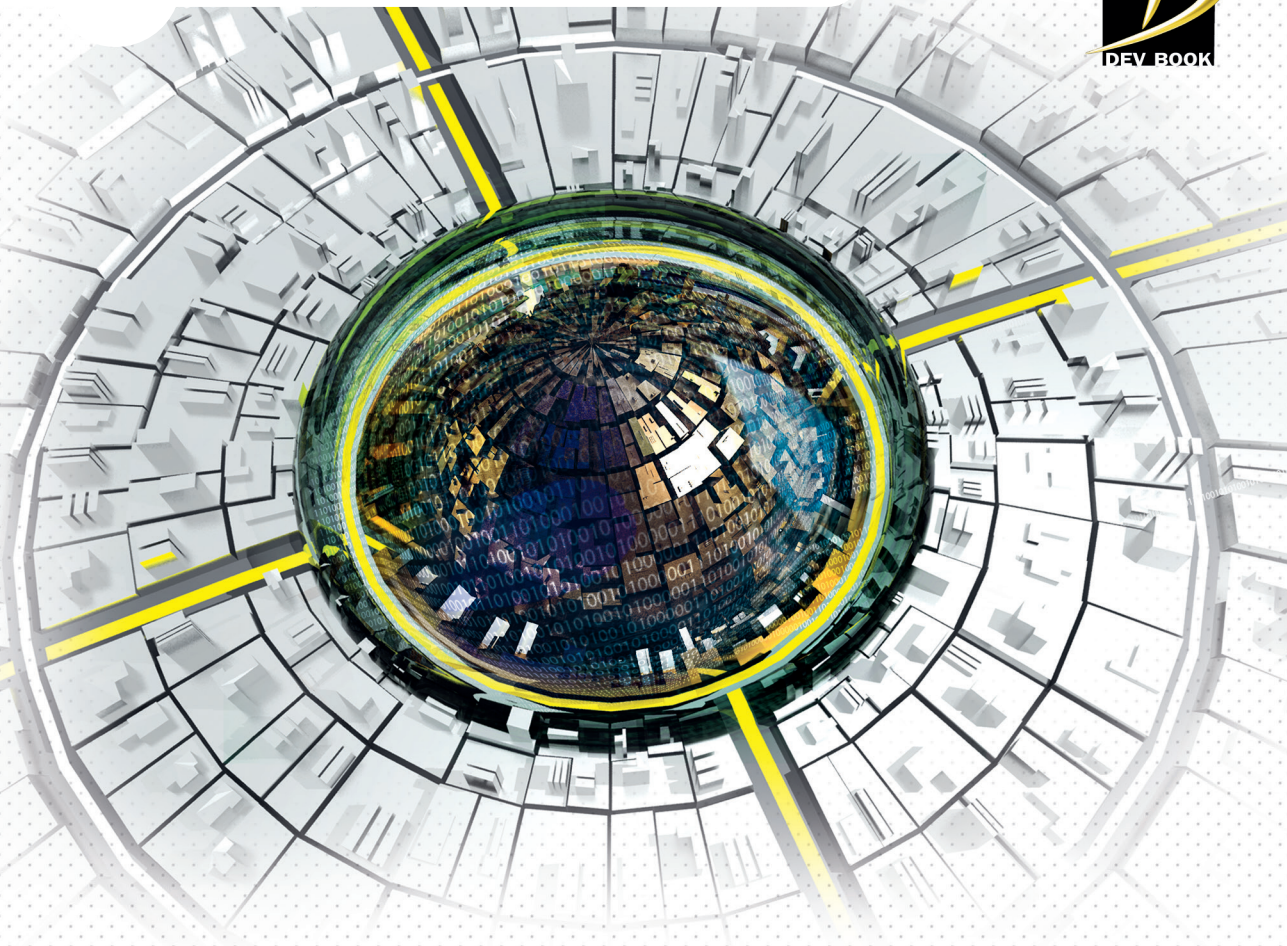
DEV BOOK



คู่มือเรียนเขียนโปรแกรม ภาษา C

ฉบับสมบูรณ์

เรียนรู้
ตั้งแต่พื้นฐานอย่างละเอียด
เป็นขั้นตอน

อ่านเข้าใจง่าย
มีแบบฝึกหัดและตัวอย่าง
พร้อมคำอธิบายทุกหัวข้อ

เหมาะสำหรับ
นักเรียน นักศึกษา และผู้ที่สนใจ
เขียนโปรแกรมภาษา C

ไกรศร ตั้งโอภากุล, กิตินันท์ พลสวัสดิ์

บรรณาธิการ

หนังสือเล่มนี้เป็นหนังสือเรียนเขียนโปรแกรมภาษา C ที่ปรับปรุงมาจากหนังสือ “คู่มือเขียนโปรแกรมภาษา C ฉบับผู้เริ่มต้น” ที่เรียบเรียงโดย คุณประภาพร ช่างไม้ ของ DEV BOOK ซึ่งใช้ในการเรียนการสอนในระดับอุดมศึกษาหลายๆ สถาบันมามากกว่า 10 ปี โดยมีวัตถุประสงค์เพื่อปรับปรุงเนื้อหาและคุณภาพของหนังสือให้ดียิ่งขึ้นตามคำติชมจากผู้อ่าน

ภายในเล่มนี้ยังคงรูปแบบโครงสร้างเนื้อหา ตามหลักสูตรการเรียนการสอนในระดับอุดมศึกษา ตั้งแต่ชนิดข้อมูล ตัวดำเนินการ การรับและแสดงผลข้อมูล คำสั่งการทำงานของโปรแกรม ยังรวมไปถึงการสร้างและใช้งานฟังก์ชัน การเขียนโปรแกรมทำงานกับชนิดข้อมูลต่างๆ ไม่ว่าจะเป็นอาร์เรย์ พอยน์เตอร์ หรือโครงสร้างข้อมูลแบบต่างๆ อีกทั้งยังมีตัวอย่างประกอบการอธิบายอย่างละเอียดทุกหัวข้อ พร้อมแบบฝึกหัดท้ายบทให้ผู้อ่านได้ฝึกทักษะการเขียนโปรแกรม ซึ่งจะเห็นได้ว่าเหมาะสำหรับใช้ในการเรียนการสอนการเขียนโปรแกรมภาษา C เป็นอย่างยิ่ง

อย่างไรก็ตามทาง DEV BOOK ขอขอบคุณผู้อ่านทุกๆ ท่านที่ติดตามหนังสือของเราเสมอมา และขอขอบคุณทุกๆ คำติชมที่ทำให้เราได้ปรับปรุงและพัฒนาอยู่เสมอ ทาง DEV BOOK หวังเป็นอย่างยิ่งว่า หนังสือเล่มนี้จะเป็นหนังสืออีกเล่มหนึ่งที่ช่วยให้ผู้อ่านทุกท่านเข้าใจและเรียนรู้การเขียนโปรแกรมภาษา C ได้อย่างดีเยี่ยม

สุดท้ายนี้หากหนังสือเล่มนี้มีข้อผิดพลาดประการใด ต้องขออภัยมา ณ ที่นี้ด้วย

ด้วยความเคารพยิ่ง
กิตินันท์ พลสวัสดิ์

คำนำ

ภาษาคอมพิวเตอร์ไม่ว่าจะเป็นภาษา Assembly, C, Java, Visual Basic และอื่นๆ อีกมากมาย ทุกๆ ภาษามีจุดประสงค์หลักอย่างเดียวกันคือ ให้นักพัฒนาโปรแกรมหรือโปรแกรมเมอร์สามารถเขียนโปรแกรมให้ได้ผลลัพธ์ออกมาตามต้องการได้ โดยแต่ละภาษาอาจจะมีแนวคิดในการพัฒนาที่แตกต่างกันไปบ้าง แต่อย่างไรก็ตามทุกภาษาจะได้ผลสำเร็จเหมือนกันคือ ผลลัพธ์การทำงานของโปรแกรม

ภาษา C เป็นภาษาระดับสูงที่ถูกพัฒนาขึ้นเพื่อนำไปใช้ในการสร้างระบบปฏิบัติการ ต่อมาได้พัฒนาต่อเพื่อช่วยให้โปรแกรมเมอร์พัฒนาโปรแกรมตามต้องการได้ง่ายยิ่งขึ้น และถือได้ว่าเป็นภาษาที่เหมาะสำหรับผู้เริ่มต้นเรียนรู้การเขียนโปรแกรม เนื่องจากเป็นภาษาเชิงโครงสร้างที่เข้าใจง่าย ไม่ซับซ้อน โดยเน้นที่ขั้นตอนการทำงานของโปรแกรมเป็นหลักสำคัญ ทำให้เข้าใจแนวคิดการพัฒนาโปรแกรมได้เป็นอย่างดี

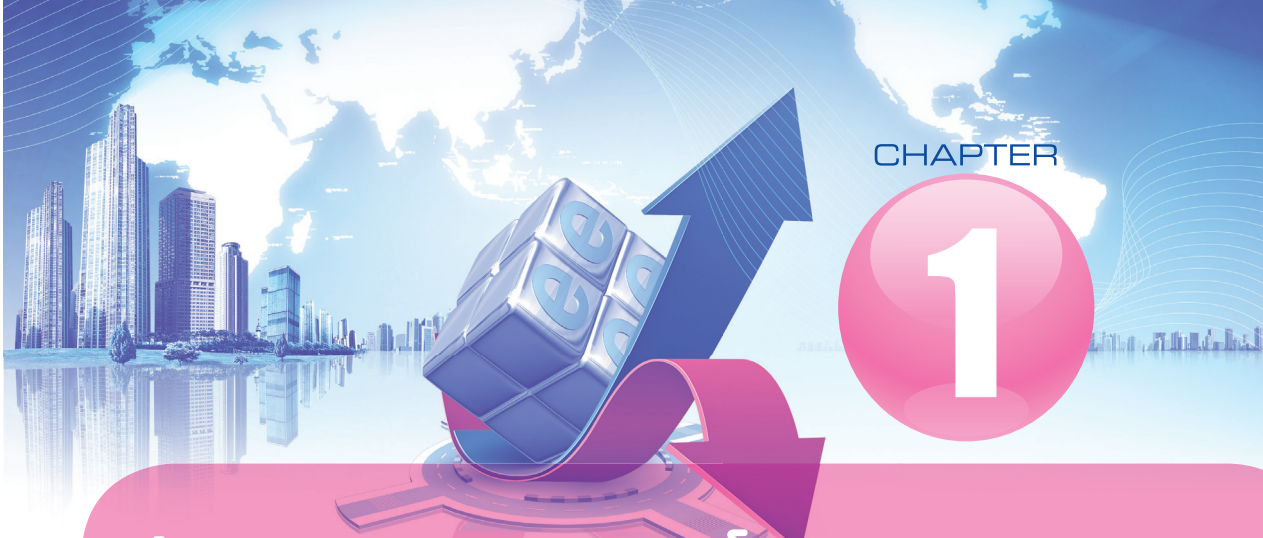
ดังนั้น หนังสือเล่มนี้ผู้เขียนเรียบเรียงขึ้นสำหรับผู้ที่สนใจเริ่มหัดเขียนโปรแกรม โดยเริ่มจากให้ผู้อ่านเข้าใจการทำงานของภาษาโปรแกรม รู้จักเครื่องมือสำหรับใช้พัฒนาโปรแกรม และหลักการเขียนโปรแกรมพื้นฐานที่ควรรู้ เช่น ชนิดข้อมูล ตัวดำเนินการ การรับและแสดงผลข้อมูล คำสั่งการทำงานของโปรแกรม เป็นต้น นอกจากนี้ยังรวมไปถึงการสร้างและใช้งานฟังก์ชัน การเขียนโปรแกรมทำงานกับชนิดข้อมูลต่างๆ ไม่ว่าจะเป็นอาร์เรย์ พอยน์เตอร์ และโครงสร้างข้อมูลแบบต่างๆ ซึ่งถือได้ว่าเหมาะสำหรับผู้เริ่มต้นเขียนโปรแกรมภาษา C หรือใช้ประกอบการเรียนการสอนเป็นอย่างยิ่ง

สุดท้ายนี้ผู้เขียนหวังเป็นอย่างยิ่งว่า หนังสือเล่มนี้จะช่วยให้ผู้อ่านทุกๆ ท่านเข้าใจการเขียนโปรแกรมด้วยภาษา C และสามารถนำไปประยุกต์ใช้งานต่อไปได้ และหากหนังสือเล่มนี้มีข้อผิดพลาดประการใด ต้องขออภัยมา ณ ที่นี้ด้วย

ขอขอบพระคุณบิดามารดา ที่ให้กำเนิดและเลี้ยงดูจนเติบใหญ่
ขอขอบพระคุณครู อาจารย์ ที่ท่านประทานความรู้มาให้

ด้วยความเคารพยิ่ง
ไกรศร ตั้งโอภากุล

Contents



CHAPTER

1

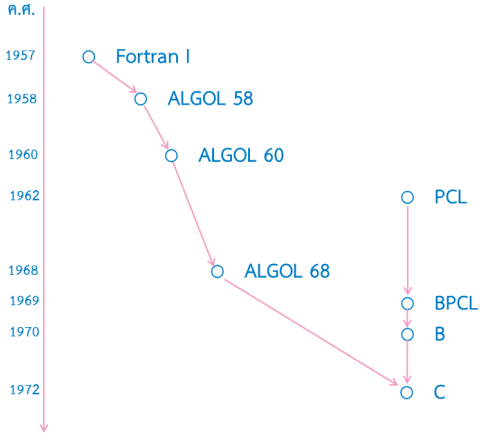
รู้จักภาษา C กับการพัฒนาโปรแกรม

ภาษา C เป็นภาษาระดับสูง ที่ถูกสร้างขึ้นมาเพื่อให้การเขียนโปรแกรมง่ายยิ่งขึ้น ปัจจุบันถือว่าเป็นภาษาคอมพิวเตอร์สำหรับผู้เริ่มต้นที่ต้องการเรียนรู้การเขียนโปรแกรม เนื่องจากเป็นภาษาเชิงโครงสร้างที่มีหลักการทำงานไม่ซับซ้อน เข้าใจง่าย

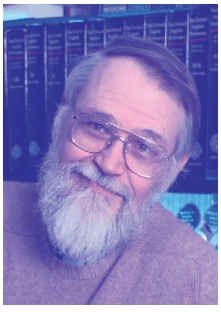
เนื้อหาในบทนี้ผู้อ่านจะได้ทราบประวัติความเป็นมา, จุดเด่น, โครงสร้างของภาษา C และนอกจากนี้ผู้เขียนจะแนะนำเครื่องมือที่ใช้ในการเขียนโปรแกรมภาษา C ที่ง่ายและสะดวกให้กับผู้อ่านด้วย

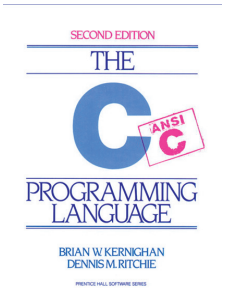
ความเป็นมาของภาษา C

ภาษา C เป็นภาษาคอมพิวเตอร์ที่ได้พัฒนาขึ้นในปี ค.ศ. 1972 (พ.ศ. 2515) โดยเดนนิส ริสซี (Dennis Ritchie) แห่ง Bell Telephone Laboratories, Inc. (ในปัจจุบันก็คือ AT & T Bell Laboratories) ซึ่งภาษา C มีการพัฒนามาจากภาษา B ในช่วงแรกภาษา C ได้ถูกนำมาใช้เพื่อสร้างระบบปฏิบัติการ Unix หากนำวิวัฒนาการของภาษา C มาแสดงออกเป็นแผนภาพจะได้ดังนี้



ในปี ค.ศ. 1978 (พ.ศ. 2521) เดนนิส ริสซี และนายเบรน เครนิกฮาน (Dennis Ritchie and Brian W. Kernighan) ได้แต่งหนังสือชื่อ "The C Programming Language" โดยนำเสนอภาษา C ที่สามารถนำมาปรับใช้กับคอมพิวเตอร์ในรูปแบบต่างๆ ได้มากยิ่งขึ้น และทำให้ภาษา C ได้รับความนิยมอย่างมาก จนกระทั่งในปี ค.ศ. 1988 (พ.ศ. 2531) ได้มีการสร้างมาตรฐานของภาษา C ขึ้นมาในชื่อของ ANSI C ภายใต้ความร่วมมือระหว่างสถาบัน ANSI (American National Standards Institute) กับนายเดนนิส ริสซี และนายเบรน เครนิกฮาน





▲ รูปซ้ายมือคือ Mr.Dennis Ritchie, รูปตรงกลาง Mr.Brian W. Kernighan และรูปขวามือคือหนังสือที่ทั้งคู่ได้แต่งขึ้นร่วมกัน

ในปี ค.ศ 1990 (พ.ศ. 2533) องค์กรมาตรฐานสากล หรือ ISO (International Standards Organization) ได้ยอมรับมาตรฐานที่ได้สร้างขึ้นมานี้ ภายใต้ชื่อ ANSI/ISO C

จุดเด่นของภาษา C

ในภาษาโปรแกรมทุกภาษามีจุดเด่นในการประมวลผลของภาษาแตกต่างกัน ในข้อหัวนี้ผู้อ่านจะได้รู้จักกับจุดเด่นของภาษา C ซึ่งมีดังนี้

- เป็นภาษาคอมพิวเตอร์ที่มีแนวคิดในการพัฒนาแบบ "โปรแกรมเชิงโครงสร้าง (Structure Programming)" ทำให้ภาษา C เป็นภาษาที่เหมาะสำหรับนำมาพัฒนาระบบ
- เป็นภาษาคอมพิวเตอร์ที่เป็นภาษามาตรฐาน ซึ่งการทำงานของภาษาจะไม่ขึ้นกับฮาร์ดแวร์ ทำให้สามารถนำไปใช้ใน CPU รุ่นต่างๆ ได้
- เป็นภาษาระดับสูงที่ทำงานเหมือนภาษาระดับต่ำ สามารถทำงานแทนภาษา Assembly ได้
- ความสามารถของคอมไพเลอร์ในภาษา C มีประสิทธิภาพสูง ทำงานได้รวดเร็ว โดยใช้รหัสออบเจ็กต์ (Object) ที่สั้น ทำให้เหมาะสำหรับงานที่ต้องการความรวดเร็ว

TIPS

คอมไพเลอร์ (Complier) จะทำหน้าที่แปลงภาษาโปรแกรมทั้งหมดที่เขียนให้เป็นภาษาเครื่อง พร้อมกันนี้จะทำหน้าที่ตรวจสอบไวยากรณ์ของภาษา หากมีข้อผิดพลาดก็จะส่งข้อความหรือคำเตือนออกมา เมื่อกระบวนการทำงานของคอมไพเลอร์เสร็จสมบูรณ์ โปรแกรมจึงเริ่มทำงาน ซึ่งแตกต่างกับหลักการทำงานของ **อินเตอร์พรีเตอร์ (interpreter)** ซึ่งการทำงานจะแปลงภาษาโปรแกรมและประมวลผลคำสั่งทีละคำสั่ง

โครงสร้างของภาษา C

ในโปรแกรมที่พัฒนาด้วยภาษา C ทุกโปรแกรมจะมีโครงสร้างการพัฒนาที่ไม่แตกต่างกัน ซึ่งประกอบด้วย 6 ส่วนหลักๆ โดยแต่ละส่วนจะมีหน้าที่แตกต่างกันดังนี้

1. **ส่วนของเฮดเดอร์ไฟล์** (Header File or Processing Directive) ส่วนนี้จะมีจุดสังเกตที่สำคัญคือจะขึ้นต้นด้วยเครื่องหมาย # เสมอ การทำงานของคอมไพเลอร์จะทำงานในส่วนนี้เป็นส่วนแรก ในส่วนของเฮดเดอร์ไฟล์จะเป็นส่วนที่เก็บไลบรารีมาตรฐานของภาษา C ซึ่งจะถูกดึงเข้ามารวมกับโปรแกรมในขณะที่กำลังคอมไพล์ โดยใช้คำสั่ง #include ซึ่งสามารถเขียนได้เป็น 2 รูปแบบคือ

รูปแบบที่ 1

```
#include<HeaderName>
```

รูปแบบที่ 2

```
#include "HeaderName"
```

โดยที่ HeaderName เป็นชื่อ Header File

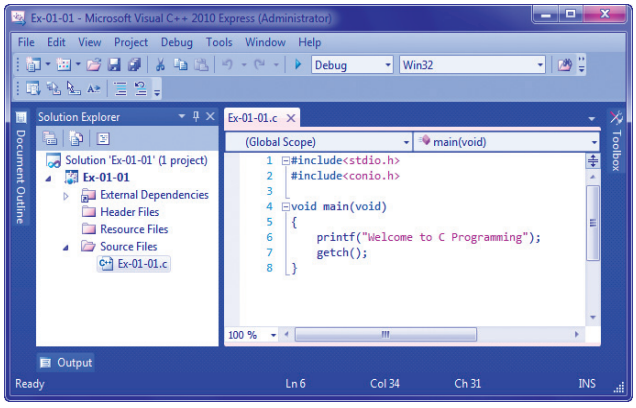
ผู้อ่านคงสงสัยแล้วว่า ทั้งสองแบบนี้มีความแตกต่างกันอย่างไร ทั้งสองแบบนี้ต่างกันที่การทำงานคือแบบที่ใช้เครื่องหมาย <...> คอมไพเลอร์จะค้นหาเฮดเดอร์ไฟล์จากไลบรารีของภาษา C เพียงที่เดียวเท่านั้น ส่วนที่ใช้เครื่องหมาย "..." คอมไพเลอร์จะค้นหาเฮดเดอร์ไฟล์จากไลบรารีที่เก็บ Source Code ของเราก่อน ถ้าหากไม่เจอก็จะไปค้นหาที่ไลบรารีของภาษา C และในจุดที่ผู้อ่านควรสังเกตคือ เฮดเดอร์ไฟล์นี้จะมีสกุลไฟล์เป็น .h เท่านั้น

ในการเขียนโปรแกรมภาษา C เฮดเดอร์ไฟล์ที่เก็บไลบรารีมาตรฐานในการจัดการเกี่ยวกับอินพุตและเอาต์พุตของโปรแกรมก็คือ stdio.h ซึ่งถือว่าเป็นส่วนสำคัญที่ขาดไม่ได้

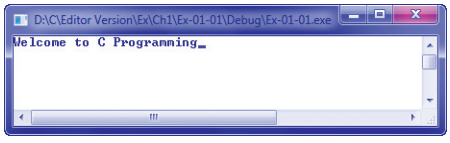
2. **ส่วนของตัวแปร Global** เป็นส่วนประกาศตัวแปรที่สามารถใช้ร่วมกันได้ทั้งโปรแกรม ซึ่งส่วนนี้จะมีหรือไม่มีก็ได้
3. **ส่วนของฟังก์ชัน** (Function) เป็นส่วนการทำงานของโปรแกรม ในโครงสร้างของภาษา C จะบังคับให้มีอย่างน้อย 1 ฟังก์ชันคือ ฟังก์ชัน main() ซึ่งเป็นฟังก์ชันเริ่มการทำงานของโปรแกรม (คอมไพเลอร์จะประมวลผลที่ฟังก์ชัน main() เป็นฟังก์ชันแรก) โดยในขอบเขตของฟังก์ชันจะเริ่มต้นด้วยเครื่องหมาย { และสิ้นสุดด้วยเครื่องหมาย }

4. **ส่วนของตัวแปร Local** เป็นส่วนประกาศตัวแปรที่สามารถใช้ได้เฉพาะภายในฟังก์ชันของตนเองเท่านั้น ซึ่งส่วนนี้จะมีหรือไม่มีก็ได้
5. **ส่วนของตัวโปรแกรม** เป็นส่วนคำสั่งการทำงานของโปรแกรม โดยที่คำสั่งในแต่ละคำสั่งจะต้องจบด้วยเครื่องหมาย ; เสมอ
6. **ส่วนของตัวส่งค่ากลับ** เป็นส่วนของการส่งค่าข้อมูลกลับเมื่อฟังก์ชันจบการทำงาน โดยค่าที่ส่งกลับนั้น จะต้องเป็นค่าที่มีชนิดของข้อมูลตรงกับชนิดของข้อมูลที่ฟังก์ชันคืนค่ากลับ (return type) ในกรณีที่ไม่ต้องการให้ฟังก์ชันมีการส่งค่ากลับ สามารถกำหนดได้โดยใช้คีย์เวิร์ด void

ตัวอย่างที่ 1.1 เป็นตัวอย่างการเขียนโปรแกรมภาษา C อย่างง่าย เพื่อแสดงส่วนต่างๆ ของโปรแกรม ในกรณีประกาศฟังก์ชัน main() ที่ไม่มีการคืนค่าใดๆ (void) ดังรูป



เมื่อทดสอบการทำงานของโปรแกรมจะได้ผลลัพธ์ดังนี้



อธิบายการทำงานของโปรแกรม

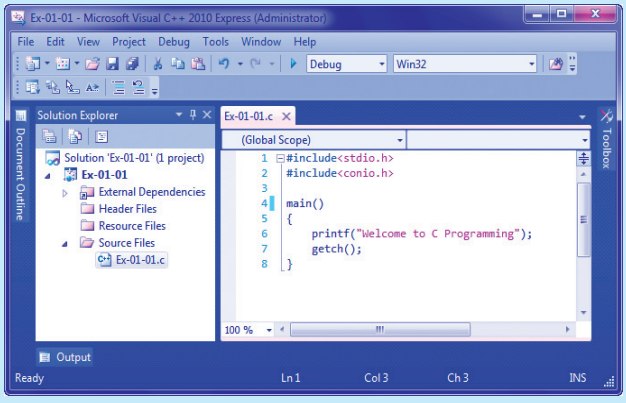
บรรทัดที่ 1–2	เป็นการเรียกใช้งานส่วนของเฮดเดอร์ไฟล์ สังเกตได้ที่เครื่องหมาย # โดยมีการเรียกใช้ไลบรารี stdio.h ซึ่งเป็นไลบรารีจัดการเกี่ยวกับอินพุตและเอาต์พุต และไลบรารี conio.h ซึ่งเป็นไลบรารีจัดการเกี่ยวกับหน้าจอทั้งหมด
บรรทัดที่ 4	ส่วนของฟังก์ชัน main() โดยประกาศชนิดข้อมูลที่คืนค่ากลับเป็น void และค่าที่รับเข้ามาในฟังก์ชันเป็น void หมายถึง ฟังก์ชันนี้จะไม่มีการคืนค่าใดๆ กลับออกไป และไม่มีการรับค่าใดๆ เข้ามาในฟังก์ชัน
บรรทัดที่ 5	ส่วนเริ่มต้นของฟังก์ชัน main()
บรรทัดที่ 6	เป็นคำสั่งแสดงผลทางจอภาพ โดยให้พิมพ์คำว่า Welcome to C Programming
บรรทัดที่ 7	เป็นคำสั่งรับอักขระจากแป้นพิมพ์ ในที่นี้เพื่อกำหนดไม่ให้โปรแกรมปิดหน้าต่างผลลัพธ์เมื่อแสดงผลลัพธ์ที่ต้องการแล้ว
บรรทัดที่ 8	ส่วนสิ้นสุดของฟังก์ชัน main()

NOTE

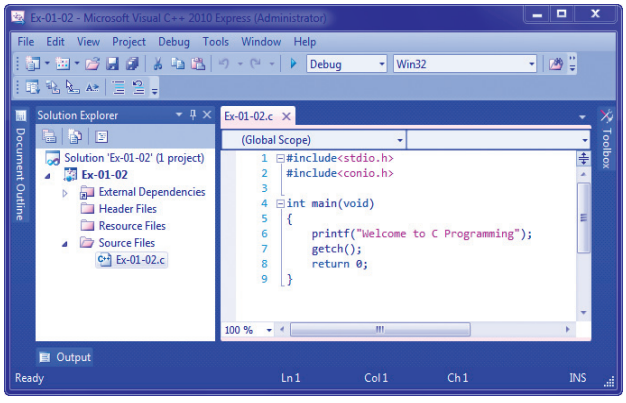
เมื่อผู้อ่านได้อ่านมาถึงจุดนี้ ผู้อ่านอาจจะไม่เข้าใจความหมายของคำบางคำ เช่น void, return type เป็นต้น แต่ผู้อ่านไม่ต้องเป็นกังวล เนื่องจากส่วนนี้ผู้เขียนมีเป้าหมายให้ผู้อ่านมองเห็นภาพรวมของการเขียนโปรแกรมภาษา C เท่านั้น โดยในส่วนต่างๆ ที่ผู้อ่านเป็นกังวล ผู้เขียนจะขออธิบายในบทต่อๆ ไป

TIPS

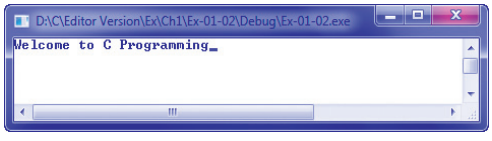
ในกรณีประกาศฟังก์ชัน main() โดยไม่ระบุค่าส่งกลับจะมีค่า Default ของค่าส่งกลับเป็น integer ดังรูป



ตัวอย่างที่ 1.2 เป็นตัวอย่างการเขียนโปรแกรมภาษา C อย่างง่าย เพื่อแสดงส่วนต่างๆ ของโปรแกรม กรณีประกาศฟังก์ชัน main() ที่มีการส่งค่าข้อมูลกลับดังรูป



เมื่อทดสอบการทำงานของโปรแกรมจะได้ผลลัพธ์ดังนี้



อธิบายการทำงานของโปรแกรม

บรรทัดที่ 1–2	เป็นการเรียกใช้งานส่วนของเฮดเดอร์ไฟล์ สังเกตได้ที่เครื่องหมาย # โดยมีการเรียกใช้ไลบรารี stdio.h ซึ่งเป็นไลบรารีจัดการเกี่ยวกับอินพุตและเอาต์พุต และไลบรารี conio.h ซึ่งเป็นไลบรารีจัดการเกี่ยวกับหน้าจอทั้งหมด
บรรทัดที่ 4	ส่วนของฟังก์ชัน main() โดยประกาศชนิดข้อมูลที่คืนค่ากลับเป็น int และค่าที่รับเข้ามาในฟังก์ชันเป็น void หมายถึง ฟังก์ชันนี้จะมีการส่งค่าข้อมูลกลับออกไป แต่ไม่มีการรับค่าใดๆ เข้ามาในฟังก์ชัน
บรรทัดที่ 5	ส่วนเริ่มต้นของฟังก์ชัน main()
บรรทัดที่ 6	เป็นคำสั่งแสดงผลทางจอภาพ โดยให้พิมพ์คำว่า Welcome to C Programming
บรรทัดที่ 7	เป็นคำสั่งรับอักขระจากแป้นพิมพ์ ในที่นี้เพื่อกำหนดไม่ให้โปรแกรมปิดหน้าต่างผลลัพธ์เมื่อแสดงผลลัพธ์ที่ต้องการแล้ว
บรรทัดที่ 8	เป็นส่วนของการส่งค่ากลับ ในที่นี้ให้ส่งค่ากลับเป็น 0 ซึ่งเป็นค่าตัวเลข (ซึ่งตรงกับชนิดข้อมูลที่ประกาศไว้ในบรรทัดที่ 4)
บรรทัดที่ 9	ส่วนสิ้นสุดของฟังก์ชัน main()

ความรู้พื้นฐานการพัฒนาซอฟต์แวร์

ในหัวข้อที่ผ่านมาผู้อ่านคงได้รู้จักกับภาษา C กันไปบ้างแล้ว ซึ่งเป็นพื้นฐานที่ทำให้ผู้อ่านได้มองเห็นภาพรวมการเขียนโปรแกรมด้วยภาษา C แต่ก่อนที่เราจะเริ่มเขียนโปรแกรมกัน ผู้เขียนอยากให้ผู้อ่านได้เรียนรู้ขั้นตอนการพัฒนาโปรแกรมหรือซอฟต์แวร์กันเสียก่อน เนื่องจากเราเรียนภาษา C เพื่อนำไปพัฒนาโปรแกรมหรือซอฟต์แวร์ในอนาคต

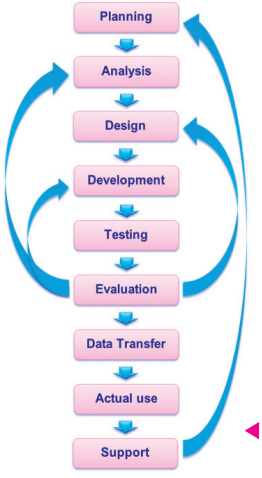
วงจรชีวิตของการพัฒนาซอฟต์แวร์ (Software Development Life Cycle-SDLC)

ในตอนนี้ผู้อ่านจะได้เรียนรู้เกี่ยวกับ**วงจรชีวิตของการพัฒนาซอฟต์แวร์** (Software Development Life Cycle) หรือเรียกสั้นๆ ว่า SDLC ซึ่งเป็นแนวทางหรือขั้นตอนการพัฒนาโปรแกรม ซอฟต์แวร์ หรือระบบสารสนเทศให้สำเร็จ โดยแนวทางหรือขั้นตอนการพัฒนาซอฟต์แวร์แต่ละซอฟต์แวร์อาจมีรูปแบบหรือรายละเอียดในบางขั้นตอนแตกต่างกัน ขึ้นอยู่กับทีมงานพัฒนาว่าต้องการใช้รูปแบบการพัฒนารูปแบบใด ที่คิดว่าเหมาะสมกับการพัฒนาระบบที่ต้องการมากที่สุด

ดังนั้น ลำดับวงจรชีวิตของการพัฒนาซอฟต์แวร์แต่ละตัวอาจจะแตกต่างกันได้ แต่สามารถสรุปเป็นขั้นตอนพอให้เข้าใจได้ดังนี้

1. **วางแผนงาน** (Planning) เป็นเสมือนจุดเริ่มต้นการพัฒนาโปรแกรมหรือระบบงาน เพื่อประมาณการต้นทุนในการพัฒนาซอฟต์แวร์ กำหนดระยะเวลา กำหนดรูปแบบและแนวทางของการพัฒนาซอฟต์แวร์คร่าวๆ ทำให้เราทราบขอบเขตและข้อกำหนดของการพัฒนาในเบื้องต้น
2. **วิเคราะห์ความต้องการของระบบ** (Analysis) ขั้นตอนนี้เป็นการค้นหาความต้องการของซอฟต์แวร์ และวิเคราะห์ความจำเป็นหรือขั้นตอนการทำงานของความต้องการนั้น
3. **ออกแบบระบบ** (Design) เป็นการออกแบบส่วนประกอบต่างๆ ของซอฟต์แวร์ให้ตรงกับความต้องการที่ได้วิเคราะห์มาแล้ว

4. **เขียนโปรแกรม** (Development) เป็นขั้นตอนการเขียนโปรแกรมตามแนวทางที่ได้ออกแบบไว้ให้โปรแกรมสามารถทำงานได้ตามต้องการ
5. **ทดสอบการทำงานของระบบ** (Testing) ขั้นตอนนี้เป็นการนำระบบหรือซอฟต์แวร์ที่ทำเสร็จแล้วมาทดสอบการใช้งานว่าทำงานถูกต้องตามความต้องการหรือไม่ การทดสอบนี้จะรวมถึงการทดสอบการเชื่อมโยงกับระบบซอฟต์แวร์อื่นๆ ที่เกี่ยวข้องด้วย เช่น การทำงานร่วมกับเครื่องพิมพ์ในเครือข่าย เป็นต้น
6. **ประเมินการทำงานของระบบ** (Evaluation) เมื่อทดสอบการทำงานของระบบแล้ว เราจะมีการประเมินว่าระบบหรือซอฟต์แวร์ที่ทดสอบสามารถทำงานได้ถูกต้องหรือไม่ มีข้อผิดพลาดประการใด เหมาะสมที่จะนำไปใช้งานได้หรือไม่ ในกรณีที่พบข้อผิดพลาดให้กลับมาแก้ไขข้อผิดพลาดที่พบ อย่างไรก็ตามการแก้ไขดังกล่าวจะแก้ไขที่ขั้นตอนใด จะขึ้นอยู่กับรายละเอียดของข้อผิดพลาดที่เกิดขึ้น เช่น หากข้อผิดพลาดที่พบเจอเป็นการเขียนโปรแกรมผิดพลาด ให้แก้ไขที่โค้ดโปรแกรม แต่หากเกิดจากการออกแบบระบบผิดพลาด ให้กลับไปแก้ไขที่ขั้นตอนการออกแบบระบบ เป็นต้น (ในรูปแบบการพัฒนาซอฟต์แวร์บางประเภท ขั้นตอนนี้อาจรวมอยู่ในการทดสอบการทำงานของระบบ)
7. **โอนย้ายข้อมูล** (Data Transfer) ขั้นตอนนี้เป็นการนำข้อมูลของระบบเก่ามาเข้าสู่ระบบใหม่ก่อนนำไปใช้จริง ในขั้นตอนนี้จะทำก็ต่อเมื่อทดสอบการทำงานและประเมินการทำงานของระบบเรียบร้อยแล้ว (ขั้นตอนนี้ อาจจะไม่มีความจำเป็นสำหรับการพัฒนาซอฟต์แวร์บางประเภท)
8. **นำไปใช้งานจริง** (Actual use) เมื่อซอฟต์แวร์พัฒนาสำเร็จและผ่านการทดสอบแล้ว ก็สามารถนำไปใช้งานจริงได้ ซึ่งขั้นตอนนี้อาจรวมไปถึงการแนะนำวิธีการใช้งานแก่ผู้ใช้ด้วย
9. **การให้ความช่วยเหลือ** (Support) ในขั้นตอนนี้เป็นการให้ความช่วยเหลือต่อผู้ใช้ เมื่อพบปัญหาในการใช้งานระบบ แต่ในกรณีที่ไม่สามารถแก้ไขปัญหาได้หรือจะต้องพัฒนาระบบเพิ่มเติมก็จะกลับไปทำตามขั้นตอนการวางแผนงานใหม่



◀ วงจรชีวิตของการพัฒนาซอฟต์แวร์ (Software Development Life Cycle-SDLC)

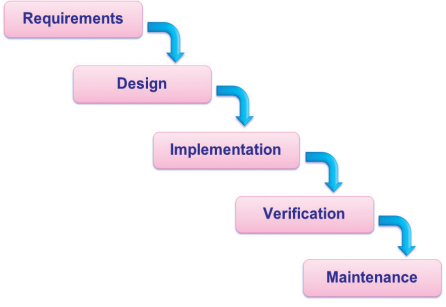
จากรูปจะเห็นได้ว่า ภาษา C ที่เราจะเรียนกันนั้นนำไปใช้ในขั้นตอนการเขียนโปรแกรม (Development) นั่นเอง ซึ่งถือเป็นหัวใจของการพัฒนาโปรแกรมหรือซอฟต์แวร์

ระเบียบวิธีการพัฒนาซอฟต์แวร์

จากวงจรชีวิตของการพัฒนาซอฟต์แวร์ ทำให้ปัจจุบันมีผู้นำเสนอระเบียบวิธีการพัฒนาซอฟต์แวร์มากมายหลายรูปแบบ แต่ละรูปแบบมีข้อดีและข้อเสียที่แตกต่างกัน ในที่นี้จะขอยกตัวอย่างเป็นกรณีศึกษาให้ผู้อ่านคือ โครงสร้างแบบน้ำตก (Waterfall Model) และโครงสร้างแบบก้นหอย (Spiral Model) เป็นต้น

โครงสร้างแบบน้ำตก (Waterfall Model)

ระเบียบวิธีการพัฒนาซอฟต์แวร์โครงสร้างแบบน้ำตกนี้ จะแบ่งกระบวนการทำงานออกเป็นขั้นตอนต่างๆ แต่ละขั้นตอนจะสืบเนื่องกันไปจากขั้นหนึ่งสู่อีกขั้นหนึ่งตามลำดับชั้นเหมือนสายน้ำตก โดยจะเริ่มต้นจากการกำหนดและวิเคราะห์ความต้องการของผู้ใช้ (Requirements) ก้าวไปสู่การสร้างแบบจำลอง และออกแบบวิธีการแก้ปัญหา (Design) ต่อด้วยการสร้างซอฟต์แวร์ (Implementation) จากนั้นก็ทดสอบการทำงานและติดตั้งซอฟต์แวร์ (Verification) และขั้นตอนสุดท้ายคือ การบำรุงรักษาให้ความช่วยเหลือแก่ผู้ใช้ซอฟต์แวร์ (Maintenance) ซึ่งสามารถแสดงได้ดังรูป



▲ ระเบียบวิธีการพัฒนาซอฟต์แวร์แบบ Waterfall Model

โครงสร้างแบบน้ำตกจะทำงานให้เสร็จทีละขั้น และจะไม่มีการย้อนกลับมาทำขั้นตอนก่อนหน้าอีก ดังนั้น จึงเหมาะสมกับระบบที่มีความต้องการที่ชัดเจน ซึ่งมีข้อดีและข้อเสียดังนี้

ข้อดี

- แบ่งงานง่ายให้เป็นงานเล็กๆ เพื่อให้ง่ายต่อการจัดการ
- มีการกำหนดซอฟต์แวร์ที่ต้องส่งมอบในแต่ละงานอย่างชัดเจน

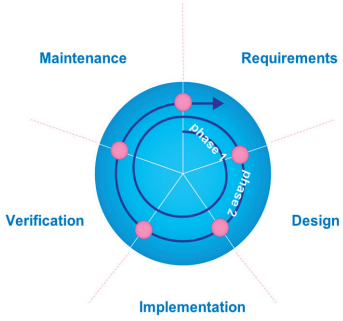
ข้อเสีย

- ถ้าหากพบข้อผิดพลาดของขั้นที่เสร็จสิ้นแล้ว จะไม่สามารถแก้ไขได้ การแก้ไขจำเป็นต้องเริ่มรอบ (Iteration) ใหม่ ซึ่งอาจจะเกิดความเสียหายอย่างใหญ่หลวง
- ผู้ใช้งานจะได้ใช้ซอฟต์แวร์ก็ต่อเมื่อถึงปลายทางของโครงการเท่านั้น

โครงสร้างแบบก้นหอย (Spiral Model)

โครงสร้างแบบก้นหอย (Spiral Model) เป็นการสร้างต้นแบบกับการทำงานอย่างเป็นขั้นตอนเช่นเดียวกับโครงสร้างแบบน้ำตก แต่จะเป็นลักษณะวนซ้ำการทำงานไปเรื่อยๆ จนพัฒนาซอฟต์แวร์เสร็จสมบูรณ์ ในหนึ่งวงรอบการทำงานจะเรียกว่า phase โดยแต่ละ phase จะมีการวิเคราะห์และจัดการความเสี่ยง เพื่อลดข้อผิดพลาดจากการพัฒนาซอฟต์แวร์ด้วย

โครงสร้างแบบก้นหอยจะมีพัฒนาการของซอฟต์แวร์เป็นเวอร์ชัน โดยเวอร์ชันต่อๆ ไปจะมีความสมบูรณ์มากยิ่งขึ้นในช่วง phase หลังๆ ซึ่งสามารถแสดงได้ดังรูป



▲ ระเบียบวิธีการพัฒนาซอฟต์แวร์แบบ Spiral Model

ข้อดี

- เนื่องจากการพัฒนาแบ่งออกเป็นช่วงๆ (phase) เมื่อพบข้อผิดพลาดจึงสามารถแก้ไขได้ทันท่วงที
- เป็น Model ที่เหมาะกับงานที่มีโอกาสเปลี่ยนแปลงบ่อย หรือมีความต้องการใหม่เรื่อยๆ มีลักษณะเป็นเวอร์ชัน

ข้อเสีย

- เป็น Model ที่เหมาะกับซอฟต์แวร์ขนาดใหญ่ เนื่องจากการวิเคราะห์และจัดการความเสี่ยงเป็นค่าใช้จ่าย ซึ่งอาจจะไม่คุ้มสำหรับโครงการขนาดเล็ก และผู้ที่จะจัดการความเสี่ยงได้ต้องมีประสบการณ์

สรุปเนื้อหาบทที่ 1

ภาษา C ได้ถูกคิดค้นโดย Dennis Ritchie ในปี ค.ศ. 1972 (พ.ศ. 2515) ซึ่งมีการพัฒนามาจากภาษา ALGOL, ภาษา BCPL และภาษา B โดยมีจุดเด่นอยู่ตรงที่

- มีแนวคิดในการพัฒนาโดยอาศัยหลักการ “โปรแกรมเชิงโครงสร้าง (Structure Programming)”
- การทำงานไม่ขึ้นกับฮาร์ดแวร์ สามารถนำไปใช้ใน CPU รุ่นต่างกันได้
- สามารถทำงานแทนภาษา Assembly ได้
- ความสามารถของคอมไพเลอร์ทำงานได้รวดเร็ว โดยใช้รหัสออบเจ็กต์ (Object) ที่สั้น
- โครงสร้างของภาษา C จะประกอบด้วยส่วนหลักๆ ดังนี้
  - ส่วนของเฮดเดอร์ไฟล์ (Header File or Processing Directive)
  - ส่วนของตัวแปร Global
  - ส่วนของฟังก์ชัน (Function)
  - ส่วนของตัวแปร Local
  - ส่วนของตัวโปรแกรม
  - ส่วนของตัวส่งค่ากลับ

หลักการพัฒนาซอฟต์แวร์ หรือวงจรชีวิตของการพัฒนาซอฟต์แวร์ (Software Development Life Cycle–SDLC) ของซอฟต์แวร์แต่ละประเภทมีความแตกต่างกัน ควรเลือกใช้งานรูปแบบการพัฒนาซอฟต์แวร์ให้เหมาะสม

คู่มือเรียนเขียนโปรแกรม

ภาษา C

ฉบับสมบูรณ์

หนังสือเล่มนี้แนะนำให้ผู้อ่านรู้จักการเขียนโปรแกรมภาษา C ตั้งแต่พื้นฐานจนสามารถนำไปใช้แก้ไขโจทย์ปัญหาได้จริง โดยเนื้อหาจะสอนตั้งแต่การติดตั้งโปรแกรม, หลักการเขียนโปรแกรมของภาษา C, การเขียน Flowchart, การใช้งาน Array, Pointer, struct, union, enum และ Linked List, การทำงานร่วมกับไฟล์ และการสร้างและใช้งานฟังก์ชัน ซึ่งมีตัวอย่างประกอบทุกหัวข้อ พร้อมคำอธิบายอย่างละเอียด เหมาะสำหรับนักเรียน นักศึกษา และผู้ที่สนใจทั่วไป

เนื้อหาประกอบด้วย

- ติดตั้งโปรแกรมสำหรับเขียนโปรแกรมภาษา C
- หลักการเขียนโปรแกรมภาษา C
- รู้จักกับชนิดข้อมูล ตัวแปร และค่าคงที่ (Data Type, Variable and Constants)
- รู้จักและใช้งานตัวดำเนินการ (Operator)
- เรียนรู้การแสดงผลและการรับข้อมูล
- เรียนรู้คำสั่งตัดสินใจเพื่อควบคุมทิศทางการทำงานของโปรแกรม (Decision Control Statement)
- เรียนรู้คำสั่งวนซ้ำเพื่อควบคุมทิศทางการทำงานของโปรแกรม (Repetition Control Statement)
- เรียนรู้การออกแบบการทำงานของโปรแกรมด้วย Flowchart
- รู้จักและใช้งานอาร์เรย์ (Array)
- รู้จักและใช้งานพอยน์เตอร์ (Pointer)
- เรียนรู้การสร้างและใช้งานฟังก์ชัน (Function)
- รู้จักการใช้งานโครงสร้างข้อมูล (Structuring Data) struct, union, enum และ Linked List
- แนะนำการทำงานร่วมกับไฟล์
- รู้จักกับ Preprocessor Directive
- รู้จักกับ Command line argument

ผู้แต่ง : ไกรศร ตั้งโอภากุล

สำเร็จการศึกษาระดับปริญญาตรี สาขาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
มีประสบการณ์ในตำแหน่งงานโปรแกรมเมอร์มามากกว่า 5 ปี

มีผลงานเขียนที่ได้รับความนิยม อาทิ
คู่มือเรียนเขียนโปรแกรมภาษา C ISBN : 978-616-200-126-0

ผู้แต่ง : กิตินันท์ พลสวัสดิ์

สำเร็จการศึกษาระดับปริญญาโท สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
มีประสบการณ์พัฒนาระบบงานมามากกว่า 5 ปี เคยเป็นวิทยากรด้านการเขียนโปรแกรมภาษา Pascal, C, Java และ VB
ปัจจุบันเป็นบรรณาธิการของ DEV BOOK

มีผลงานเขียนที่ได้รับความนิยมมากมาย อาทิ
คู่มือเรียนและใช้งาน Visual C# 2010 ฉบับสมบูรณ์ ISBN : 978-616-200-207-6
คู่มือเรียนเขียนโปรแกรมภาษา C ISBN : 978-616-200-126-0
รวมโจทย์และแบบฝึกหัดภาษา C + Java ISBN : 978-616-200-125-3

DEV BOOK

จัดจำหน่ายโดย IDC PREMIER
ISBN 885-916-100-439-4
8 859161 004394 250 บาท