	Size	Type
test		File Folder
firstProg	1 KB	C source file
firstProg	16 KB	Application
firstProg.o	1 KB	O File



20