

## หลักการพื้นฐานของภาษา

ภาษาโปรแกรม (Programming languages) มีการคิดค้นเพื่อนำมาใช้งานกับเครื่องคอมพิวเตอร์ มีหลายภาษาแต่ละภาษามีวัตถุประสงค์ในการใช้งานตามผู้คิดค้นเฉพาะด้าน ภาษาโปรแกรมที่รู้จักกันอย่างทั่วไปนั้นอาจจะมีบางภาษา เช่น โคบอล (Cobal) ปาสคาล (Pascal) เดลไฟ (Delphi) วิวิวลเบสิก (Visual BASIC) ซี (C) จาวา (Java) เป็นต้น ภาษาซี (C programming language) เป็นภาษาที่เรียกว่าภาษามีโครงสร้าง ในการออกแบบโปรแกรมในลักษณะชุดคำสั่ง โดยมีจุดเด่นด้านประสิทธิภาพการทำงานที่รวดเร็ว มีความยืดหยุ่นการเขียนโปรแกรมสูง

ภาษาซี เป็นภาษาโปรแกรมบนคอมพิวเตอร์ได้รับการสร้างขึ้นเมื่อ ค.ศ. 1972 โดย เดนนิส ริตชี (Dennis Ritchie) ที่ห้องปฏิบัติการเบลล์เทเลโฟน (Bell Telephone Laboratories) (Steven และ Lutfar, 2006)

ภาษาซีเป็นภาษาโปรแกรมหนึ่งที่ได้รับการนิยมนามากที่สุดตลอดกาล จัดลำดับโดย <http://www.langpop.com> ด้วยเพราะสถาปัตยกรรมคอมพิวเตอร์ เพียงส่วนน้อยเท่านั้น ที่ไม่มีตัวแปลโปรแกรมของภาษาซี ภาษาซีมีอิทธิพล (powerful) อย่างมากต่อภาษาโปรแกรมที่นิยมอื่น ๆ ที่เด่นชัดที่สุดก็คือ ภาษาซีพลัสพลัส (C++) ซึ่งเป็นส่วนขยายของภาษาซี

ภาษาซีเป็นภาษาที่มีความยืดหยุ่นสูง (flexible) ภาษาที่ใช้ในการเขียนโปรแกรมเป็นระบบเชิงคำสั่ง (หรือเชิงกระบวนการ) ถูกออกแบบขึ้น เพื่อใช้แปลด้วยตัวแปลโปรแกรมแบบการเชื่อมโยงที่ตรงไปตรงมา สามารถเข้าถึงหน่วยความจำในระดับล่าง เพื่อสร้างภาษาที่จับคู่อย่างมีประสิทธิภาพกับชุดคำสั่งเครื่อง

และแทบไม่ต้องการสนับสนุนใด ๆ ขณะทำงาน ภาษาซีจึงเป็นประโยชน์สำหรับหลายโปรแกรม

ในปี ค.ศ. 1983 ได้มีการจัดตั้งหน่วยงานมาตรฐานด้านการกำหนดรหัสของตัวอักษรของอเมริกาขึ้นเป็นหน่วยชื่อ American National Standards Institute (ANSI) สำหรับกำหนดมาตรฐานให้กับภาษาซี ขึ้นเพื่อเป็นข้อกำหนดในการแปลความหมายให้กับทุก ๆ รูปแบบที่มีการใช้งาน ภาษาซี (Brian, W. K., Online)

### ภาษาคอมพิวเตอร์

ภาษาคอมพิวเตอร์ หมายถึง ชุดคำสั่งที่เขียนขึ้นตามรูปแบบและโครงสร้างของภาษา เพื่อสั่งงานให้คอมพิวเตอร์ทำงานตามชุดคำสั่งหรือโปรแกรมที่ถูกเขียนขึ้นโดยโปรแกรมเมอร์ (Programmer) โดยทั่วไปภาษาคอมพิวเตอร์แบ่งออกตามความยากง่ายในการทำความเข้าใจของการเขียนโปรแกรมได้ 3 ระดับดังนี้

**ระดับที่ 1. ภาษาระดับต่ำ (low level language)** เป็นภาษาที่คอมพิวเตอร์ที่สั่งงานถึงการทำงานของฮาร์ดแวร์ ได้โดยตรง ภาษาระดับต่ำจึงทำงานได้เร็ว การเขียนโปรแกรมด้วยภาษาระดับต่ำ โปรแกรมเมอร์ต้องมีความรู้ด้านโครงสร้างระบบคอมพิวเตอร์ (system computer architecture) เช่น หน่วยประมวลผลกลาง หน่วยความจำ (ROM, RAM) หน่วยรับข้อมูล และหน่วยแสดงผล ว่ามีส่วนประกอบที่มีความเชื่อมโยง ทำงานร่วมกันกับส่วนใด และอย่างไรเป็นอย่างดี จึงทำให้สามารถเขียนโปรแกรมได้สมบูรณ์แบบ ซึ่งภาษาระดับต่ำสามารถแบ่งออกได้ 2 ภาษาคือ

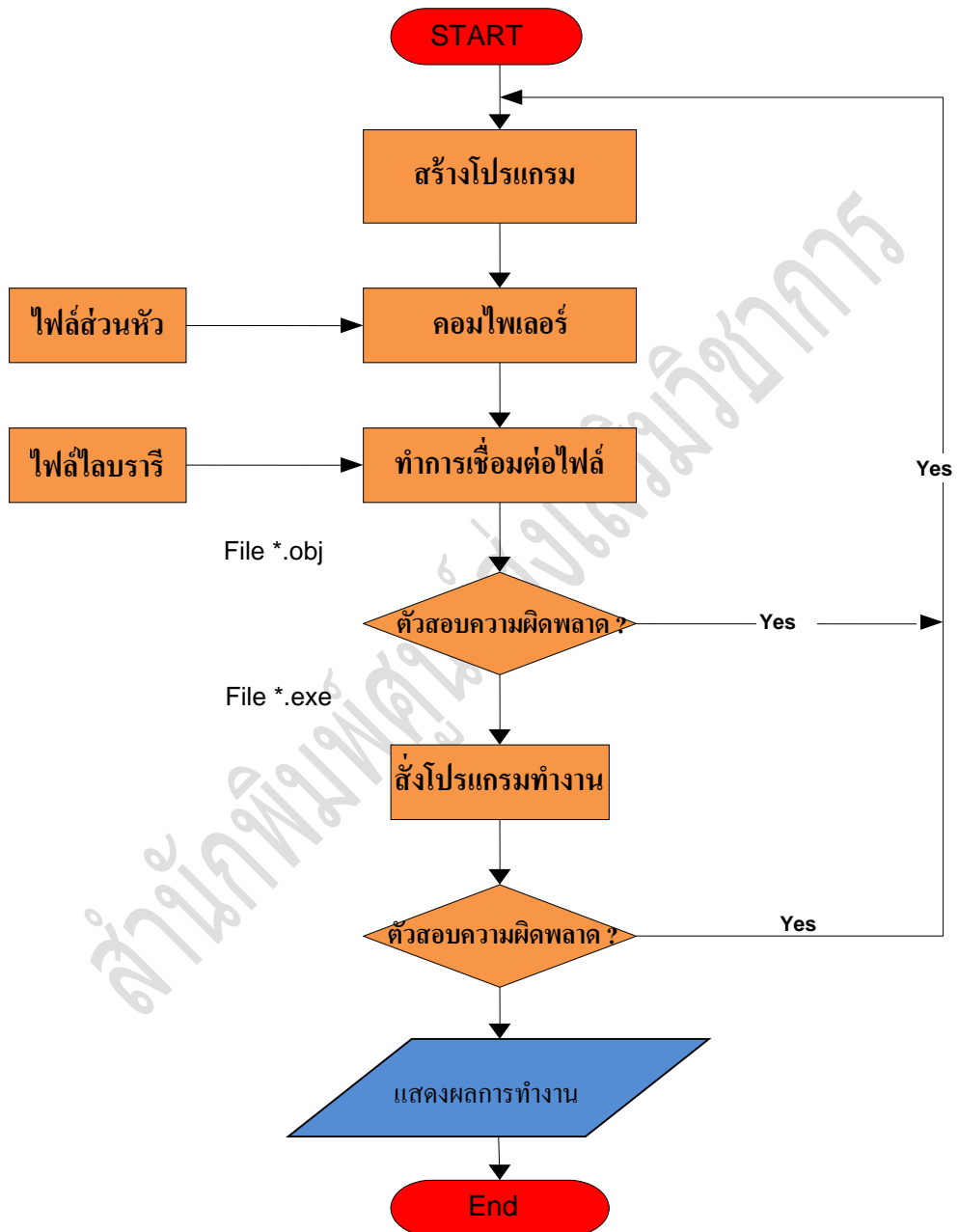
**ภาษาเครื่อง (machine language)** เป็นชุดคำสั่งที่อยู่ในระบบเลขฐาน 2 (binary number) ติดต่อกับฮาร์ดแวร์ได้โดยตรง คอมพิวเตอร์สามารถทำงานตามคำสั่งได้ทันที โดยไม่ต้องใช้ตัวแปลภาษาทำให้คอมพิวเตอร์สามารถทำงานตามที่โปรแกรมได้ทำการกำหนดให้อุปกรณ์ใดทำงานก่อนหลังอย่างไรได้โดยตรง

**ภาษาแอสเซมบลี (assembly language)** เป็นภาษาที่เขียนอยู่ในรูปแบบของชุดคำสั่งในระบบเลขฐาน 16 (hexadecimal number) เป็นภาษาที่ง่ายกว่าภาษาเครื่องแต่ก็ยังเข้าใจยากกว่าภาษาระดับกลางและภาษาระดับสูง การประมวลผลการทำงานได้รวดเร็ว การติดต่อกับฮาร์ดแวร์ทำได้ดี การสั่งงานให้คอมพิวเตอร์ทำงานต้องมีการแปลภาษาเป็นภาษาเครื่องก่อนโดยใช้ตัวแปลภาษาที่เรียกว่า แอสเซมเบลอร์ (assembler) ก่อนที่ถอดความหมายให้เครื่องคอมพิวเตอร์ทำงานตามที่กำหนดของโปรแกรม

**ระดับที่ 2. ภาษาระดับกลาง (medium level language)** เป็นภาษาที่มีลักษณะผสมกันระหว่างภาษาระดับสูงกับภาษาระดับต่ำ รูปแบบของคำสั่งคล้ายกับประโยคทางภาษาอังกฤษ บางคำสั่งนำรูปแบบภาษาระดับต่ำมาใช้งานร่วมด้วย การเขียนโปรแกรมได้จึงมีความเข้าใจได้เร็วกว่าภาษาระดับต่ำ ในสั่งงานให้คอมพิวเตอร์ทำงานจะต้องมีตัวแปลความหมายโดยใช้ตัวแปลที่เรียกกันว่า “คอมไพเลอร์” (compiler) ดังเช่น **ภาษาซี** เมื่อทำการสร้างโค้ดตัวโปรแกรม ตามรูปแบบของโครงสร้างภาษาซีแล้วต้องมามาผ่านตัวแปล คอมไพเลอร์ของภาษาซีก่อนจึงจะได้เพิ่มข้อมูลชนิดประมวลผล (exe) ที่นำไปทำงานบนเครื่องคอมพิวเตอร์ใด ๆ ก็สามารถทำงานตามที่โปรแกรมกำหนดได้

**ระดับที่ 3. ภาษาระดับสูง (high level language)** เป็นภาษาที่สามารถศึกษาและสามารถเข้าใจได้ง่าย มีลักษณะคำสั่งคล้ายกับประโยคทางภาษาอังกฤษในชีวิตประจำวัน จึงง่ายต่อการทำความเข้าใจและใช้เวลาในการเขียนโปรแกรมน้อย การสั่งงานให้คอมพิวเตอร์ทำงานจะมีขั้นตอนที่ซับซ้อนกว่าภาษาระดับต่ำและภาษาระดับกลาง การสั่งงานเป็นการสั่งงานได้เพียงบางส่วนของคอมพิวเตอร์เท่านั้น การแปลคำสั่งต้องใช้ตัวแปลภาษาระดับสูงเรียกว่า อินเตอร์พรีเตอร์ (Interpreter) หรือ คอมไพเลอร์ (compiler) เช่น ภาษาเบสิก (BASIC language), ปาสคาล (pascal language) และภาษาจาวา (java language) เป็นต้น โดยการทำงานเหมาะกับการโปรแกรมบนระบบเครื่องคอมพิวเตอร์ส่วนบุคคล (personal computer) ทั่วไป เช่นภาษาจาวา ในการทำงานของโปรแกรมต้องทำการคอมไพเลอร์ ก่อนจึงสามารถทำงานได้โดยขณะทำงานต้องอาศัยตัว อินเตอร์พรีเตอร์จาวา มาทำงานร่วมด้วยเสมอ ซึ่งแตกต่างกับกลุ่มของภาษาระดับต่ำกับภาษาระดับกลาง มักเป็นการเขียนโปรแกรมเพื่อนำไปควบคุมอุปกรณ์ไฟฟ้า อุปกรณ์อิเล็กทรอนิกส์ ตัวโปรแกรมนี้อาจมีขนาดเล็กถึงเล็กมาก กลุ่มภาษาระดับต่ำและกลุ่มภาษาระดับกลางนี้จึงสามารถเขียนโปรแกรมบน เครื่องคอมพิวเตอร์ที่เป็นเครื่องไมโครโปรเซสเซอร์ หรือนำมาเขียนโปรแกรมบนเครื่องอุปกรณ์ไฟฟ้าชนิดฝังตัว (embedded) ได้

## ขบวนการประมวลผลของภาษาซี



รูปที่ 1.1 โฟลว์ชาร์ตการประมวลผลของภาษาซี

การเขียนโปรแกรมภาษาซี เริ่มต้นจากการสร้างโปรแกรม (source program หรือ source code) ซึ่งต้องเขียนตามลักษณะโครงสร้างของภาษาซี (ตามรายละเอียดบทถัดไป) ทำการบันทึก (save หรือ save as) เป็นไฟล์นามสกุล .c นำโค้ดที่ได้ทำการเขียนทำการคอมไพล์ ทำการตรวจสอบข้อผิดพลาดว่ามีหรือไม่ ถ้ามีให้กลับไปแก้ไขที่โปรแกรม แต่ถ้าไม่มีข้อผิดพลาดตัวระบบภาษาซีจะสร้างไฟล์นามสกุล .obj แล้วทำการเชื่อมต่อไฟล์จากไฟล์ไลบรารีและระบบจะทำการสร้างไฟล์นามสกุล .obj และทำการตรวจสอบความผิดพลาดในการเชื่อมต่อ ถ้าไม่มีข้อผิดพลาดจะได้ไฟล์นามสกุล .exe ซึ่งเป็นไฟล์ที่พร้อมนำไปสั่งงานให้แสดงผลการทำงานตามที่กำหนดของโปรแกรม ไฟล์ที่เป็นนามสกุล .exe นี้ไม่สามารถทำการแก้ไขได้ด้วยเพราะเป็นไฟล์ที่เป็นรหัสให้เครื่องคอมพิวเตอร์ทำงาน ดังนั้นถ้าต้องการปรับปรุงหรือแก้ไขโปรแกรมผู้เขียนโปรแกรมต้องมีไฟล์ที่เป็นนามสกุล .c มาเข้าขบวนการประมวลผลภาษาซีใหม่ตั้งแต่เริ่มต้น

## ส่วนประกอบของภาษาซี

โครงสร้างโปรแกรมภาษาซี สามารถแบ่งออกได้เป็น 2 ส่วน คือ ไฟล์ส่วนหัว (header files) และส่วนของตัวโปรแกรม

### ไฟล์ส่วนหัวโปรแกรม (header files)

เป็นไฟล์ที่มีส่วนขยายเป็นนามสกุล \*.h เป็นไฟล์ที่สร้างฟังก์ชันใช้งานในการอำนวยความสะดวกให้กับผู้ใช้งานโปรแกรมภาษาซี ที่สามารถนำฟังก์ชันที่ถูกสร้างไว้เป็นมาตรฐานแล้วนำมาใช้งานได้ทันที ซึ่งฟังก์ชันต่างๆ ได้เก็บไว้ในไลบรารี (library) ของภาษาซีเพื่อใช้ร่วมกับ (include) ในการคอมไพล์โปรแกรม เช่น

stdio.h (standard input output) เป็นไฟล์ที่เก็บไลบรารีมาตรฐานเกี่ยวกับการรับข้อมูลและการแสดงผล ซึ่งฟังก์ชัน printf() ก็ถูกนิยามไว้ใน stdio.h ดังนั้นการเขียนโปรแกรมที่มีการเรียกใช้ฟังก์ชัน printf() ต้องทำการประกาศไฟล์ stdio.h เพื่อใช้รวมในการคอมไพล์โปรแกรมด้วย

## ส่วนของตัวโปรแกรม

โปรแกรมภาษาซีจะประกอบด้วยฟังก์ชัน ซึ่งอาจจะมีหนึ่งฟังก์ชัน หรือหลายฟังก์ชันก็ได้ แต่ต้องมีฟังก์ชัน main() ซึ่งเป็นฟังก์ชันหลักของโปรแกรมอยู่ด้วยเสมอ โดยฟังก์ชัน จะอยู่เป็นฟังก์ชันแรกหรืออยู่ส่วนใดของโปรแกรมก็ได้ แต่โดยทั่วไปนิยมกำหนดให้เป็นฟังก์ชันแรก โดยกำหนดให้มีเครื่องหมายปีกกาเปิด ({} เป็นเครื่องหมายเริ่มต้นการเขียนโปรแกรม เครื่องหมายปีกกาปิด (}) เป็นเครื่องหมายจบโปรแกรม ซึ่งภายในฟังก์ชัน main() จะประกอบไปด้วยชุดคำสั่งและฟังก์ชันต่าง ๆ ซึ่งเกือบทั้งหมดจะปิดท้ายด้วย เครื่องหมายเซมิคอลอน (;) การเขียนโปรแกรมภาษาซีจะเขียนด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็ก และส่วนคำอธิบาย (comment) จะใช้เครื่องหมาย /\* และ \*/ คั่นตามลำดับซึ่งจะใช้ในการอธิบายโปรแกรมให้ผู้สร้างและผู้พัฒนาโปรแกรมเข้าใจวัตถุประสงค์ของการเขียนคำสั่งหรือฟังก์ชันนั้นว่าเขียนไว้เพื่ออะไร ซึ่งส่วนคำอธิบายจะไม่มีผลต่อการคอมไพล์ (Alexander, Online)

## ตัวอย่าง 1.1 แสดงตัวอย่างโครงสร้างโปรแกรมภาษาซีเบื้องต้น

```
// ส่วนของคำอธิบาย (comment)

/* Example Program */

// ไฟล์ส่วนหัวของโปรแกรม (header files)
#include<stdio.h>

// ไฟล์ส่วนหัวของโปรแกรม (header files)
#include<conio.h>

// ฟังก์ชันหลักของโปรแกรม
main()
    // ปีกาเปิดเริ่มต้นการเขียนโปรแกรม
    {
        // ส่วนของตัวโปรแกรม
        clrscr();
        printf("*****\n");
        printf(" Information  \n");
        printf("*****\n");
        getch();
        return 0;
    // ปีกาปิดจบโปรแกรม
    }
```



## ผลการทำงาน โปรแกรม แสดงรูปที่ 1.2

The screenshot shows a Turbo C++ IDE window titled "Command Prompt - tc". The main editor area displays the following C code:

```

* Example Program */
#include<stdio.h>
main()
{
clrscr();
printf("*****\n");
printf(" Information  \n");
printf("*****\n");
getch();
return 0;
}

```

The status bar at the bottom of the editor shows "5:24" and "Message Output". Below the editor, the output window displays the result of the program's execution:

```

*****
Information
*****

```

The IDE's menu bar includes File, Edit, Search, Run, Compile, Debug, Project, Options, Window, and Help. The status bar at the very bottom lists function keys: F1 Help, Alt-F8 Next Msg, Alt-F7 Prev Msg, Alt-F9 Compile, F9 Make, and F10 Menu.

รูปที่ 1.2 แสดงตัวโปรแกรมภาษาซีและผลการทำงานของโปรแกรมตัวอย่าง

### ตัวอย่างการโปรแกรมภาษาซีเบื้องต้น

ตามตัวอย่างโปรแกรมภาษาซีนี้ เป็นการแสดงคุณสมบัติที่ใช้งานทั่วไปของภาษาซี ให้ความเข้าใจเบื้องต้น ดังนั้นผู้เขียนโปรแกรมยังไม่ต้องทำความเข้าใจกับรายละเอียดของไวยากรณ์ (syntax) จากตัวอย่าง แต่ให้ศึกษาคุณสมบัติต่างๆ ของภาษาซีที่จะมีส่วนคล้ายกับภาษาโปรแกรมอื่นๆ

จากตัวอย่างรูปที่ 1.2 ในการการเขียนโปรแกรมภาษาซี มีขั้นตอนดำเนินการโดยทำการเขียนโค้ดโปรแกรมลงบนส่วนของ พื้นที่การเขียนโปรแกรม ลำดับถัดไปสั่งให้ตัวโปรแกรมภาษาซี ทำการคอมไพล์เลอร์ (Compiler) โดยการสั่งที่เมนู

Compile หรือกด Alt+F9 ตัวโปรแกรมที่โปรแกรมเมอร์ได้ทำการเขียน ว่าถูกต้องหรือไม่ถ้ามีข้อผิดพลาดตัวคอมไพเลอร์ จะแจ้งว่าผิดพลาดในส่วนใด ถ้าโปรแกรมผ่านการคอมไพเลอร์โดยไม่มีข้อผิดพลาด ลำดับถัดมาต้องสั่งให้ทำการแสดงผลการทำงานของโปรแกรม โดยสั่งผ่านเมนู Run หรือ กด Ctrl+F9

## โครงสร้างของโปรแกรมภาษาซี

โปรแกรมภาษาซี เป็นการเขียนโปรแกรมแบบโครงสร้าง ซึ่งแต่ละโปรแกรมจะประกอบด้วยฟังก์ชันที่เป็นตัวฟังก์ชันหลักที่เรียกว่า main ต้องมีและยังต้องประกอบด้วยฟังก์ชันที่ทำหน้าที่อื่นๆ อีกหลายฟังก์ชัน โดยการจัดเรียงตำแหน่งของฟังก์ชันที่ประกอบอาจอยู่ก่อนหรือหลังฟังก์ชัน main ก็ได้ โดยในการประมวลผลการทำงานของโปรแกรมต้องกระทำในส่วนของ ฟังก์ชัน main ก่อนเสมอ หลังจากนั้นในฟังก์ชัน main ถ้ามีการเรียกใช้ฟังก์ชันใดๆ ได้ตามลำดับของการเขียนโปรแกรม (statement) ตามตัวอย่างที่ 1.2

## ตัวอย่างที่ 1.2 แสดงตัวอย่างการคำนวณพื้นที่วงกลม

```
#include<stdio.h>
#include<conio.h>
main()
{
    clrscr();
    float radius, area;
    printf("Radius = ?");
    scanf("%f",&radius);
    area = 3.14159 * radius * radius;
    printf("Area= %f", area);
    return 0;
}
```

จากตัวอย่าง 1.2 มีข้อสังเกตดังนี้

1. การพิมพ์คำสั่งทั้งหมดให้ใช้ตัวเล็ก (lowercase) เท่านั้น ส่วนบางส่วนของโปรแกรมที่ใช้ตัวคำสั่ง เป็นคำอธิบายหรือเป็นส่วนที่จะสื่อสารกับผู้ใช้โปรแกรมสามารถใช้ตัวพิมพ์ใหญ่รวมด้วยได้ เช่น ในคำสั่ง `printf("Radius = ?");` พบว่ามีคำว่า Radius มีตัวใหญ่คือ R ถือว่าต้องถูกเพราะเป็นข้อความที่ใช้ในการสื่อสารกับผู้ใช้โปรแกรมให้ทำการกรอกค่า รัศมีที่ต้องการให้โปรแกรมทำการคำนวณพื้นที่วงกลม

2. ในบรรทัดที่ 1 และ 2 มีการอ้างเพิ่มข้อมูลชื่อ `stdio.h` และ `conio.h` ตามลำดับ ซึ่งเป็นเพิ่มข้อมูลที่จะต้องรวมเข้ามาในตอนแปล โปรแกรม (compiler) ซึ่งตัวแปล โปรแกรมจะทำการรวมอัตโนมัติ
3. บรรทัดที่ 3 เป็นส่วนหัวของฟังก์ชัน `main` หลังจากชื่อฟังก์ชันต้องตามด้วย `()` เสมอ โดยในวงเล็บไม่มีข้อมูลใดๆอยู่แสดงว่า ฟังก์ชันนี้ไม่ต้องอาร์กิวเมนต์ (argument)
4. ส่วนบรรทัดที่อยู่ภายใต้วงเล็บปีกกา จำนวน 7 บรรทัดเป็นฟังก์ชันที่เป็นประโยคเชิงซ้อน อยู่ในฟังก์ชัน `main`
5. บรรทัดแรกในฟังก์ชัน `main` ที่ย่อหน้าเข้ามาเป็นการเรียกใช้ฟังก์ชัน `clrscr()` จาก `conio.h` ทำหน้าที่ในการล้างข้อความหน้าจอภาพ
6. บรรทัดถัดมาเป็นการประกาศ (จอง, สร้าง) ตัวแปล (variable declaration) ชื่อ `radius` และ `area` ที่เป็นตัวแปลชนิดใช้เก็บค่าตัวเลขที่เป็นทศนิยม (รายละเอียดจะกล่าวภายหลัง)
7. บรรทัดถัดมาเป็นการแสดงข้อความเพื่อขอข้อมูล ในการรับข้อมูลจากคำสั่ง `scanf` (รายละเอียดจะกล่าวภายหลัง)
8. เป็นบรรทัดที่สั่งให้ทำการคำนวณ เป็นประโยคกำหนดค่า (assignment statement) โดยการคำนวณค่าพื้นที่จากค่ารัศมีที่ป้อนเข้าไป กับเครื่องหมายดอกจัน (\*) ในที่นี้หมายถึงการคูณ แล้วนำค่าเข้าไปเก็บตัวแปรชื่อ `area`

9. เป็นคำสั่งสุดท้าย (printf) แสดงค่าคำนวณ โดยมีข้อความ Area= อธิบายผลการแสดงการคำนวณค่าพื้นที่
10. ที่สำคัญในการเขียนโปรแกรมของแต่ละบรรทัดหรือคำสั่ง ต้องจบด้วยเครื่องหมายอัฒภาค (;) เสมอ
11. ในการเว้นวรรค ย่อหน้า หรือการจัดบรรทัดเป็นการให้การอ่านโปรแกรมเข้าง่าย และถือว่าการเขียนโปรแกรมที่ดี

### ผลการทำงานโปรแกรมได้ดังนี้

Radius =? 12 (ตัวอย่างผู้ใช้โปรแกรมกรอกค่ารัศมี เท่ากับ 12)

Area = 452.88947

โดยการกรอกค่า รัศมีในโปรแกรมทำการคำนวณในที่นี้คือ 12 จึงได้ผลการคำนวณพื้นที่วงกลมมีค่าเท่ากับ 452.88947

### สรุป

ภาษาโปรแกรมคอมพิวเตอร์ มีการคิดค้นพัฒนาอยู่ตลอดเวลา ตั้งแต่อดีตถึงปัจจุบัน เพื่อให้ การเขียนโปรแกรมคอมพิวเตอร์ตอบสนองการใช้งาน ให้มากที่สุด ดังนั้นในการพัฒนาภาษาคอมพิวเตอร์ จึงมี 3 ระดับ คือ ภาษาระดับต่ำ เป็นภาษาคอมพิวเตอร์ที่ผู้เขียน โปรแกรม ต้องเข้าใจฮาร์ดแวร์เป็นอย่างดี ภาษาระดับกลางเป็นภาษาที่มีผู้เขียนโปรแกรมมีความรู้ฮาร์ดแวร์ระดับโครงสร้าง ของสถาปัตยกรรมคอมพิวเตอร์ ก็สามารถเขียนโปรแกรมควบคุมได้ เช่น ภาษาซีนั่นเอง ส่วนระดับสุดท้ายคือ ภาษาระดับสูง เป็นภาษาที่มีผู้เขียนโปรแกรม มีความรู้ในระดับการใช้งาน ก็สามารถเขียนโปรแกรมได้แล้ว

## แบบฝึกหัดท้ายบทที่ 1

**ตอนที่ 1** จงเติมคำหรือข้อความในช่องว่างต่อไปนี้ให้ถูกต้อง

1. โปรแกรมภาษาโครงสร้างประกอบด้วยโปรแกรมใดบ้าง  
.....
2. ภาษาในการเขียนโปรแกรมภาษาใดเป็นที่นิยมมากที่สุด  
.....
3. หน่วยงานมาตรฐานที่ทำหน้าที่ในการกำหนดรหัสตัวอักษรชื่อเต็มว่า  
.....
4. ภาษาคอมพิวเตอร์ระดับต่ำประกอบด้วยภาษาใดบ้าง  
.....
5. ไลต์ภาษาซีเมื่อทำการบันทึกต้องบันทึกเป็นนามสกุลหรือแฟ้มชนิดใด  
.....
6. แฟ้มที่สร้างขึ้นเมื่อทำการแปลคำสั่งภาษาซีสมบูรณ์ประกอบด้วยแฟ้มชนิดใดบ้าง  
.....
7. แฟ้มชนิดใดที่ไม่สามารถอ่านเมื่อถูกแปลคำสั่งแล้วไม่สามารถนำมาอ่านหรือแก้ไขคำสั่งได้  
.....

8. เมื่อโค้ดถูกแปลคำสั่งและทำการเชื่อมต่อแฟ้มไลบรารีสมบรณ์แล้วจะทำการสร้างเพิ่มขึ้นเป็นนามสกุลใด

.....

9. เครื่องหมายคำสั่งใดของภาษาซีทำหน้าที่กำหนดขอบเขตจุดเริ่มต้นและจุดสิ้นสุดของโปรแกรมหรือฟังก์ชัน

.....

10. โค้ดโปรแกรมภาษาซีสามารถเขียนโดยไม่ต้องขึ้นบรรทัดใหม่ได้หรือไม่

.....

## ตอนที่ 2 จงทำเครื่องหมายกากบาท (x) ทับหน้าข้อที่ถูกต้องที่สุด

1. ภาษาซีจัดเป็นภาษาโปรแกรมคอมพิวเตอร์ระดับใด
 

ก. ภาษาเครื่อง	ข. ภาษาแอสเซมบลี
ค. ภาษาระดับกลาง	ง. ภาษาระดับสูง
2. เพิ่มข้อมูลชนิดใด ไม่มี ความเกี่ยวข้องกับกระบวนการพัฒนาภาษาซี
 

ก. .c	ข. .obj
ค. .exe	ง. .abc
3. เพิ่มข้อมูลภาษาซีที่ผู้เขียนโปรแกรมต้องทำการเก็บไว้เป็นลิสต์ส่วนตัว เป็นเพิ่มข้อมูลชนิดใด
 

ก. .c	ข. .obj
ค. .exe	ง. .abc
4. เพิ่มข้อมูลชนิดใดที่ผู้เขียนโปรแกรมควรส่งให้ลูกค้านำไปใช้งานเพื่อ ป้องกันการสำเนาโค้ด โปรแกรม
 

ก. .c	ข. .obj
ค. .exe	ง. .abc
5. การสั่งให้ทำการแปลคำสั่ง (compiler) สามารถสั่งด้วยคีย์ลัดข้อใด
 

ก. Ctrl+F9	ข. Alt+F9
ค. Ctrl+F1	ง. Alt+F1
6. เมื่อแปลคำสั่งสมบูรณ์แล้วต้องการให้โปรแกรมทำงานสามารถสั่งด้วยคีย์ ลัดข้อใด
 

ก. Ctrl+F9	ข. Alt+F9
ค. Ctrl+F1	ง. Alt+F1
7. ฟังก์ชันหลักของการเขียนโปรแกรมภาษาซีทุกครั้งต้องมีฟังก์ชันนี้เสมอ คือฟังก์ชันใด
 

ก. main()	ข. printf()
ค. getch()	ง. return 0





## เอกสารอ้างอิง

- ศรันชัย อินทโกสุม (2539). ทฤษฎีและตัวอย่างโจทย์การเขียนโปรแกรมด้วยภาษาซี  
 กรุงเทพฯ : แมคกรอฮิล อินเทอร์เน็ต เนชั่นแนล เอ็นเตอร์ไพรส์, ینگค์.
- ธันวา ศรีประโมง (2539). การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม. พิมพ์ครั้งที่  
 ที่ 4. กรุงเทพฯ : มหาวิทยาลัยเทคโนโลยีมหานคร.
- วิจักขณ์ ศรีสังจะเลิศวาจา และคุณฤๅ ประเสริฐธิตพิงษ์ ออนไลน์ :  
[www.satit.su.ac.th/soottin](http://www.satit.su.ac.th/soottin).
- Alexander, A. Tutorial Online : <http://www.cprogramming.com>.
- Brian, W. K. Programming in C: A Tutorial Online :  
<http://www.lysator.liu.se/c/bwktutor.html>.
- DedaSys. (2009). Programming Language Popularity Online :  
<http://www.langpop.com>.
- Dennis R., (1971). Borland replaced Turbo C with Turbo C++ Online :  
[http://www.cprogrammingexpert.com/C/Tutorial/turbo\\_cpp\\_ide.aspx](http://www.cprogrammingexpert.com/C/Tutorial/turbo_cpp_ide.aspx)
- Steven, H. & Lutfar, R. (2006). Art of Programming Contest: C Programming |  
 Data Structure | Algorithms (ACM supported), 2<sup>nd</sup> Edition.