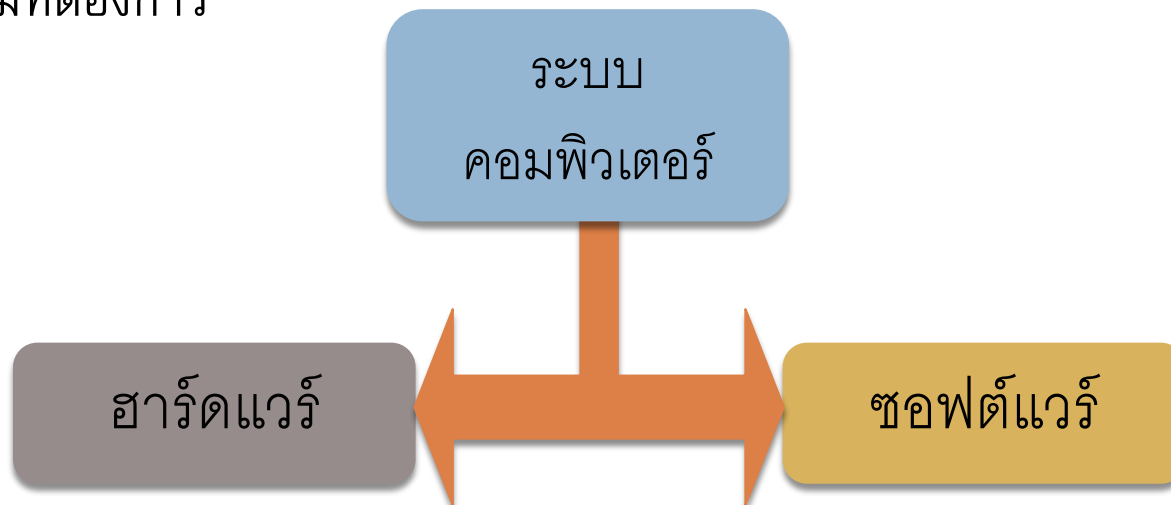


องค์ประกอบของคอมพิวเตอร์ และภาษาซี

ระบบคอมพิวเตอร์

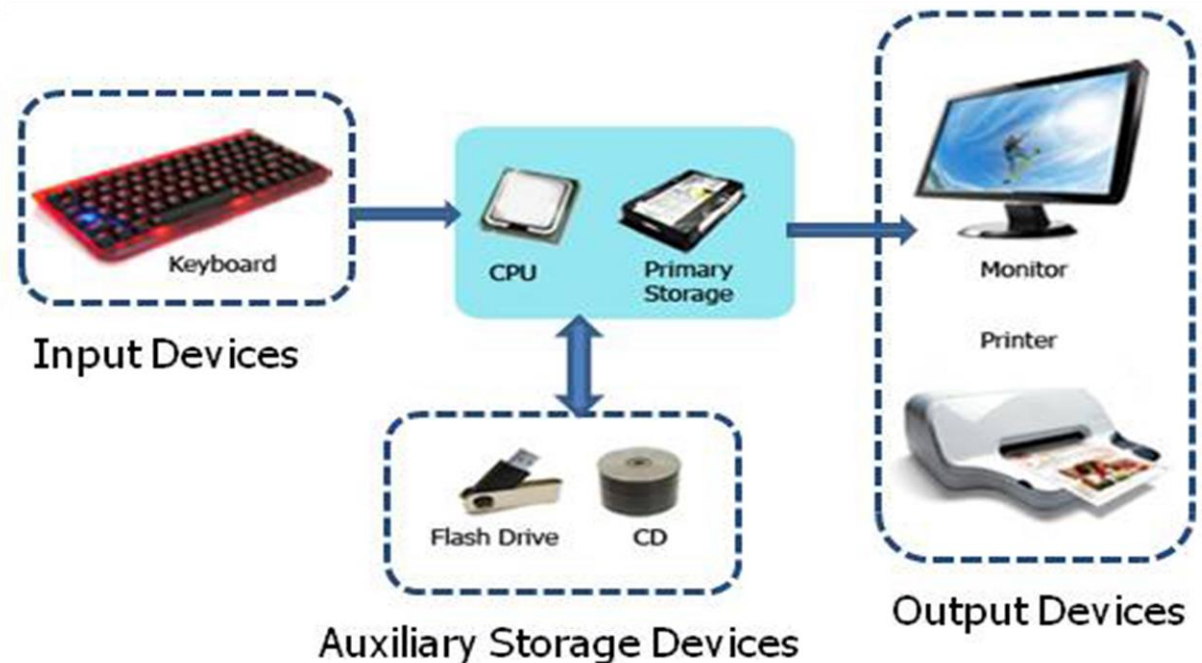
□ คอมพิวเตอร์

- อุปกรณ์ทางอิเล็กทรอนิกส์ที่ทำงานตามคำสั่งในการจัดการกับข้อมูล
- ระบบคอมพิวเตอร์ได้ถูกแบ่งออกเป็น 2 องค์ประกอบได้แก่
 - ฮาร์ดแวร์ (Hardware) : เป็นอุปกรณ์ทางกายภาพที่ประกอบขึ้น
 - ซอฟต์แวร์ (Software) : โปรแกรมที่ถูกสร้างขึ้นมาเพื่อสั่งให้ฮาร์ดแวร์ทำงานได้ตามที่ต้องการ

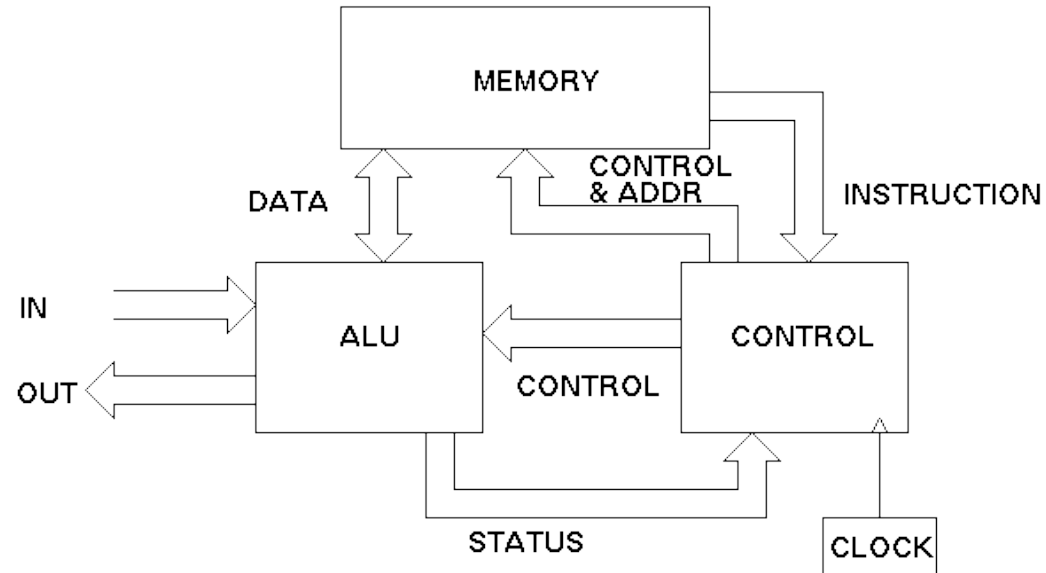


Computer Hardware

- องค์ประกอบฮาร์ดแวร์ของระบบคอมพิวเตอร์ประกอบไปด้วย 4 ส่วนหลัก
 - ▣ ตัวประมวลผลกลาง (Central processing unit หรือเรียกสั้นๆว่า ซีพียู (CPU))
 - ▣ หน่วยความจำ (Memory) ซึ่งรวม Primary storage และ Auxiliary storage
 - ▣ อุปกรณ์อินพุต
(Input Device)
 - ▣ อุปกรณ์เอาต์พุต
(Output Device)



Central Processing Unit (CPU)



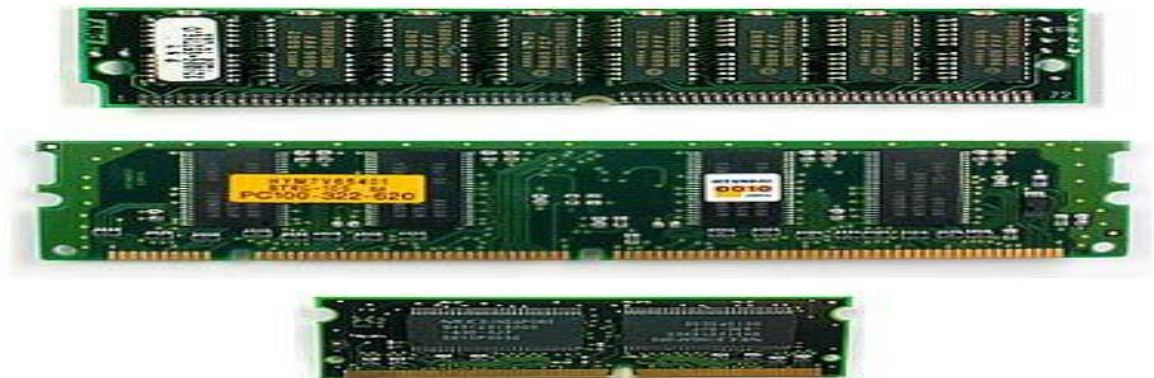
- ซีพียูเป็นอุปกรณ์ที่ทำงานตามชุดคำสั่งที่กำหนด
- มีความสามารถในการคำนวณทางคณิตศาสตร์ ตรรกศาสตร์ และสามารถติดต่อรวมทั้งควบคุมอุปกรณ์ต่างๆได้

Central Processing Unit (CPU) ต่อ

- จำนวนบิตของซีพียูจะเป็นตัวเลขที่บอกขนาดความยาวของรีจิสเตอร์ในการประมวลผล ปัจจุบันซีพียูที่วางจำหน่ายส่วนใหญ่มีขนาด 64 บิต (แต่รองรับการทำงานแบบ 32 บิตด้วย)
- ในอดีตความเร็วของซีพียูจะบอกเป็นความเร็วของสัญญาณนาฬิกาที่ป้อนให้กับตัวซีพียู เช่น Pentium4 2GHz, Celeron 1.2 GHz
- ปัจจุบันจึงมีการเปลี่ยนแปลงสถาปัตยกรรมภายในของซีพียูโดยเพิ่มหน่วยประมวลผลที่เรียกว่า ALU (Arithmetic Logic Unit) มากกว่าหนึ่งหน่วยลงไปในซีพียู ปัจจุบันที่เรารู้จักกันว่า คอร์ (Core)
 - ▣ สำหรับ Intel มีเทคโนโลยีที่ชื่อว่า HyperThreading เพิ่มเติม
 - Core i3 : 2 cores / 4 Threads
 - Core i5 : 4 cores / 4 Threads
 - Core i7 : 4 cores / 8 Threads

หน่วยความจำ (Memory)

- หน่วยความจำของคอมพิวเตอร์สามารถแบ่งออกเป็น 2 ชนิดคือ
 - ▣ หน่วยความจำหลัก
 - ▣ หน่วยความจำสำรอง
- หน่วยความจำหลัก (Primary Storage)
 - ▣ มีหน้าที่เก็บชุดคำสั่งที่ใช้สั่งงานซีพียู และ เก็บข้อมูลต่างๆที่จะให้ซีพียูประมวลผล รวมทั้งเก็บผลลัพธ์ที่ได้จากการประมวลผล
 - ▣ หน่วยความจำหลักในคอมพิวเตอร์เรียกว่า **RAM (Random Access Memory)**



หน่วยความจำ (Memory) ต่อ

□ หน่วยความจำสำรอง (Auxiliary memory)

- บางที่เรียกกันว่า **Secondary memory** หมายถึงหน่วยความจำหรือหน่วยเก็บที่ใช้เก็บข้อมูลนอกเครื่องคอมพิวเตอร์ (เพิ่มไปจากหน่วยความจำหลัก)
- ซึ่งสามารถใช้เก็บข้อมูลแบบถาวรได้ แต่การทำงานจะช้ากว่าหน่วยความจำหลัก แต่มีราคาที่ถูกลงกว่า หน่วยความจำสำรองเช่น **Hard disk, CD-ROM และ Flash drive**



อุปกรณ์อินพุต (Input Devices)

- เป็นอุปกรณ์ที่ใช้รับข้อมูลในรูปแบบต่างๆ จากภายนอกคอมพิวเตอร์ แล้วแปลงข้อมูลที่รับให้อยู่ในรูปแบบ **Binary** ซึ่งสามารถสื่อสารกับซีพียูได้
 - ▣ เครื่องแสกน(Scanner)
 - ▣ เครื่องอ่านบาร์โค้ด(Barcode Reader)
 - ▣ คีย์บอร์ด(Keyboard)
 - ▣ เมาส์(Mouse)
 - ▣ Trackball
 - ▣ Joystick ฯลฯ



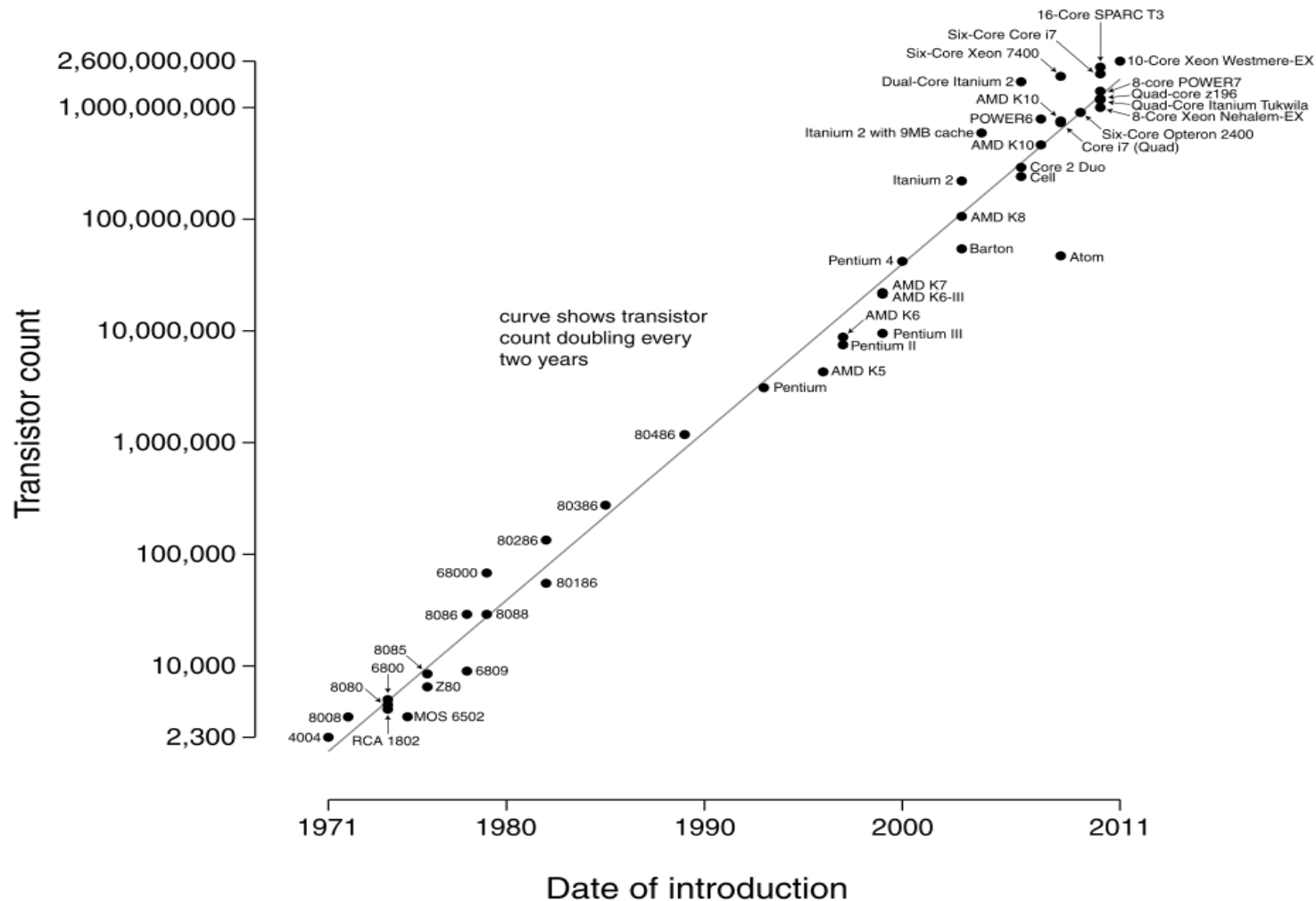
อุปกรณ์เอาต์พุต (Output Devices)

- เป็นอุปกรณ์ที่แปลงข้อมูลแบบ **Binary** จากซีพียู/คอมพิวเตอร์ไปเป็นข้อมูลในรูปแบบอื่นหรือให้อุปกรณ์นั้นทำงานอย่างใดอย่างหนึ่งตามลักษณะข้อมูล
 - เครื่องพิมพ์ (Printer)
 - ลำโพง (Speaker)
 - จอภาพ (Monitors) เป็นต้น

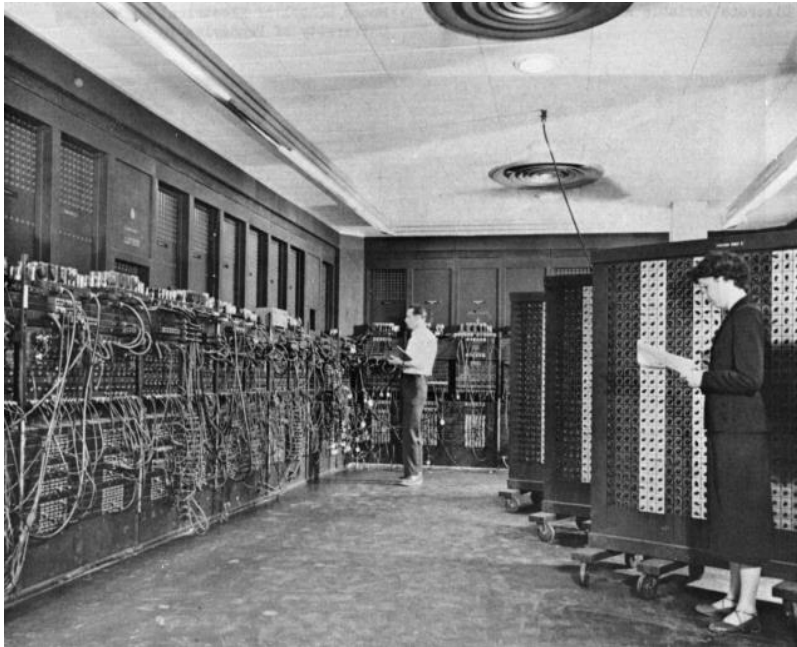


วิวัฒนาการของคอมพิวเตอร์

Microprocessor Transistor Counts 1971-2011 & Moore's Law



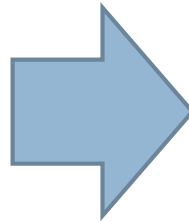
วิวัฒนาการของคอมพิวเตอร์



ENIAC

น้ำหนัก : 27 ตัน

ขนาดประมาณ : 63 m²



Netbook



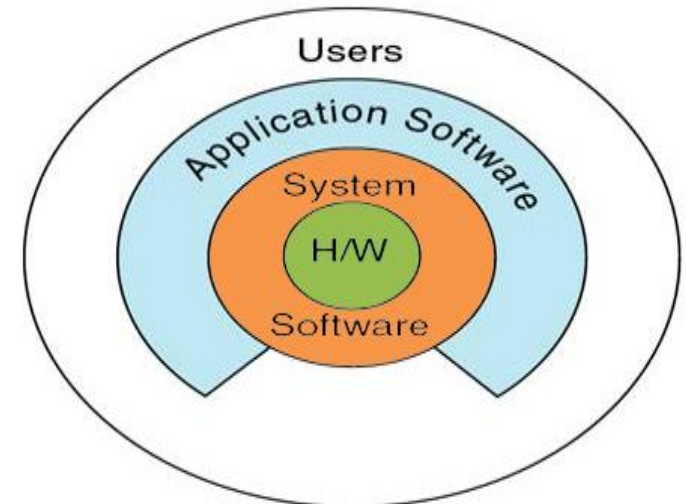
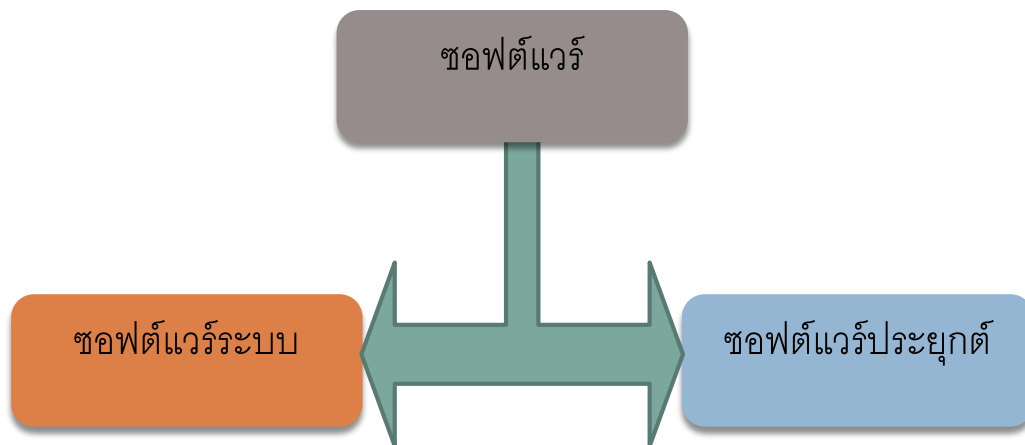
Tablet



Smart phone

Software

- ซอฟต์แวร์คอมพิวเตอร์ถูกแบ่งออกเป็น 2 ประเภทใหญ่
 - ▣ ซอฟต์แวร์ระบบ (System software) และ
 - ▣ ซอฟต์แวร์ประยุกต์ (Application software)



การพัฒนาซอฟต์แวร์ (Software Development)

- การพัฒนาซอฟต์แวร์หมายถึงขบวนการ การเขียนรหัส/โปรแกรมด้วยภาษาคอมพิวเตอร์ซึ่งการพัฒนาซอฟต์แวร์แบ่งออกเป็นขั้นตอนต่างๆ ดังนี้
 - ▣ การกำหนดความต้องการ ปัญหาที่ต้องการใช้คอมพิวเตอร์ช่วยในแก้ไข
 - ▣ การวิเคราะห์ตัวปัญหา ได้แก่การกำหนดข้อมูลขาเข้าและรูปแบบผลลัพธ์ที่ต้องการได้
 - ▣ การออกแบบ ได้แก่การออกแบบวางแผนถึงวิธีการแก้ไขปัญหาค่าที่จะเกิดขึ้น
 - ▣ การเขียนโปรแกรม ได้แก่การเขียนโปรแกรมด้วยภาษาทางคอมพิวเตอร์เพื่อแก้ไขปัญหา
 - ▣ การทดสอบ ได้แก่การทดสอบเพื่อตรวจสอบผลการแก้ไขปัญหา
 - ▣ การปรับปรุงแก้ไขโปรแกรม ได้แก่ในกรณีที่โปรแกรมยังไม่สามารถทำงานได้อย่างสมบูรณ์ต้องนำไปแก้ไขปรับปรุงใหม่เพื่อให้ได้ผลลัพธ์ที่ถูกต้องตามที่ต้องการ

คำศัพท์เทคนิคที่ใช้ในการพัฒนาซอฟต์แวร์

- **Edit** คือการเขียนและแก้ไขโปรแกรมด้วยภาษาคอมพิวเตอร์
- **Build หรือ Compile** คือการแปลโปรแกรมที่เขียนด้วยภาษาคอมพิวเตอร์ให้เป็นภาษาเครื่อง
- **Execute หรือ Run** คือการสั่งงานภาษาเครื่องที่ได้ให้ทำงาน
- **Debug** คือการหาจุดผิดพลาดของโปรแกรม
- **Integrated Development Environment (IDE) software** เป็นเครื่องมือในการพัฒนาซอฟต์แวร์แบบเบ็ดเสร็จสำหรับวงจรการ
“edit-compile-execute-debug”

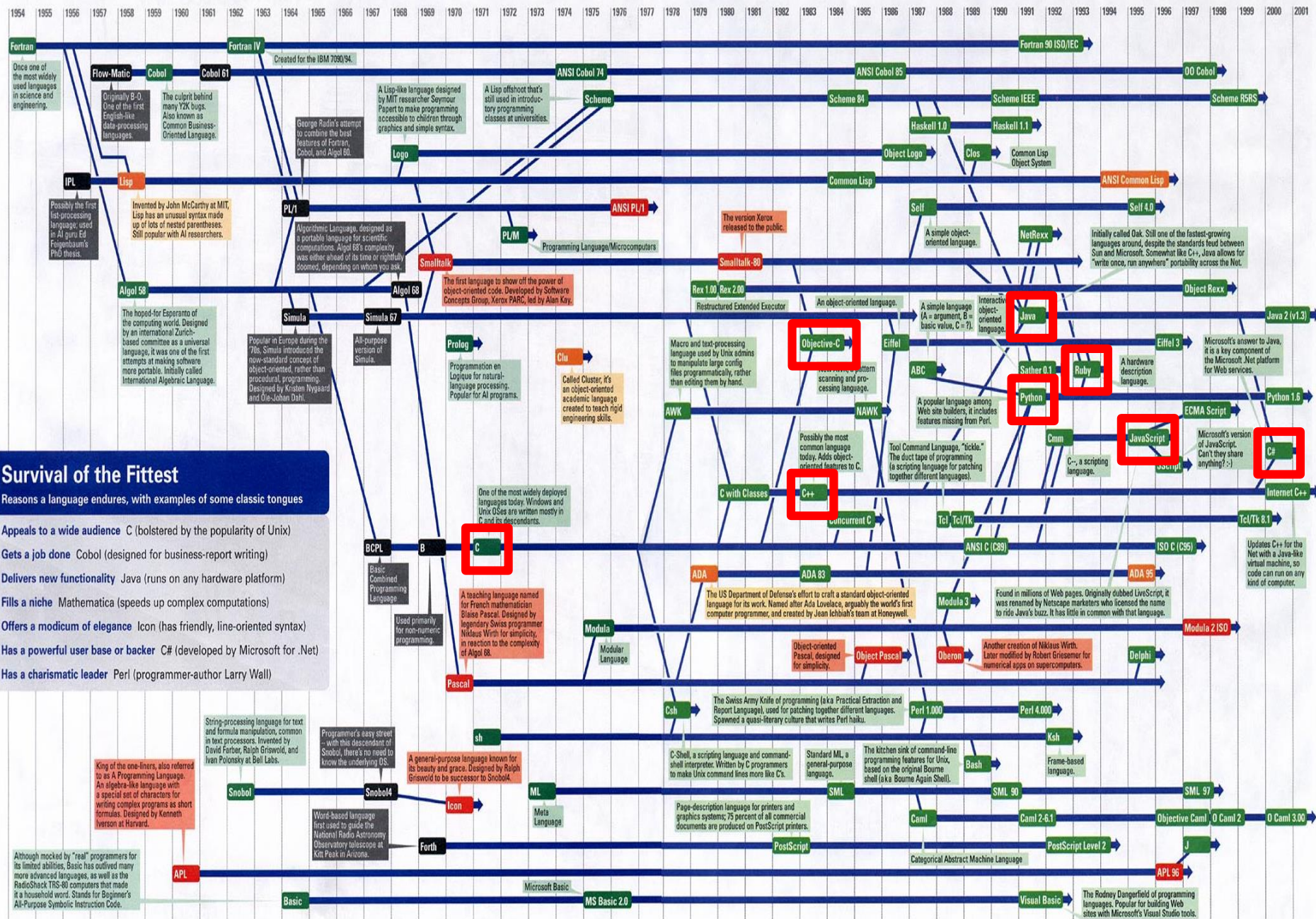
ภาษาคอมพิวเตอร์ (Computer Languages)

- ภาษาคอมพิวเตอร์นั้นสามารถแบ่งออกเป็นประเภทใหญ่ๆ 3 ประเภทคือ
 - ▣ ภาษาโปรแกรมระดับต่ำ (Low-level Programming Language)
 - ▣ ภาษาโปรแกรมระดับสูง (High-level Programming Language)
 - ▣ ภาษาโปรแกรมระดับสูงมาก (Very High-level Programming Language)

แนะนำภาษาซี

- ภาษาซีได้ถูกพัฒนาขึ้นโดยนายเดนนิส ริทชี ณ ศูนย์วิจัยเบล ในสหรัฐอเมริกา เมื่อปี ค.ศ. 1972 จนกระทั่งปี ค.ศ. 1978 นายไบรอัน เคอร์นิแกน และนายเดนนิส ริทชี ได้จัดทำข้อกำหนดมาตรฐานของภาษาซี ซึ่งนิยมเรียกกันว่า **K&R** หลังจากนั้นปี ค.ศ. 1980 ภาษาซี ก็ได้รับความนิยมมากขึ้น มีการพัฒนา **compiler** ภาษาซีออกมามากมาย
- ภาษาซีเป็นภาษาที่มีความใกล้เคียงกับภาษาระดับต่ำ จึงทำให้มีความเร็วในการทำงานสูง และในขณะเดียวกันภาษาซีก็ยังคงมีความเป็นภาษาระดับสูง ทำให้นักพัฒนาสามารถที่จะพัฒนาโปรแกรมได้ โดยเน้นไปที่การแก้ปัญหาที่ต้องการได้อย่างอิสระโดยไม่ต้องคำนึงถึงฮาร์ดแวร์
- ภาษาซีเป็นภาษาโปรแกรมคอมพิวเตอร์ที่ได้รับความนิยมและมีการพัฒนาอย่างต่อเนื่อง
 - ▣ ระบบปฏิบัติการยอดนิยมส่วนใหญ่ไม่ว่าจะเป็น **Windows XP, Linux, UNIX, MAC OS X** พัฒนาขึ้นด้วยภาษาซี
 - ▣ ภาษาซียังถูกใช้ในการพัฒนาภาษาระดับสูงอื่นๆ เช่น **PHP, Perl, Ruby** และ **Python**
 - ▣ ภาษาซีเป็นภาษาต้นตระกูลของภาษายอดนิยมอื่นๆ เช่น **C++, Java, PHP** ซึ่งจะมีโครงสร้างและไวยากรณ์ของภาษาค่อยคลึงกับภาษาซี

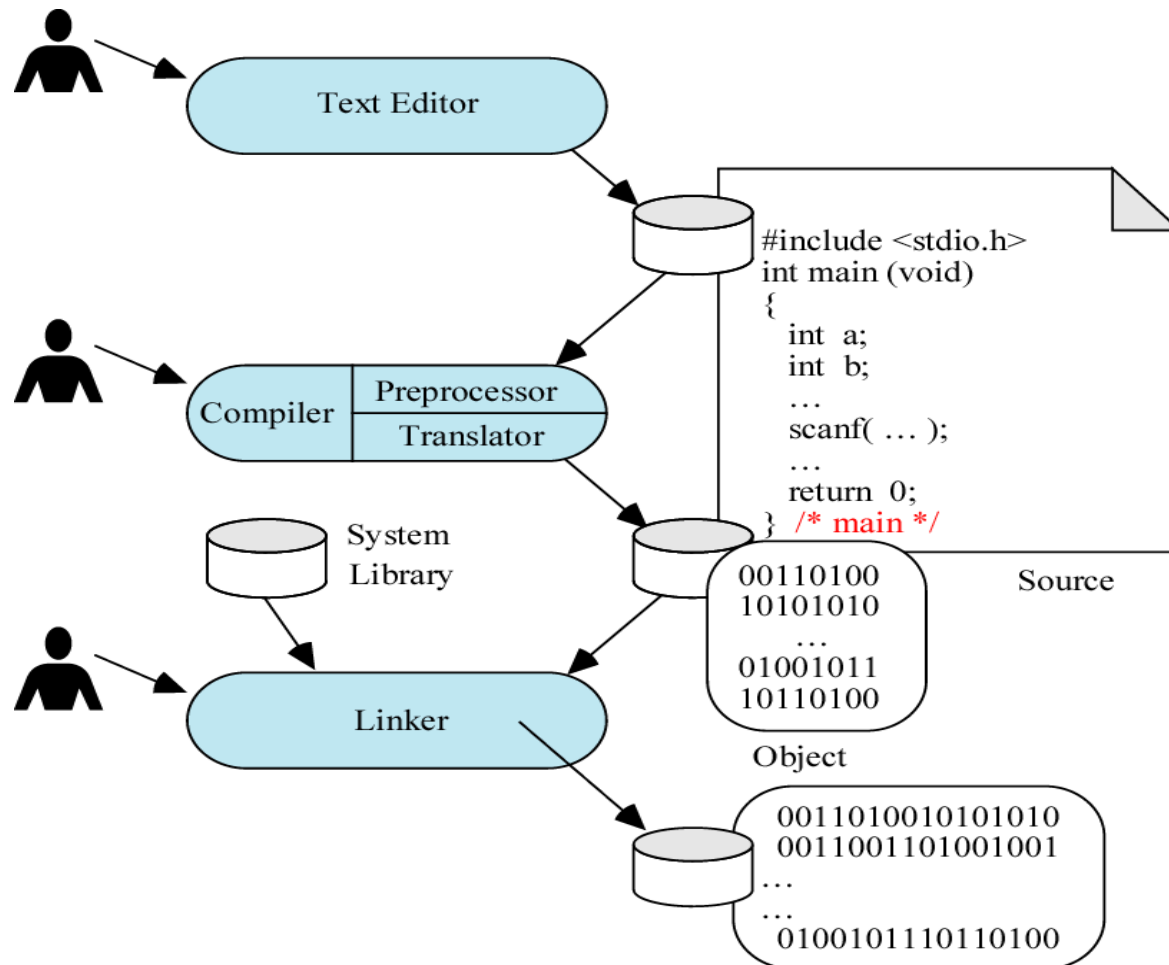
สายพันธุ์ภาษาโปรแกรม



ภาษาโปรแกรมยอดนิยม July 2015 (tiobe.com)

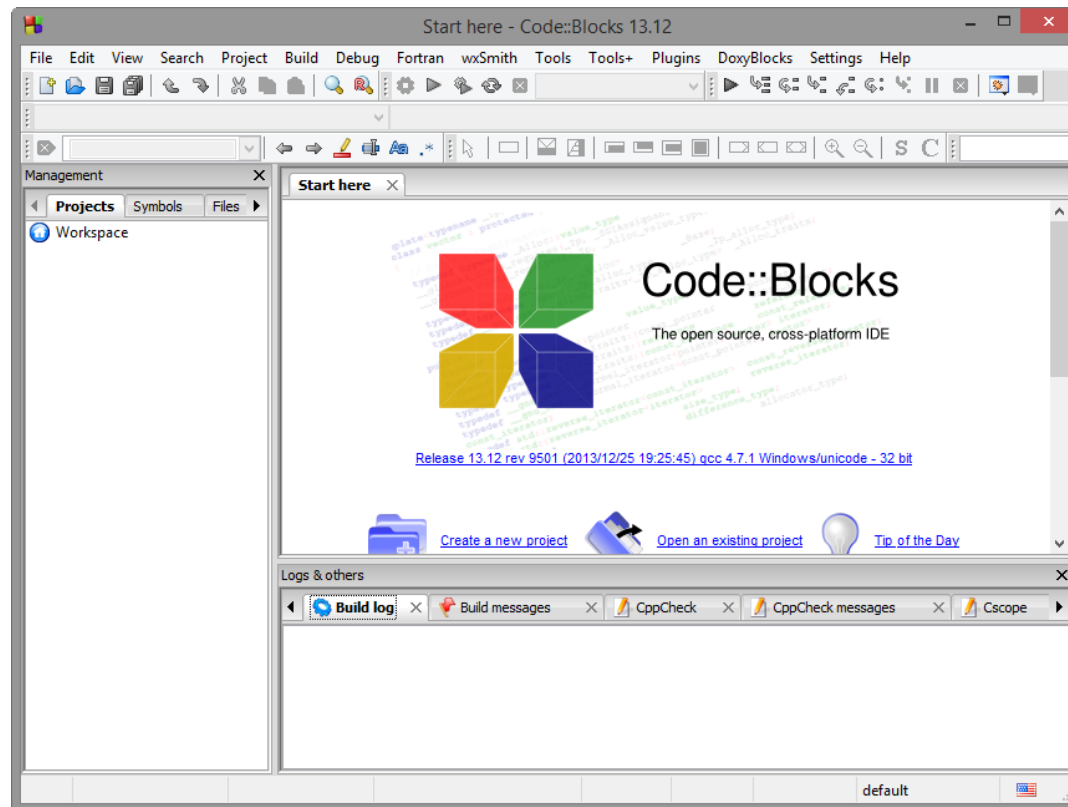
Jul 2015	Jul 2014	Change	Programming Language
1	2	▲	Java
2	1	▼	C
3	4	▲	C++
4	6	▲	C#
5	8	▲	Python
6	3	▼	Objective-C
7	7		PHP
8	12	▲▲	Visual Basic .NET
9	9		JavaScript
10	-	▲▲	Visual Basic

ขั้นตอนการพัฒนาโปรแกรมภาษาซี



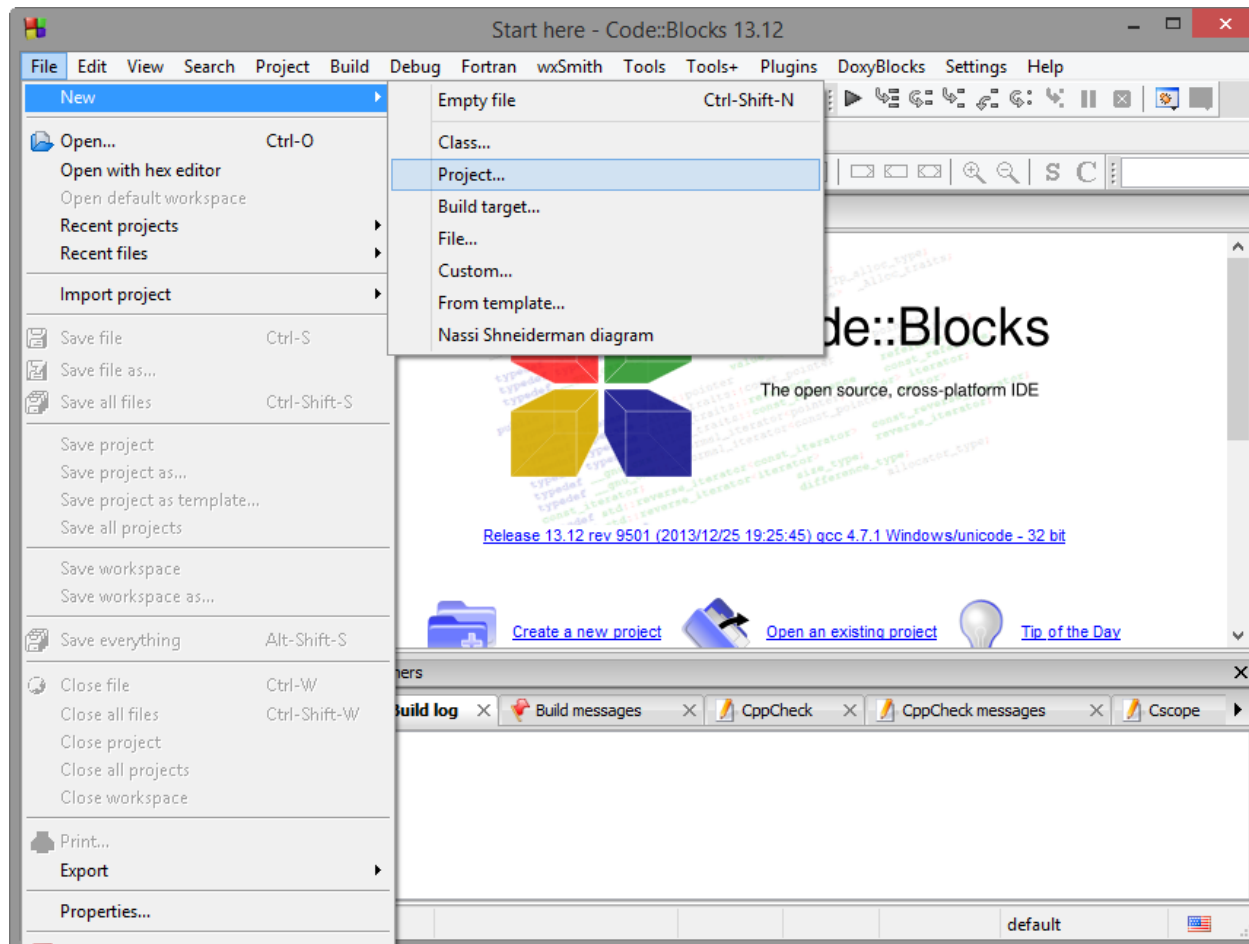
CodeBlocks IDE

- Free IDE for developing software using C/C++ programming language
- Site: <http://www.codeblocks.org/>



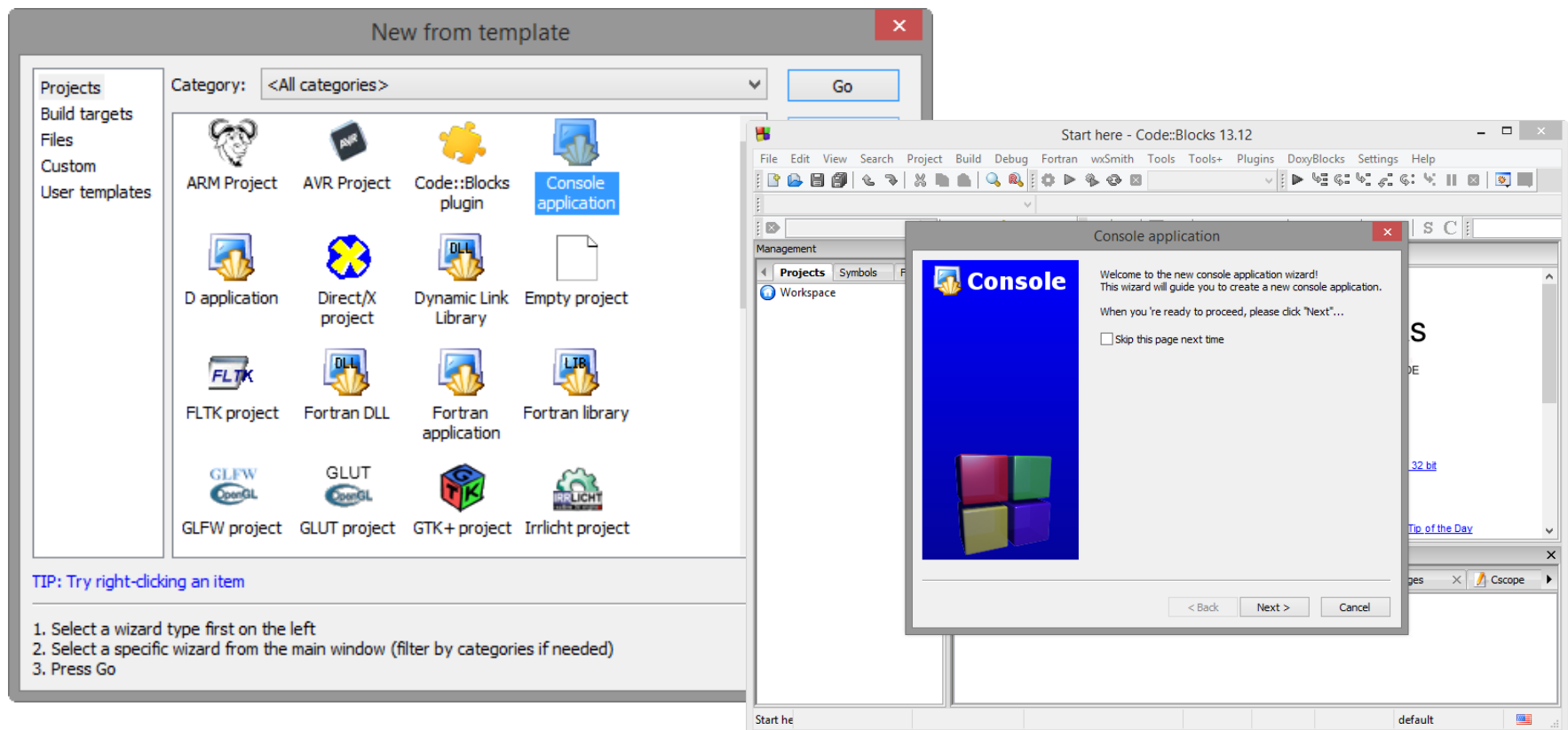
New Project

□ File → New → Project



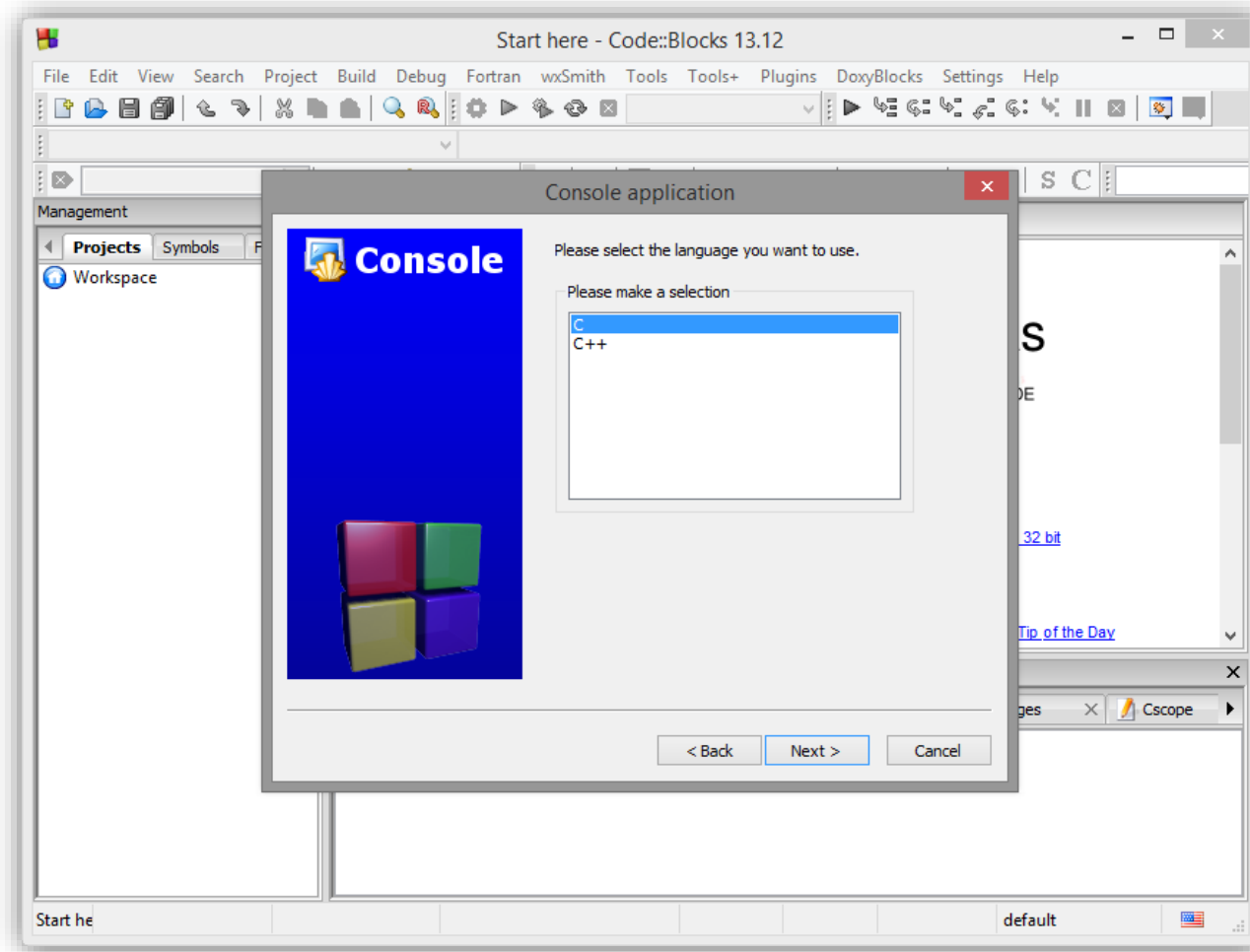
Choose Project Type

- เลือก **Console Application**
- คลิก **Go**, แล้วกด **Next**



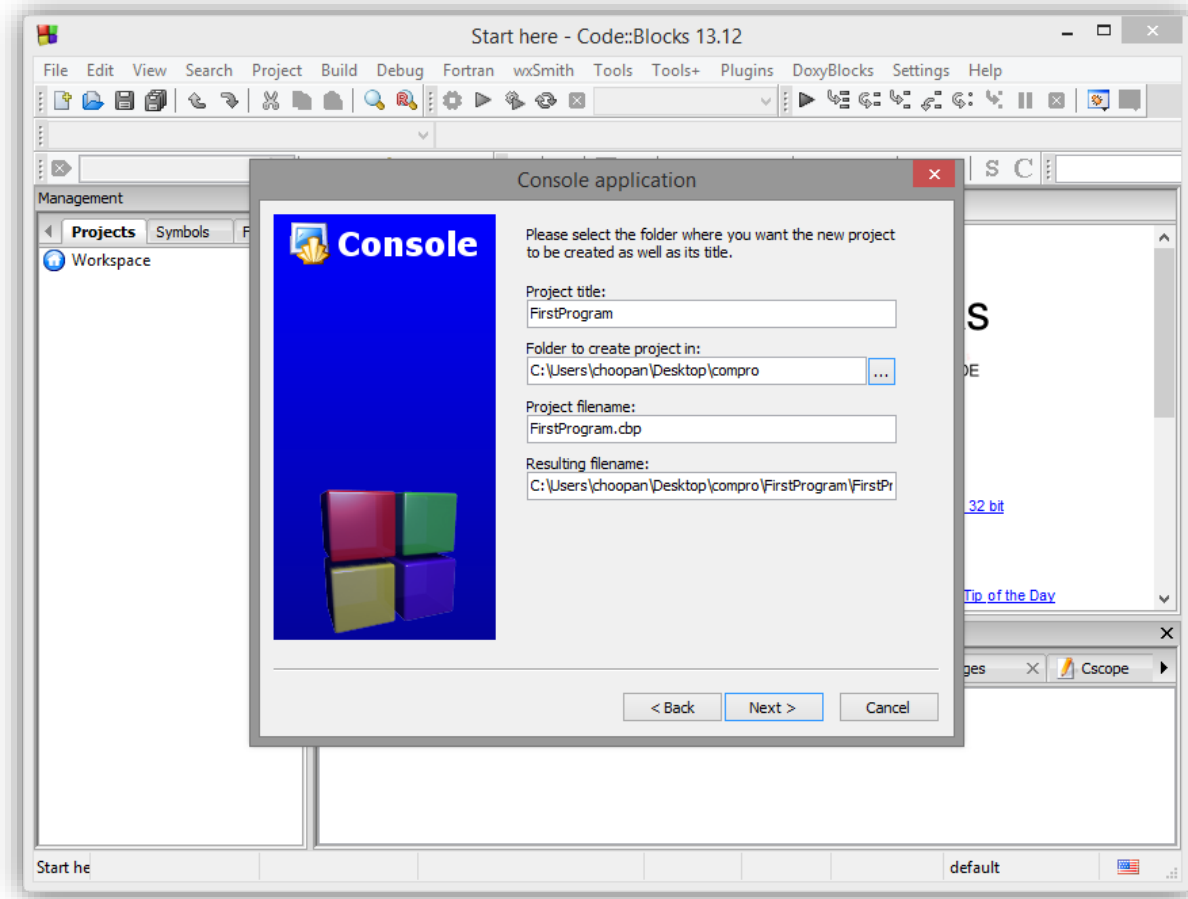
Select the language

□ เลือก C แล้วกด Next



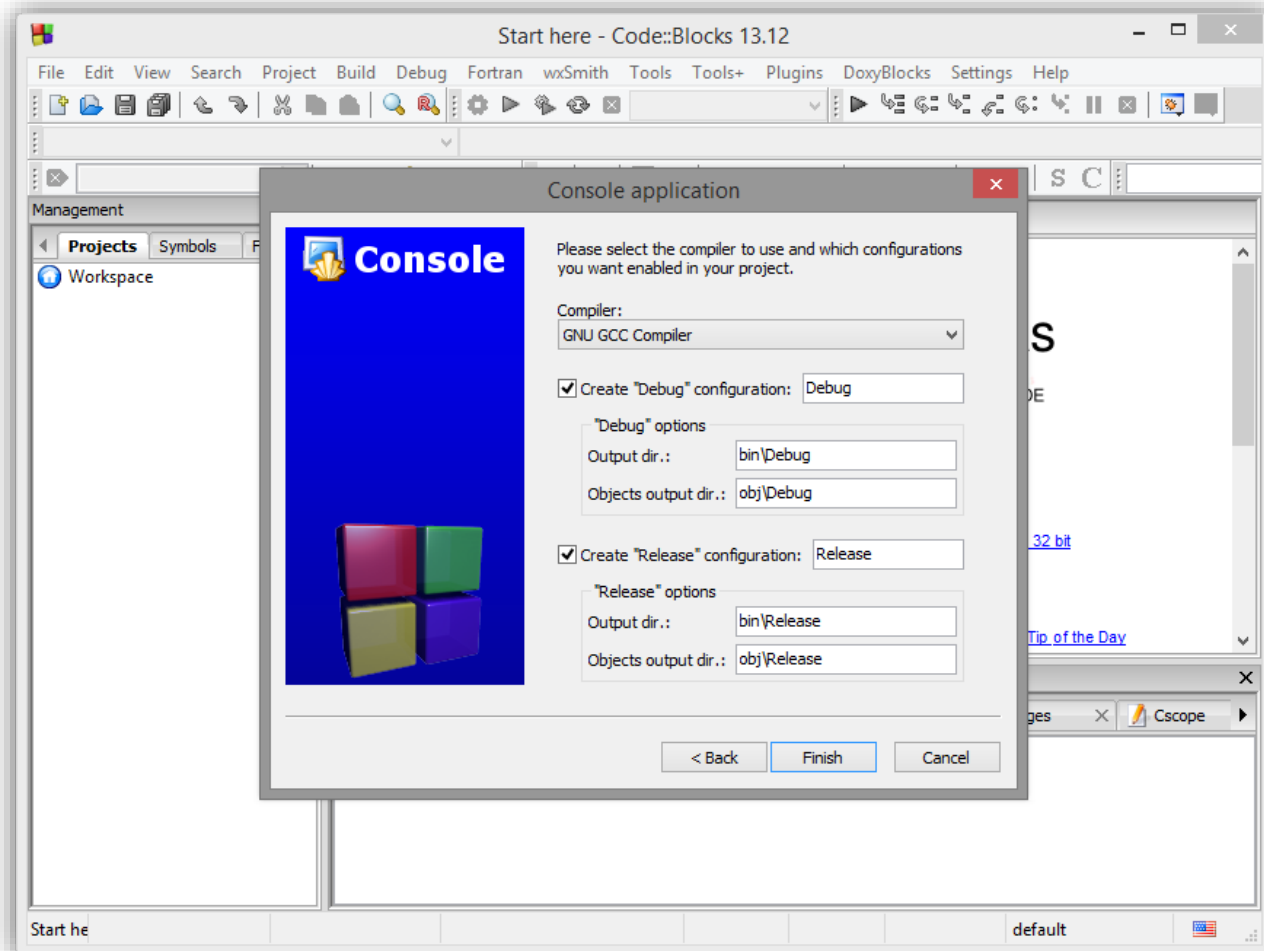
Project Location

- พิมพ์ชื่อโปรเจ็ค ที่ “project title”
- เลือกตำแหน่งใน Harddisk ที่จะเก็บไฟล์ต่างๆ

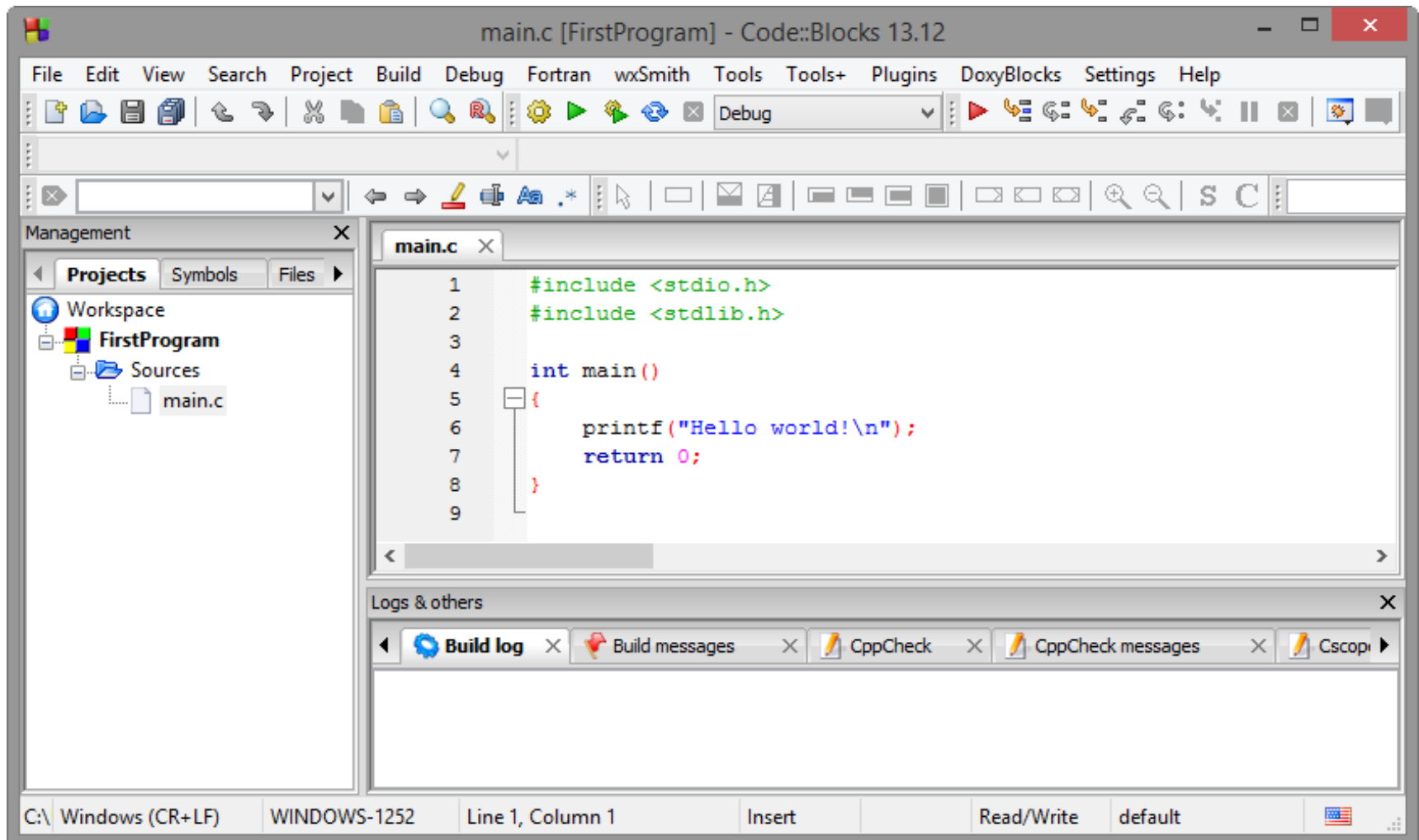


Choose Compiler

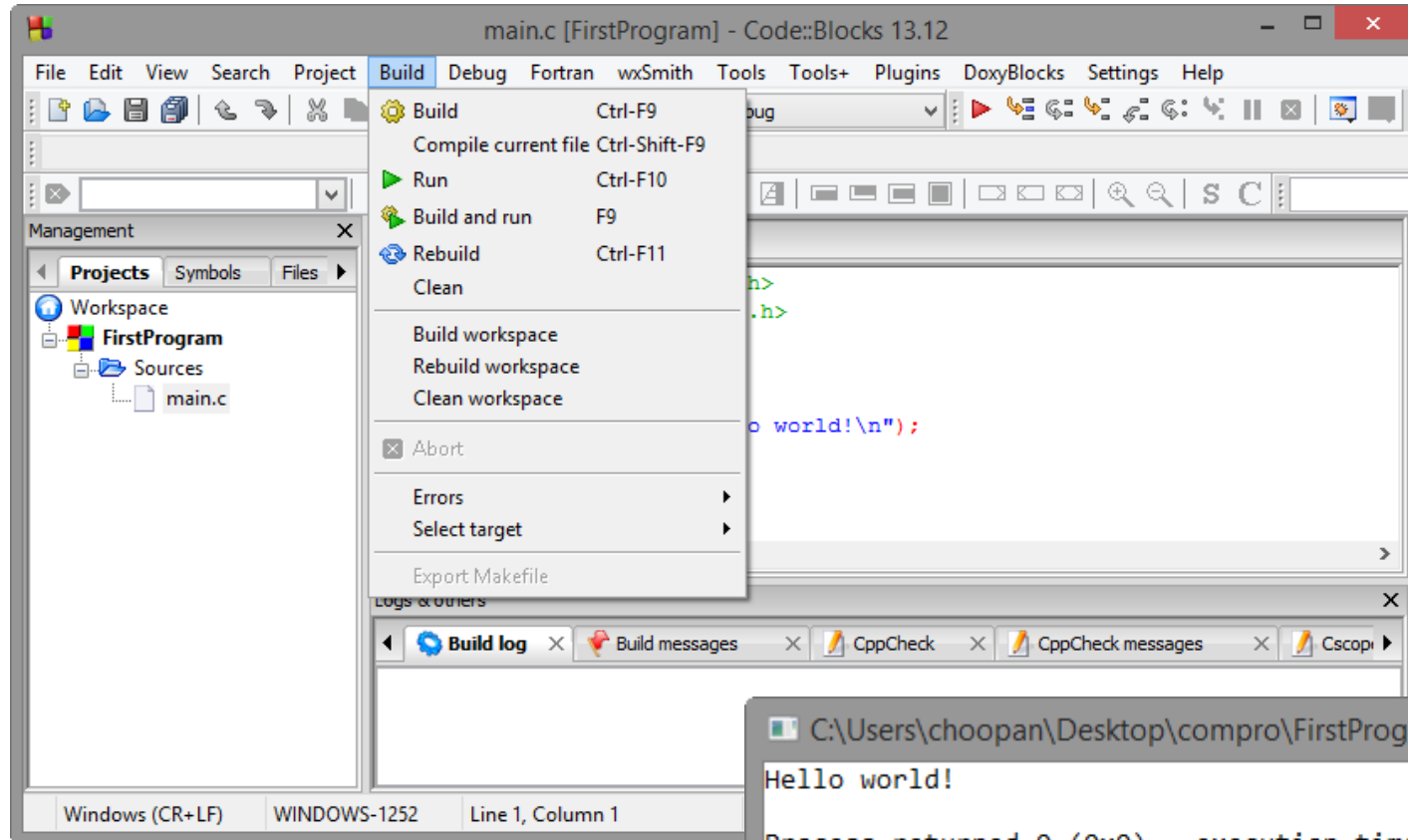
□ กดปุ่ม **Finish**



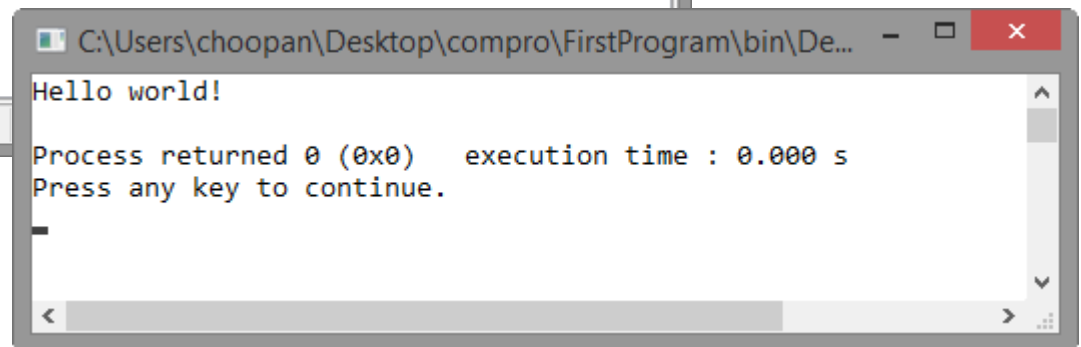
Default Source Code



Run Menu



- Build
- Run
- **Build and Run (F9)**



โครงสร้างของภาษาซี

Preprocessor Directives

Global Declarations

```
int main (void)
```

```
{
```

Local Definitions

Statements

```
}
```

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

Global Declarations

```
int main (void)
```

```
{
```

Local Definitions

```
printf("Hello World!\n");
```

```
return 0;
```

```
} /* main */
```

Preprocessor command to include standard input/output information for your program.



ตัวอย่าง Source Code ภาษา C

Preprocessor
Directives

Global Declarations

int main (void)

{

Local Definitions

Statements

}

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

```
int main()
```

```
{
```

Your code

```
return 0;
```

```
}
```

Hello World

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

int main()
{
    printf("Hello World");

    return 0;
}
```

printf(“ข้อความ”);

เป็นฟังก์ชันมาตรฐานของภาษา C ใช้สำหรับแสดง *ข้อความ* ออกบนหน้าจอ

ทดลองอะไรเล่น ๆ

- ให้ลองเอา **semicolon(;**) ออกจากบรรทัด

printf("Hello World");

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

```
int main()
```

```
{
```

```
    printf("Hello World")
```

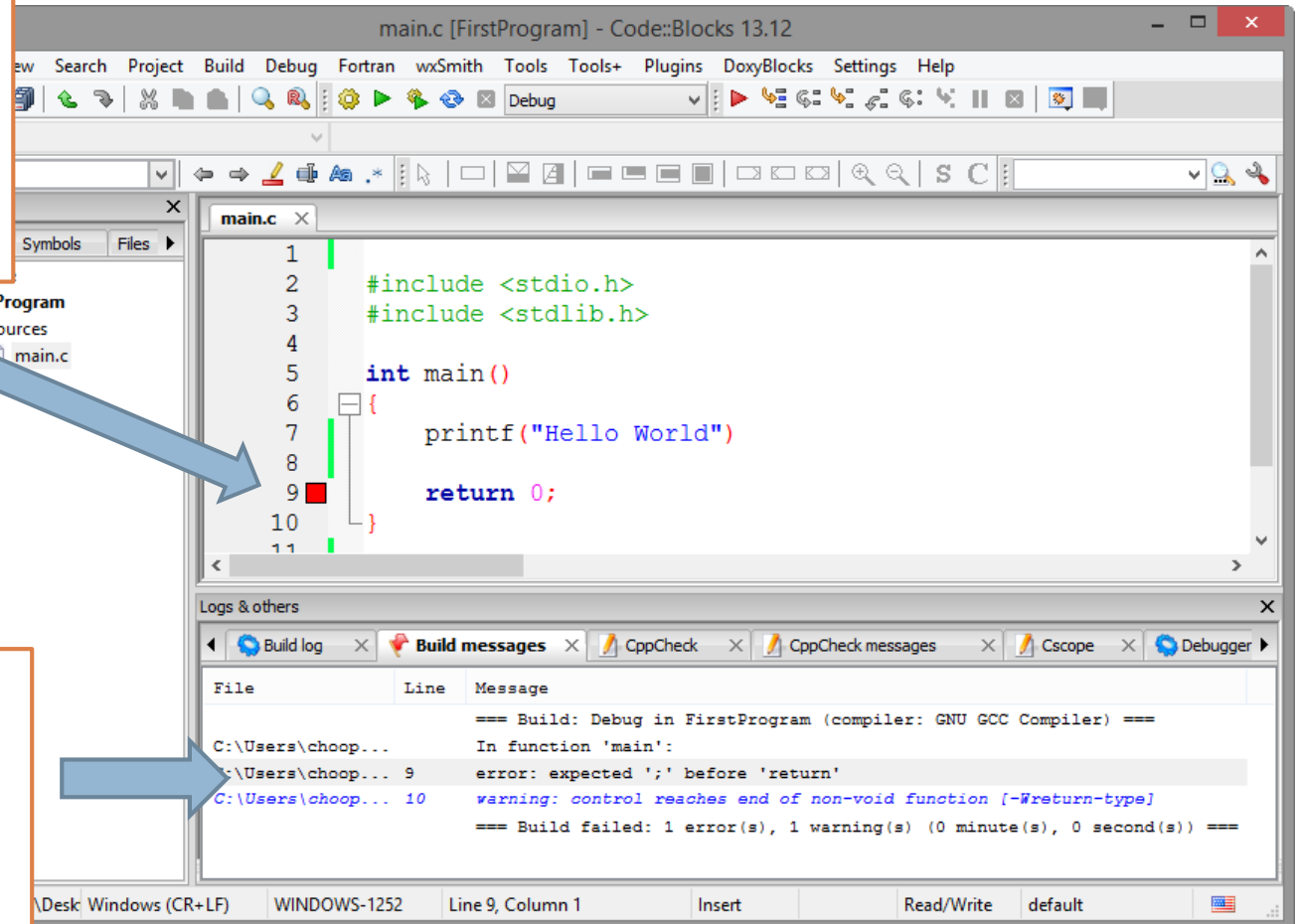


```
    return 0;
```

```
}
```

Compile Error

โปรแกรมจะแสดง จุด
สีเหลี่ยมสีแดง ใกล้ำๆ
บริเวณที่มีปัญหาในการ
Compile โปรแกรม



```
1
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  int main()
6  {
7      printf("Hello World")
8
9      return 0;
10 }
11
```

Logs & others

File	Line	Message
=== Build: Debug in FirstProgram (compiler: GNU GCC Compiler) ===		
In function 'main':		
C:\Users\choop...	9	error: expected ';' before 'return'
C:\Users\choop...	10	warning: control reaches end of non-void function [-Wreturn-type]
=== Build failed: 1 error(s), 1 warning(s) (0 minute(s), 0 second(s)) ===		

\\Desk Windows (CR+LF) WINDOWS-1252 Line 9, Column 1 Insert Read/Write default

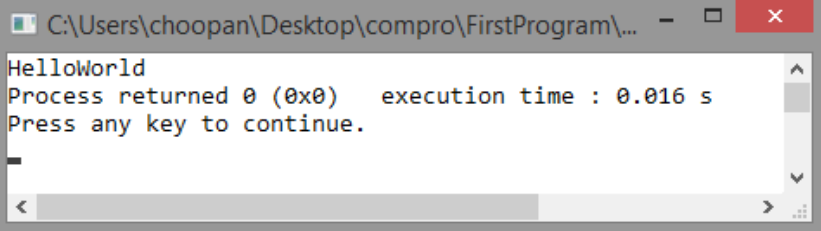
ข้อความแจ้งว่า ประมาณ
บรรทัดที่ 9
syntax error
before "return"

Hello World! (again)

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

```
int main()
{
    printf("Hello");
    printf("World");

    return 0;
}
```

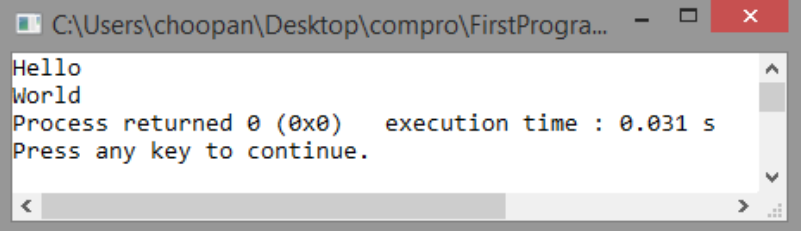


```
C:\Users\choopan\Desktop\compro\FirstProgram\...
HelloWorld
Process returned 0 (0x0)   execution time : 0.016 s
Press any key to continue.
```

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

```
int main()
{
    printf("Hello\n");
    printf("World");

    return 0;
}
```



```
C:\Users\choopan\Desktop\compro\FirstProgra...
Hello
World
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

\n เป็นตัวอักขระพิเศษที่ใช้บอก **printf** ว่าจะขึ้นบรรทัดใหม่

Quiz 1

- จงหาผลลัพธ์จากการทำงานของโปรแกรม

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

int main()
{
    printf("Hello");
    printf("1 2 3");
    printf("4 5 6");

    return 0;
}
```

Quiz 2

- จงหาผลลัพธ์จากการทำงานของโปรแกรม

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

int main()
{
    printf("Hello\nWorld");
    printf("1 2\n 3\n");

    return 0;
}
```

การใช้ **comment** ในภาษาซี

- การเขียน **comment** ในโปรแกรมต้นฉบับ
 - ▣ `//` เป็นการใช้งานแบบ **C++**
 - ข้อความหลัง `//` จนถึงสุดบรรทัดคอมไพเลอร์จะไม่สนใจ
 - ▣ `/*      */` เป็นแบบ **C**
 - ซึ่งคอมไพเลอร์จะไม่อ่านข้อความระหว่าง `/*` และ `*/`

QUIZ 3

- จงหาผลลัพธ์จากการทำงานของโปรแกรม

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

int main()
{
    printf("Hello World\n");
    // printf("1 2 3");
    printf("4 5 6");

    return 0;
}
```

QUIZ 4

- จงหาผลลัพธ์จากการทำงานของโปรแกรม

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

int main()
{
    /* printf("Hello World\n");
    printf("1 2 3");*/
    prinft("4 5 6");

    return 0;
}
```

QUIZ 5

- จงหาผลลัพธ์จากการทำงานของโปรแกรม

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

int main()
{
    /* printf("Hello World\n"); */
    printf("1 2 3");
    // printf("4 5 6");

    return 0;
}
```