

```
{  
    for (i=0; i<pbtrec->length; i++)  
        pbtrec->data[i+pbtrec->start]=ENCODE1(pbtrec->data[i],pbtrec->key);  
    else if (pbtrec->funcnum==2)  
    {  
        for (i=0; i<pbtrec->length; i++)  
            pbtrec->data[i+pbtrec->start]=ENCODE2(pbtrec->data[i],pbtrec->key);  
    }  
    else if (pbtrec->funcnum==3 || pbtrec->funcnum != 0)
```

เอกสารประกอบการเรียน



เริ่มต้นภาษาซี

มาตรฐานการเรียนรู้

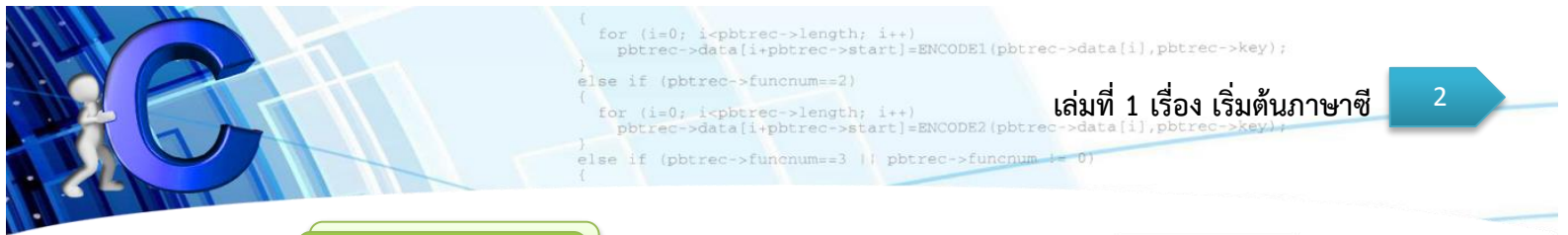
มาตรฐาน ง 3.1 เข้าใจ เห็นคุณค่าและใช้กระบวนการเทคโนโลยีสารสนเทศในการสืบค้นข้อมูล การเรียนรู้ การสื่อสาร การแก้ปัญหา การทำงาน และอาชีพอย่างมีประสิทธิภาพ ประสิทธิผล มีคุณธรรม

ตัวชี้วัด

1. ขั้นตอนการพัฒนาโปรแกรม มี 5 ขั้นตอน ได้แก่ การวิเคราะห์ปัญหา การออกแบบโปรแกรม การเขียนโปรแกรม การทดสอบโปรแกรม และการจัดทำเอกสารประกอบ
2. การเขียนโปรแกรม เช่น ซี จาวา ปาสคาล วิซวลเบสิก ซีชาร์ป
3. การเขียนโปรแกรมในงานด้านต่างๆ เช่น การจัดการข้อมูล การวิเคราะห์ข้อมูล การแก้ปัญหา ในวิชาคณิตศาสตร์ และวิทยาศาสตร์ การสร้างชิ้นงาน

สาระสำคัญ

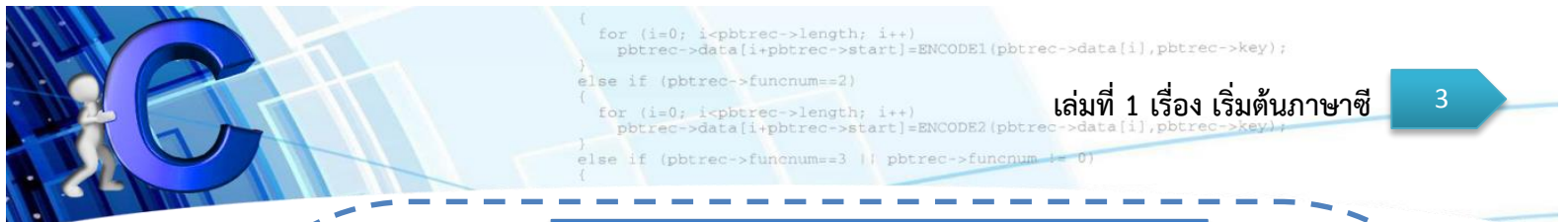
ภาษาซี ถูกสร้างขึ้นมาเพื่อช่วยให้การเขียนโปรแกรมง่ายขึ้น เป็นภาษาที่มีหลักการทำงานที่ไม่ซับซ้อน เข้าใจง่าย และในการเขียนโปรแกรมคอมพิวเตอร์ ผู้เขียนโปรแกรมจะต้องทราบถึงขั้นตอนวิธีการของการแก้ปัญหาของโปรแกรม ข้อมูลที่จะนำเข้า ผลลัพธ์ที่ต้องการ ตามขั้นตอนการพัฒนาโปรแกรมทั้ง 5 ขั้นตอน คือ การวิเคราะห์ปัญหา การออกแบบโปรแกรม การเขียนโปรแกรม การทดสอบโปรแกรม และการจัดทำเอกสารประกอบ



```
{  
    for (i=0; i<pbtre->length; i++)  
        pbtre->data[i+pbtre->start]=ENCODE1(pbtre->data[i],pbtre->key);  
    else if (pbtre->funcnum==2)  
    {  
        for (i=0; i<pbtre->length; i++)  
            pbtre->data[i+pbtre->start]=ENCODE2(pbtre->data[i],pbtre->key);  
    }  
    else if (pbtre->funcnum==3 || pbtre->funcnum != 0)  
    {
```

ผลการเรียนรู้

1. นักเรียนอธิบายขั้นตอนการพัฒนาโปรแกรมได้
2. นักเรียนบอกประวัติความเป็นมาของภาษาซีได้
3. นักเรียนบอกลักษณะโครงสร้างภาษาซีได้
4. นักเรียนอธิบายลักษณะของตัวแปลภาษาแต่ละชนิดได้
5. นักเรียนเข้าใจความรู้เบื้องต้นเกี่ยวกับการติดตั้งโปรแกรมภาษาซี
6. นักเรียนสามารถติดตั้งโปรแกรม Dev c ได้
7. นักเรียนเข้าใจความรู้เบื้องต้นเกี่ยวกับการใช้โปรแกรมภาษาซี
8. นักเรียนสามารถใช้งานโปรแกรม Dev c ได้



```
( for (i=0; i<pbtre->length; i++)
pbtre->data[i+pbtre->start]=ENCODE1(pbtre->data[i],pbtre->key);
else if (pbtre->funcnum==2)
{
for (i=0; i<pbtre->length; i++)
pbtre->data[i+pbtre->start]=ENCODE2(pbtre->data[i],pbtre->key);
}
else if (pbtre->funcnum==3 || pbtre->funcnum != 0)
```

แบบทดสอบก่อนเรียน เรื่อง เริ่มต้นภาษาซี

- คำชี้แจง
1. แบบทดสอบฉบับนี้มี 10 ข้อ ข้อละ 1 คะแนน คะแนนเต็ม 10 คะแนน เวลา 20 นาที
 2. แบบทดสอบนี้เป็นแบบเลือกตอบ
 3. ให้นักเรียนทำเครื่องหมาย X ลงในกระดาษคำตอบ โดยเลือกตัวอักษร ก ข ค และ ง ที่ถูกต้องที่สุดเพียงข้อเดียว

1. ภาษาซีสืบสานมาจากภาษาใด
 - ก. Pascal
 - ข. B
 - ค. Java
 - ง. HTML
2. ข้อใดคือผู้พัฒนาภาษาซี
 - ก. เดนิส ริชี
 - ข. ชาลส์ แบบเบจ
 - ค. เอดา เลิฟเลซ
 - ง. ทิม เบอร์เนิร์ส-ลี
3. ภาษาซีจัดเป็นภาษาระดับใด
 - ก. ระดับต่ำ
 - ข. ระดับกลาง
 - ค. ระดับสูง
 - ง. ภาษาเครื่อง
4. โปรแกรมภาษาซีจะเริ่มทำงานจากฟังก์ชันใด
 - ก. ฟังก์ชัน main
 - ข. ฟังก์ชัน include
 - ค. ฟังก์ชัน library
 - ง. ฟังก์ชัน declare
5. ข้อใดไม่ใช่ขั้นตอนการพัฒนาโปรแกรมโดยใช้ภาษา
 - ก. การเขียนและแก้ไขโปรแกรม
 - ข. คอมไพล์โปรแกรม
 - ค. การลิงค์โปรแกรม
 - ง. การประกาศตัวแปร

6. ข้อใดไม่ใช่ส่วนประกอบหลักของโครงสร้างภาษาซี
 - ก. ส่วนหัวของโปรแกรม (Preprocessor Directive)
 - ข. ส่วนฟังก์ชันหลัก (Main Function)
 - ค. ส่วนของคำอธิบายโปรแกรม (Program Comment)
 - ง. ส่วนคอมไพเลอร์ (Compiler)
7. ภาษาซีจัดเป็นภาษาที่ใช้ตัวแปลภาษาชนิดใด
 - ก. Compiler
 - ข. Translator
 - ค. Interpreter
 - ง. Assembler
8. เครื่องหมายใดต่อไปนี้ ที่นำมาใช้สำหรับอธิบายรายละเอียดภายในโปรแกรม
 - ก. @ @
 - ข. /* */
 - ค. []
 - ง. { }
9. ประโยคคำสั่งในภาษาซีต้องลงท้ายด้วยเครื่องหมายใด
 - ก. , (comma)
 - ข. \n
 - ค. ; (semi-colon)
 - ง. : (colon)
10. หากต้องการนำฟังก์ชันมาตรฐาน (stdio.h) มาใช้งาน จะต้องประกาศอย่างไร
 - ก. main(stdio.h)
 - ข. { stdio.h }
 - ค. printf("<stdio.h>")
 - ง. #include "stdio.h"

ทำแบบทดสอบก่อนเรียนกันแล้ว
ศึกษาใบความรู้ที่ 1 กันต่อเลยคะ



ใบความรู้ที่ 1

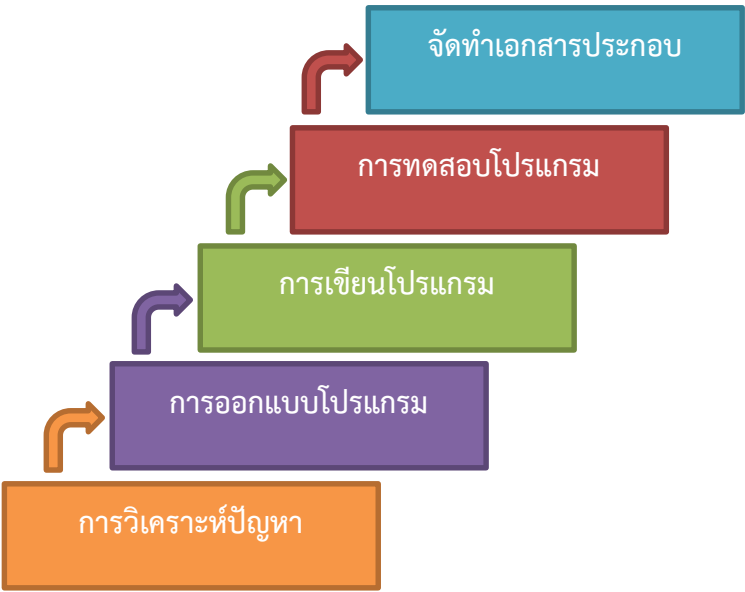
เรื่อง ขั้นตอนการพัฒนาโปรแกรม

C

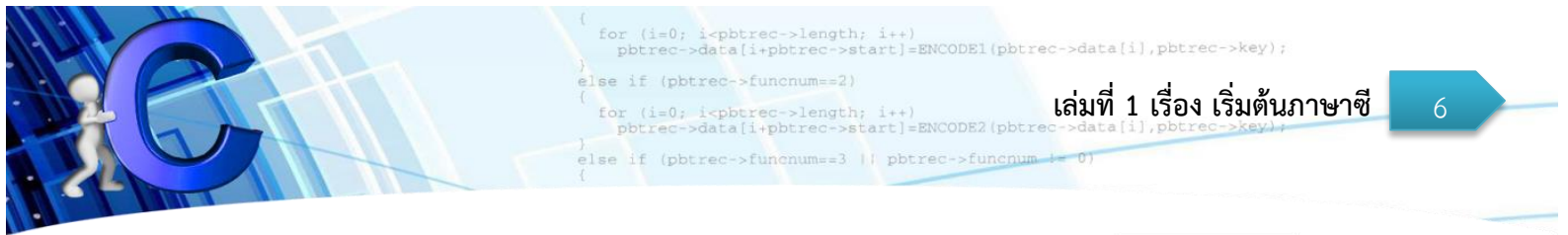
ขั้นตอนการพัฒนาโปรแกรม

ในการเขียนโปรแกรมคอมพิวเตอร์เพื่อแก้ปัญหาตรงกับความต้องการ ผู้เขียนโปรแกรมจะต้องทราบจุดหมายของการเขียนโปรแกรม เช่น ข้อมูลที่จะต้องนำเข้า ลักษณะการทำงานของโปรแกรม รูปแบบการแสดงผลของผลลัพธ์ ซึ่งหากมีการวางแผนจะทำให้การเขียนโปรแกรมเป็นไปตามลำดับขั้นตอนจะทำให้ลดการซ้ำซ้อนของขั้นตอนการเขียนโปรแกรมและไม่ทำให้เสียเวลามาก ขั้นตอนการพัฒนาโปรแกรมจะช่วยให้การเขียนโปรแกรมเป็นไปอย่างเป็นระบบ ช่วยให้ง่ายต่อการเขียนโปรแกรมในการแก้ปัญหา โดยทั่วไปแล้วขั้นตอนการพัฒนาโปรแกรมจะประกอบด้วย 5 ขั้นตอนดังนี้

1. การวิเคราะห์ปัญหา
2. การออกแบบโปรแกรม
3. การเขียนโปรแกรม
4. การทดสอบโปรแกรม
5. จัดทำเอกสารประกอบ



ภาพที่ 1.1 แสดงขั้นตอนการพัฒนาโปรแกรม



```
{ for (i=0; i<pbtrec->length; i++)  
  pbtrec->data[i+pbtrec->start]=ENCODE1(pbtrec->data[i],pbtrec->key);  
} else if (pbtrec->funcnum==2)  
{ for (i=0; i<pbtrec->length; i++)  
  pbtrec->data[i+pbtrec->start]=ENCODE2(pbtrec->data[i],pbtrec->key);  
} else if (pbtrec->funcnum==3 || pbtrec->funcnum != 0)  
{
```

1. การวิเคราะห์ปัญหา

เป็นขั้นตอนแรกที่คุณเขียนโปรแกรมจะต้องทำ เพื่อการทำงานเป็นไปอย่างมีประสิทธิภาพ ผู้เขียนโปรแกรมจะต้องวิเคราะห์ปัญหาให้ทราบถึงความต้องการให้โปรแกรมแก้ปัญหาอะไร ทำงานอย่างไร ข้อมูลนำเข้ามีอะไรบ้าง โดยมีขั้นตอนย่อยดังนี้

- 1) กำหนดขอบเขตของปัญหา โดยการกำหนดรายละเอียดให้ชัดเจนว่าจะให้โปรแกรมประมวลผลสิ่งใดบ้าง ตัวแปร ค่าคงที่ที่ต้องใช้เป็นลักษณะใด
- 2) กำหนดข้อมูลเข้าและผลลัพธ์ โดยต้องรู้ว่าข้อมูลที่จะส่งเข้าเป็นอย่างไร มีอะไรบ้าง เพื่อให้โปรแกรมประมวลผลและแสดงผลลัพธ์
- 3) กำหนดวิธีการประมวลผล โดยจะต้องรู้ว่าจะให้คอมพิวเตอร์ประมวลผลอย่างไร จึงได้ผลลัพธ์ตามต้องการ

2. การออกแบบโปรแกรม

ขั้นตอนนี้จะใช้เครื่องมือช่วยในการออกแบบโปรแกรม โดยเขียนเป็นลำดับขั้นตอนการทำงานของโปรแกรมที่เรียกว่า อัลกอริทึม (algorithm) ซึ่งจะแสดงขั้นตอนการแก้ปัญหา ใช้ประโยคที่ชัดเจน ไม่คลุมเครือ มีรายละเอียดเพียงพอที่จะนำไปเขียนโปรแกรมให้ทำงานจริง อัลกอริทึมนั้นอาจเขียนให้อยู่ในรูปของรหัสจำลองหรือซูโดโค้ด (pseudo code) หรือเขียนเป็นผังงาน (flowchart) ก็ได้ โดยซูโดโค้ดจะเป็นคำอธิบายขั้นตอนการทำงานของโปรแกรมเป็นคำย่อส่วนผังงานใช้สัญลักษณ์ต่างๆ แทนการทำงานและทิศทางของโปรแกรม

3. การเขียนโปรแกรม

ขั้นตอนนี้จะเขียนโปรแกรมเพื่อให้คอมพิวเตอร์สามารถประมวลผลได้ โดยเปลี่ยนขั้นตอนการทำงานให้อยู่ในรูปรหัสภาษาคอมพิวเตอร์ การเขียนโปรแกรมจะต้องเขียนตามภาษาที่คอมพิวเตอร์เข้าใจโดยอาจใช้ภาษาระดับสูง หรือระดับต่ำ ซึ่งสามารถเลือกได้หลายภาษา

4. การทดสอบและแก้ไขโปรแกรม

หลังจากเขียนโปรแกรมจะต้องทดสอบความถูกต้องของโปรแกรมที่เขียนขึ้น หากจุดผิดพลาดของโปรแกรม และตรวจสอบจนไม่พบจุดผิดพลาดอีก จุดผิดพลาดของโปรแกรมนี้นี้เรียกว่า บั๊ก (bug) ส่วนการแก้ไขข้อผิดพลาดให้ถูกต้องเรียกว่า ดีบั๊ก (debug) โดยทั่วไปแล้วข้อผิดพลาดจากการเขียนโปรแกรมจะมี 2 ประเภท คือ

- 1) การเขียนคำสั่งไม่ถูกต้องตามหลักการเขียนโปรแกรมนั้นๆ ซึ่งเรียกว่า syntax error หรือ coding error ข้อผิดพลาดประเภทนี้เรามักพบตอนแปลภาษาโปรแกรมเป็นรหัสภาษาเครื่อง
- 2) ข้อผิดพลาดทางตรรกะ หรือ logic error เป็นข้อผิดพลาดที่โปรแกรมทำงานได้ แต่ผลลัพธ์ออกมาไม่ถูกต้อง

5. ทำเอกสารและบำรุงรักษาโปรแกรม

ขั้นตอนนี้จะทำให้ผู้ใช้สามารถใช้งานโปรแกรมได้อย่างมีประสิทธิภาพและสะดวกในการตรวจสอบข้อผิดพลาดโดยเขียนเป็นเอกสารประกอบขึ้นมา



แบบฝึกหัดที่ 1.1

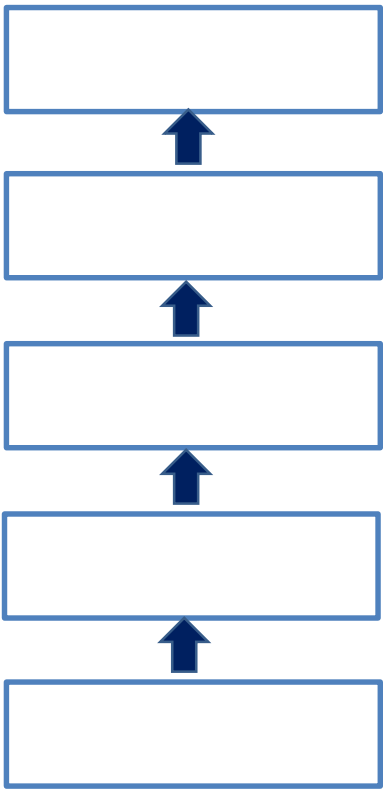
เรื่อง ขั้นตอนการพัฒนาโปรแกรม

จุดประสงค์การเรียนรู้

1. นักเรียนอธิบายขั้นตอนการพัฒนาโปรแกรมได้

คำชี้แจง

ให้นักเรียนเขียนขั้นตอนการพัฒนาโปรแกรมลงในกรอบสี่เหลี่ยมโดยเรียงตามลำดับขั้นตอนให้ถูกต้อง จากตัวเลือกที่กำหนดให้



ใบความรู้ที่ 2

เรื่อง ประวัติภาษาซี

C

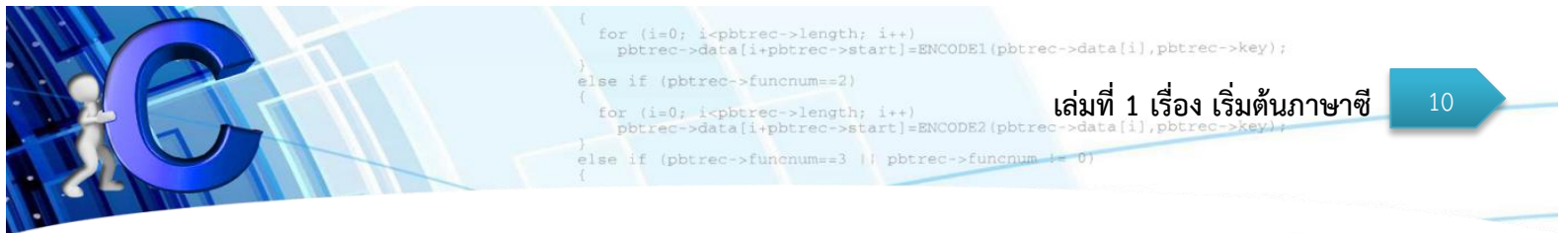
ประวัติภาษาซี



เดนนิส ริตชี (Dennis Ritchie)

ภาษาซีเป็นภาษาที่ถือว่าเป็นภาษาระดับสูง ถูกพัฒนาโดย เดนนิส ริตชี (Dennis Ritchie) แห่งห้องทดลองเบลล์ (Bell Laboratories) ที่เมอร์ริลล์ มลรัฐนิวเจอร์ซีย์ ในปี ค.ศ. 1972 โดยเดนนิสได้ใช้หลักการของภาษา B ซึ่งพัฒนาขึ้นโดย เคน ทอมสัน (Ken Tomson) ตั้งอยู่บนพื้นฐานของภาษา BCPL ที่ถูกพัฒนาโดย Martin Richards การออกแบบและพัฒนาภาษาซีของ เดนนิส ริตชี มีจุดมุ่งหมายให้เป็นภาษาสำหรับใช้เขียนโปรแกรมปฏิบัติการระบบยูนิกซ์ และได้ตั้งชื่อว่า ซี (C) เพราะเห็นว่า ซี (C) เป็นตัวอักษรต่อจากบี (B) ของภาษา BCPL

ในปี ค.ศ. 1978 เดนนิส ริตชี และนายเบรน เคนิกฮาน (Brian W. Kernighan) ได้แต่งหนังสือชื่อ “The C Programming Language” โดยสามารถนำภาษา C ที่สามารถนำมาปรับใช้กับคอมพิวเตอร์ในรูปแบบต่างๆได้มากขึ้น ในปี 1988 ได้มีการสร้างมาตรฐานของภาษา C โดย ANSI C และในปี 1990 องค์กรมาตรฐานสากล (ISO) ได้ยอมรับมาตรฐานที่ได้สร้างขึ้น ภายใต้ชื่อ ANSI/ISO C



```
{ for (i=0; i<pbtrec->length; i++)  
  pbtrec->data[i+pbtrec->start]=ENCODE1(pbtrec->data[i],pbtrec->key);  
} else if (pbtrec->funcnum==2)  
{ for (i=0; i<pbtrec->length; i++)  
  pbtrec->data[i+pbtrec->start]=ENCODE2(pbtrec->data[i],pbtrec->key);  
} else if (pbtrec->funcnum==3 || pbtrec->funcnum != 0)
```



วิวัฒนาการของภาษาซี

- ค.ศ. 1970 มีการพัฒนาภาษา B โดย Ken Thompson ซึ่งทำงานบนเครื่อง DEC PDP-7 ซึ่ง ทำงานบนเครื่องไมโครคอมพิวเตอร์ไม่ได้ และยังมีข้อจำกัดในการใช้งานอยู่ (ภาษา B สืบทอดมาจาก ภาษา BCPL ซึ่งเขียนโดย Marth Richards)

- ค.ศ. 1972 Dennis M. Ritchie และ Ken Thompson ได้สร้างภาษา C เพื่อเพิ่มประสิทธิภาพ ภาษา B ให้ดียิ่งขึ้น ในระยะแรกภาษา C ไม่เป็นที่นิยมแก่นักโปรแกรมเมอร์โดยทั่วไปนัก

- ค.ศ. 1978 Brian W. Kernighan และ Dennis M. Ritchie ได้เขียนหนังสือเล่มหนึ่งชื่อว่า The C Programming Language และหนังสือเล่มนี้ทำให้บุคคลทั่วไปรู้จักและนิยมใช้ภาษา C ในการเขียน โปรแกรมมากขึ้น

- แต่เดิมภาษา C ใช้ Run บนเครื่องคอมพิวเตอร์ 8 bit ภายใต้ระบบปฏิบัติการ CP/M ของ IBM PC ซึ่งในช่วงปี ค. ศ. 1981 เป็นช่วงของการพัฒนาเครื่องไมโครคอมพิวเตอร์ ภาษา C จึงมีบทบาทสำคัญในการนำมาใช้บนเครื่อง PC ตั้งแต่นั้นเป็นต้นมา และมีการพัฒนาต่อมาอีกหลาย ๆ ค่าย ดังนั้นเพื่อกำหนดทิศทางการใช้ภาษา C ให้เป็นไปแนวทางเดียวกัน ANSI (American National Standard Institute) ได้กำหนดข้อตกลงที่เรียกว่า 3J11 เพื่อสร้างภาษา C มาตรฐานขึ้นมา เรียกว่า ANSI C

- ค.ศ. 1983 Bjarne Stroustrup แห่งห้องปฏิบัติการเบล (Bell Laboratories) ได้พัฒนา ภาษา C++ ขึ้นรายละเอียดและความสามารถของ C++ มีส่วนขยายเพิ่มจาก C ที่สำคัญ ๆ ได้แก่ แนวความคิดของการเขียนโปรแกรมแบบกำหนดวัตถุเป้าหมายหรือแบบ OOP (Object Oriented Programming) ซึ่งเป็นแนวการเขียนโปรแกรมที่เหมาะสมกับการพัฒนาโปรแกรมขนาดใหญ่ที่มีความสลับซับซ้อนมาก มีข้อมูลที่ใช้ในโปรแกรมจำนวนมาก จึงนิยมใช้เทคนิคของการเขียนโปรแกรมแบบ OOP ในการพัฒนาโปรแกรมขนาดใหญ่ในปัจจุบันนี้



ลักษณะเด่นของภาษาซี

1. ความสามารถในการใช้งานบนคอมพิวเตอร์ที่แตกต่างกันได้ คือทำงานบนคอมพิวเตอร์หลายระดับ เช่น เมนเฟรมไปจนถึงไมโครคอมพิวเตอร์ และยังสามารถนำภาษาซีมาใช้บนระบบปฏิบัติการที่หลากหลายได้

2. มีประสิทธิภาพในการทำงานสูง มีการทำงานที่รวดเร็ว การทำงานเหมือนภาษาระดับต่ำ การจัดการหน่วยความจำมีประสิทธิภาพสูง ชุดคำสั่งกระชับ

- ภาษาซีมีแนวคิดในการพัฒนาแบบ โปรแกรมเชิงโครงสร้าง จึงเหมาะสำหรับนำมาพัฒนาระบบ
- มีความยืดหยุ่นสูง สามารถเขียนใช้งานร่วมกับภาษาระดับต่ำอย่างภาษาแอสเซมบลีได้



แบบฝึกหัดที่ 1.2

เรื่อง ประวัติภาษาซี

จุดประสงค์การเรียนรู้

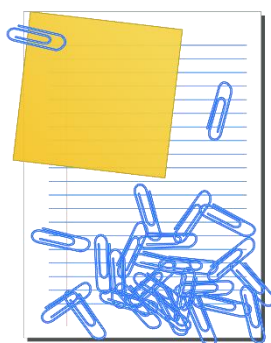
นักเรียนบอกประวัติความเป็นมาของภาษาซีได้

คำชี้แจง

ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้อง

1. ให้นักเรียนบอกประวัติความเป็นมาของภาษาซีมาโดยสังเขป

2. ให้นักเรียนบอกลักษณะเด่นของภาษาซี



ใบความรู้ที่ 3

เรื่อง โครงสร้างโปรแกรมภาษาซี

โครงสร้างโปรแกรมภาษาซี

ลักษณะโครงสร้างภาษาซีแบ่งออกได้เป็น 4 ส่วน ดังนี้

1. พรีโปรเซสเซอร์ไคเร็กทีฟ (preprocessor directives)
2. ส่วนฟังก์ชันหลัก (the main () function)
3. ประโยคคำสั่ง (compound statement)
4. ส่วนอธิบายโปรแกรม (program comment)

1. พรีโปรเซสเซอร์ไคเร็กทีฟ (preprocessor directives)

ส่วนนี้ถือเป็นส่วนสำคัญอย่างมากของภาษาซี จะใช้สำหรับเรียกไฟล์ที่โปรแกรมต้องการในการทำงาน และกำหนดค่าต่างๆ ซึ่งจะต้องเริ่มต้นด้วยเครื่องหมาย ไคเร็กทีฟ # และตามด้วยชื่อโปรแกรมหรือชื่อตัวแปรที่ต้องการกำหนดค่า ส่วนนี้อาจเรียกอีกชื่อหนึ่งว่า ส่วนหัวโปรแกรม สำหรับไคเร็กทีฟที่ใช้กันบ่อยๆ ได้แก่

#include เป็นการแจ้งให้คอมไพเลอร์อ่านไฟล์อื่นเข้ามาคอมไพล์ร่วมด้วย รูปแบบการใช้จะทำได้โดยการเขียน #include แล้วตามด้วยชื่อไฟล์ดังนี้

```
#include ชื่อไฟล์
```

```
#include <stdio.h>                   หมายความว่า อ่านไฟล์ stdio.h เข้ามาด้วย
```

การกำหนดชื่อไฟล์ตามหลัง #include นั้น อาจใช้เครื่องหมาย <> คร่อมไฟล์ก็ได้ ซึ่งจะเป็นการอ่านไฟล์ไคเร็กทอรีกำหนดไว้ก่อน แต่ถ้าใช้ “ ” เป็นการอ่านไฟล์ไคเร็กทอรีปัจจุบันที่กำลังติดต่อยู่ และไฟล์ที่จะ include มานี้จะต้องไม่มีฟังก์ชัน main() โดยมากแล้วจะประกอบด้วยโปรแกรมน้อยๆ ค่าคงที่ หรือข้อกำหนดต่าง เฮดเดอร์ไฟล์ชื่อ stdio.h ถือเป็นเฮดเดอร์ไฟล์ที่มักถูกเรียกใช้งานอยู่เสมอ เพราะเกี่ยวข้องกับอินพุตและเอาต์พุต เช่น ฟังก์ชัน printf() ฟังก์ชัน scanf() ซึ่งทั้งสองฟังก์ชัน

ถูกผนวกอยู่ในเฮดเดอร์ไฟล์ `stdio.h` และหากโปรแกรมมีการเรียกใช้งานสองฟังก์ชันนี้ แต่ไม่ได้ผนวกเฮดเดอร์ไฟล์ `stdio.h` เข้าไป เมื่อคอมไพล์โปรแกรมก็จะเกิดข้อผิดพลาด

#define เป็นการกำหนดค่านิพจน์ต่างๆให้ชื่อตรงกับตัวแปร โดยมีรูปแบบดังนี้

#define A 15;	กำหนดค่า A มีค่าเท่ากับ 15
#define B 2+4/2	กำหนด B มีค่าเป็น 2+4/2

ตัวอย่างพรีโปรเซสเซอร์ที่มีอยู่ในภาษาซี

#if	#ifdef	#ifndef	#else	#elif
#endif	#include	#define	#undef	##line
#error	#pragma			

2. ส่วนฟังก์ชันหลัก (the main () function)

ส่วนนี้ทำหน้าที่เหมือนกับเป็นโปรแกรมหลักที่สั่งให้ชุดคำสั่งทำงาน ซึ่งจะประกอบไปด้วยประโยคคำสั่งต่างๆ ที่จะให้โปรแกรมทำงาน โดยนำคำสั่งต่างๆมาเรียงต่อกัน และแต่ละประโยคคำสั่งจะจบด้วยเครื่องหมายเซมิโคลอน (Semi colon ;) โดยโปรแกรมหลักนี้จะเริ่มต้นด้วย `main()` ตามด้วยเครื่องหมายปีกกาเปิด { และจบด้วยเครื่องหมายปีกกาปิด }

3. ประโยคคำสั่ง (compound statement)

ประโยคคำสั่งในที่นี้หมายถึง ชุดคำสั่งที่บรรจุอยู่ในฟังก์ชันนั้นๆ ซึ่งอาจเป็น

- ประโยคที่ใช้สำหรับประกาศตัวแปร หรือการกำหนดค่าเริ่มต้นให้กับตัวแปรต่างๆ โดยตัวแปรที่ใช้งานในโปรแกรม จำเป็นต้องได้รับการประกาศชนิดข้อมูลของตัวแปรนั้นๆด้วย
- ประโยคนิพจน์คณิตศาสตร์ เช่น ประโยคคำนวณตัวเลขต่างๆ
- ประโยคคำสั่งควบคุมอื่นๆ เช่น คำสั่งควบคุมวงจรรูป คำสั่งควบคุมเงื่อนไข เป็นต้น

4. ส่วนอธิบายโปรแกรม (program comment)

คำอธิบายโปรแกรม เป็นส่วนที่ผู้เขียนโปรแกรมนำมาใช้อธิบายจุดสำคัญต่างๆ ภายในโปรแกรม เช่น ใช้อธิบายจุดประสงค์ของโปรแกรมส่วนนั้นๆ รวมถึงป้องกันการหลงลืม กรณีที่ต้องกลับมาปรับปรุงโปรแกรม ถือเป็นส่วนสำคัญอย่างหนึ่งที่ช่วยให้ผู้เขียนโปรแกรม หรือผู้ที่นำโปรแกรมนี้อไปใช้ได้เข้าใจจุดมุ่งหมายของโปรแกรมในส่วนนั้นๆ โดยคอมไพเลอร์จะไม่นำส่วนนี้มาประมวลผลรูปแบบการเขียนคำอธิบาย สามารถเขียนให้อยู่ในรูปแบบ `/* comment */` หรือ `//comment` ก็ได้

ตัวอย่างการเขียนโปรแกรมภาษาซี

```
#include "stdio.h"

int main()

{

}
```

←

ส่วนหัว

←

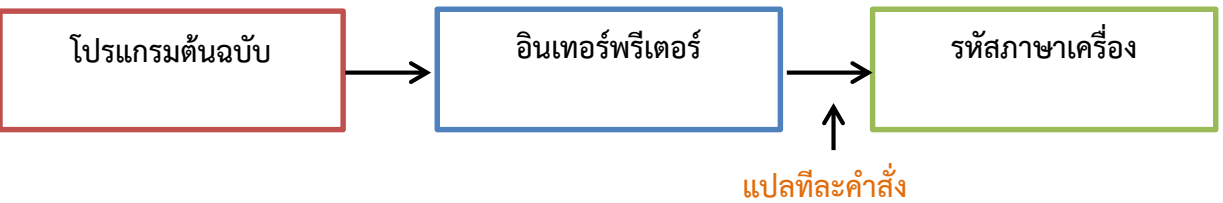
ส่วนฟังก์ชันหลัก

รูปที่ 1.2 แสดงตัวอย่างการเขียนโปรแกรมภาษาซี

ตัวแปลภาษา

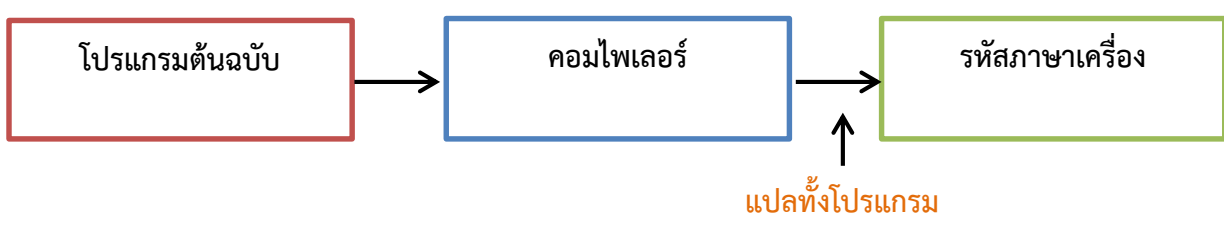
เนื่องจากภาษาคอมพิวเตอร์โดยเฉพาะภาษาระดับสูง มีจุดประสงค์เพื่อให้มนุษย์สามารถสื่อสารเพื่อการเขียนโปรแกรมได้ง่ายขึ้น แต่ภาษาระดับสูงเป็นภาษาที่คอมพิวเตอร์ไม่รู้จัก ดังนั้นจึงต้องนำภาษาระดับสูงมาผ่านกระบวนการแปลเพื่อให้เป็นภาษาเครื่อง ตัวแปลภาษาแบ่งออกเป็น 2 ชนิด ได้แก่

1. อินเทอร์พรีเตอร์ (Interpreter) ตัวแปลชนิดนี้จะทำการแปลทีละคำสั่ง และจะปฏิบัติตามในคำสั่งนั้นๆ หากไม่พบข้อผิดพลาดใดๆ จะนำคำสั่งถัดไปมาแปลต่อ จะกระทำเช่นนี้ไปเรื่อยๆ หากพบข้อผิดพลาด จะหยุดทำงานทันทีและแจ้งข้อผิดพลาดให้ทราบทางหน้าจอ และคำสั่งที่ผิดพลาดนี้มิได้ถูกสั่งให้ทำงาน โปรแกรมก็จะยังทำงานได้อยู่นั่นเอง

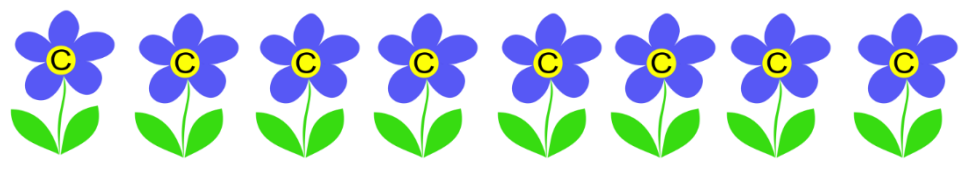


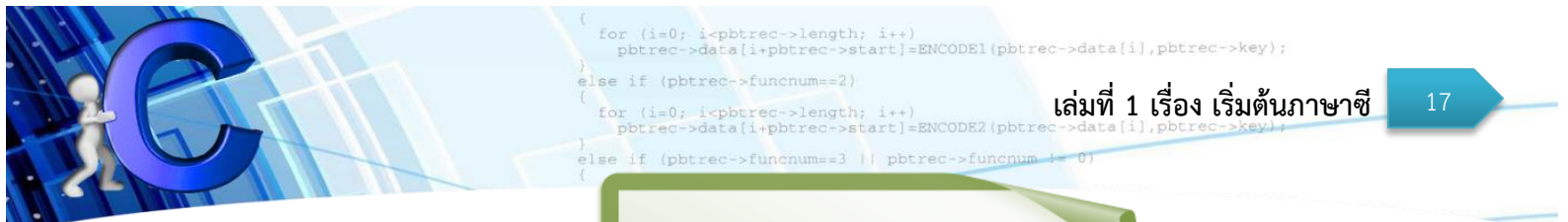
รูปที่ 1.3 แสดงขั้นตอนการทำงานของอินเทอร์พรีเตอร์

2. คอมไพเลอร์ (Compiler) ตัวแปลชนิดนี้จะต่างกับอินเทอร์พรีเตอร์ ตัวแปลชนิดนี้จะแปลแบบทั้งโปรแกรมเพียงครั้งเดียว หากพบข้อผิดพลาด จะแจ้งข้อผิดพลาดทางหน้าจอ โปรแกรมเมอร์จะต้องแก้ไขให้ถูกต้อง แล้วจึงคอมไพล์ใหม่จนไม่พบข้อผิดพลาด



รูปที่ 1.4 แสดงขั้นตอนการทำงานของคอมไพเลอร์





```
{  
    for (i=0; i<pbtre->length; i++)  
        pbtre->data[i+pbtre->start]=ENCODE1(pbtre->data[i],pbtre->key);  
    else if (pbtre->funcnum==2)  
    {  
        for (i=0; i<pbtre->length; i++)  
            pbtre->data[i+pbtre->start]=ENCODE2(pbtre->data[i],pbtre->key);  
    }  
    else if (pbtre->funcnum==3 || pbtre->funcnum != 0)  
    {  
        for (i=0; i<pbtre->length; i++)  
            pbtre->data[i+pbtre->start]=ENCODE3(pbtre->data[i],pbtre->key);  
    }  
}
```

แบบฝึกหัดที่ 1.3

เรื่อง โครงสร้างโปรแกรมภาษาซี

จุดประสงค์การเรียนรู้

1. นักเรียนบอกลักษณะโครงสร้างภาษาซีได้
2. นักเรียนอธิบายลักษณะของตัวแปลภาษาแต่ละชนิดได้

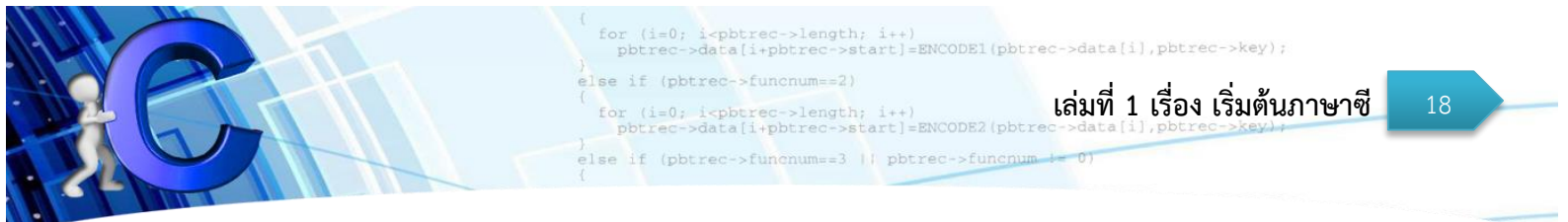
คำชี้แจง

ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้อง

1. โครงสร้างของภาษาซีประกอบด้วยอะไรบ้าง

2. ให้นักเรียนยกตัวอย่างพรีโปรเซสเซอร์ที่มีอยู่ในภาษาซีอย่างน้อย 5 ตัวอย่าง

3. ให้นักเรียนอธิบายความแตกต่างระหว่างตัวแปลภาษาอินเทอร์พรีเตอร์กับคอมไพเลอร์



```
( for (i=0; i<pbtre->length; i++)  
pbtre->data[i+pbtre->start]=ENCODE1(pbtre->data[i],pbtre->key);  
else if (pbtre->funcnum==2)  
{  
for (i=0; i<pbtre->length; i++)  
pbtre->data[i+pbtre->start]=ENCODE2(pbtre->data[i],pbtre->key);  
}  
else if (pbtre->funcnum==3 || pbtre->funcnum != 0)
```

ใบความรู้ที่ 4 เรื่อง การติดตั้งโปรแกรม Dev C



ความรู้เบื้องต้นเกี่ยวกับโปรแกรมภาษาซี

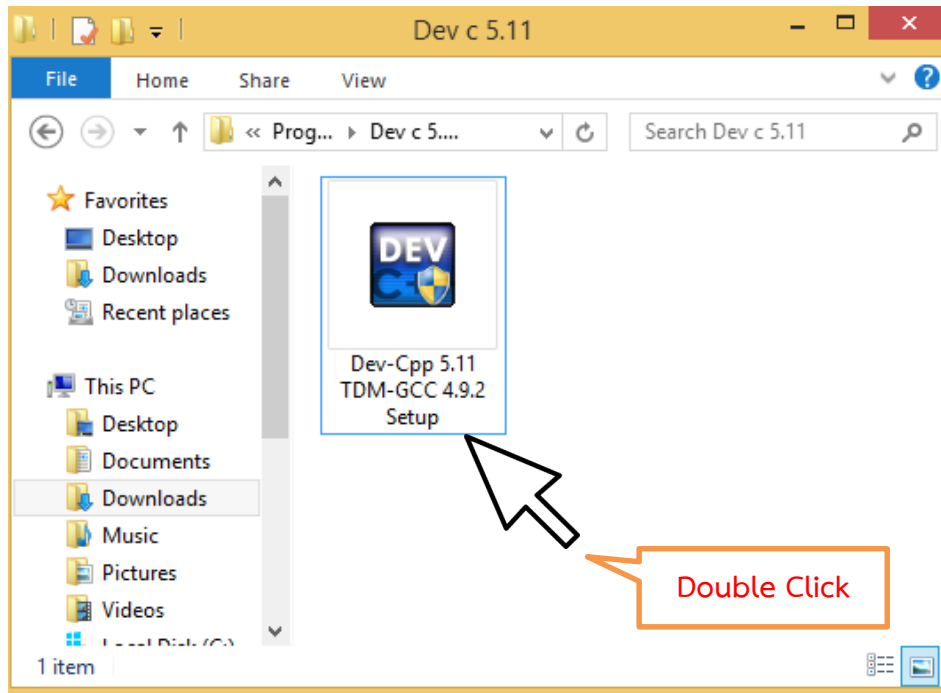
การเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษาต่างๆ สามารถเขียนคำสั่งด้วยการใช้โปรแกรม Text Editor ซึ่งอาจใช้โปรแกรมพวก word Processor เขียนโดยไม่ใช้รูปแบบพิเศษต่าง ๆ หรือใช้โปรแกรมที่มีรูปแบบพิเศษน้อย ๆ เช่น Notepad เป็นต้น เขียนคำสั่งแล้วบันทึกเป็นไฟล์โปรแกรมภาษานั้น ๆ เช่น ภาษาซี จะบันทึกเป็นไฟล์นามสกุล .c แล้วจึงเปลี่ยนภาษาที่เขียนนั้นเป็นภาษาเครื่อง โดยใช้คอมไพเลอร์ (Compiler) ของภาษาคอมพิวเตอร์ที่เขียน เช่น คอมไพเลอร์ของภาษาซี ซึ่งก็มีผู้ผลิตหลายราย และต้องเป็นไปตามมาตรฐาน ANSI C ซึ่งมีรายละเอียดเพิ่มเติมแตกต่างกันไป

ในการเขียนโปรแกรมที่สะดวก ง่ายในการคอมไพล์และสร้างรหัสเป็นภาษาเครื่อง สามารถใช้ Editor และ Compiler ที่รวมกันไว้แล้ว ในชุดพัฒนาหรือเครื่องมือที่ช่วยในการพัฒนาโปรแกรม เรียกว่า IDE (Integrated Development Environment) ซึ่งเป็นโปรแกรมที่ออกแบบมาเพื่อช่วยให้ผู้ที่เขียนโปรแกรมใช้ในการสร้างโปรแกรม โดยจะมี Editor สำหรับเขียนโค้ดของโปรแกรมและมีตัวแปลภาษามาพร้อม ปัจจุบันมีการออกชุดพัฒนามาหลายรุ่นและเปลี่ยนแปลงอย่างรวดเร็ว เช่น Turbo C++, Borland C++, Microsoft C/C++, Microsoft Visual C++, Microsoft Visual C#, Microsoft Visual C++ .NET ซึ่งชุดพัฒนาแต่ละตัวมีวิธีการนำไปใช้งานที่แตกต่างกัน เพราะพัฒนามาจากบริษัทต่างกัน แต่อย่างไรก็ตามการเขียนโปรแกรมภาษาซีไม่ว่าจะเป็น IDE ใด ก็มีหลักและวิธีการที่คล้ายกัน จะต่างกันบ้างตรงรายละเอียดบางอย่าง ที่พัฒนาให้ใช้งานได้ง่ายขึ้น

ในการเรียนการสอนวิชานี้จะใช้ editor และ compiler ที่รวมกันไว้แล้ว ใน ชุดพัฒนาหรือเครื่องมือที่ช่วยในการพัฒนาโปรแกรม ภาษาอังกฤษเรียกว่า IDE (Intregal Devenlopment Environment) ซึ่งเป็นโปรแกรมที่ออกแบบมาเพื่อช่วยในการสร้างโปรแกรมของผู้ที่จะทำการเขียนโปรแกรมทำงานได้ง่ายขึ้น ไม่ต้องแยกใช้ editor เขียนโปรแกรม แล้วเรียกใช้ compiler ทำการคอมไพล์โปรแกรมอีก โดย IDE ก็มีผู้ผลิตหลายราย โดยในการเรียนการสอนวิชานี้จะใช้ IDE คือ Bloodshed Dev-C++

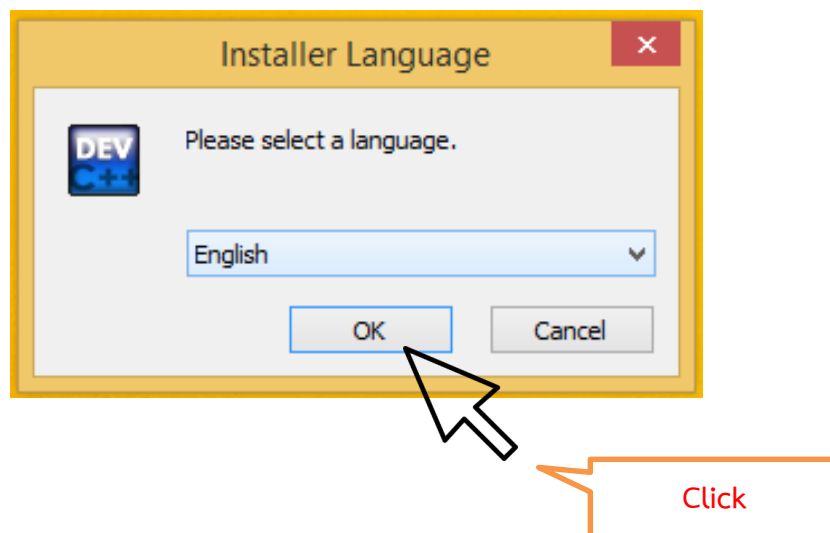
ขั้นตอนการติดตั้งโปรแกรม Dev c 5.11

1. เปิด Folder Dev c 5.11 แล้ว Double Click ที่ File Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup



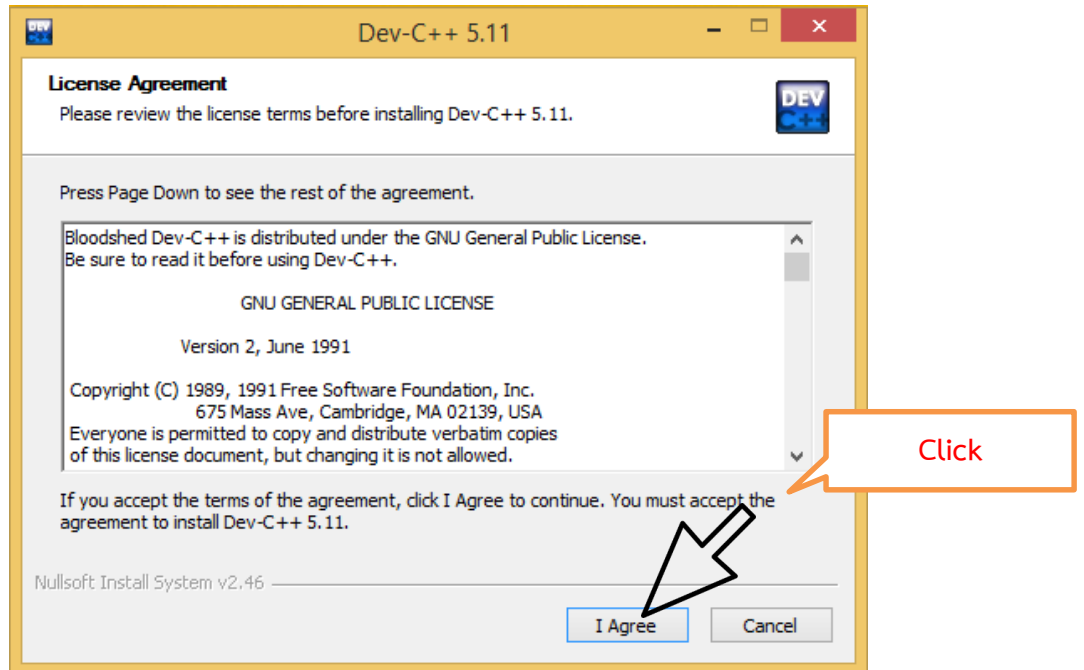
ภาพที่ 1.5 แสดงหน้าต่างไฟล์ที่ใช้ในการติดตั้งโปรแกรม Dev c

2. จะเข้าสู่หน้าต่างเลือกภาษาในการติดตั้ง ให้เลือกภาษา English กดปุ่ม OK > เพื่อดำเนินการต่อ



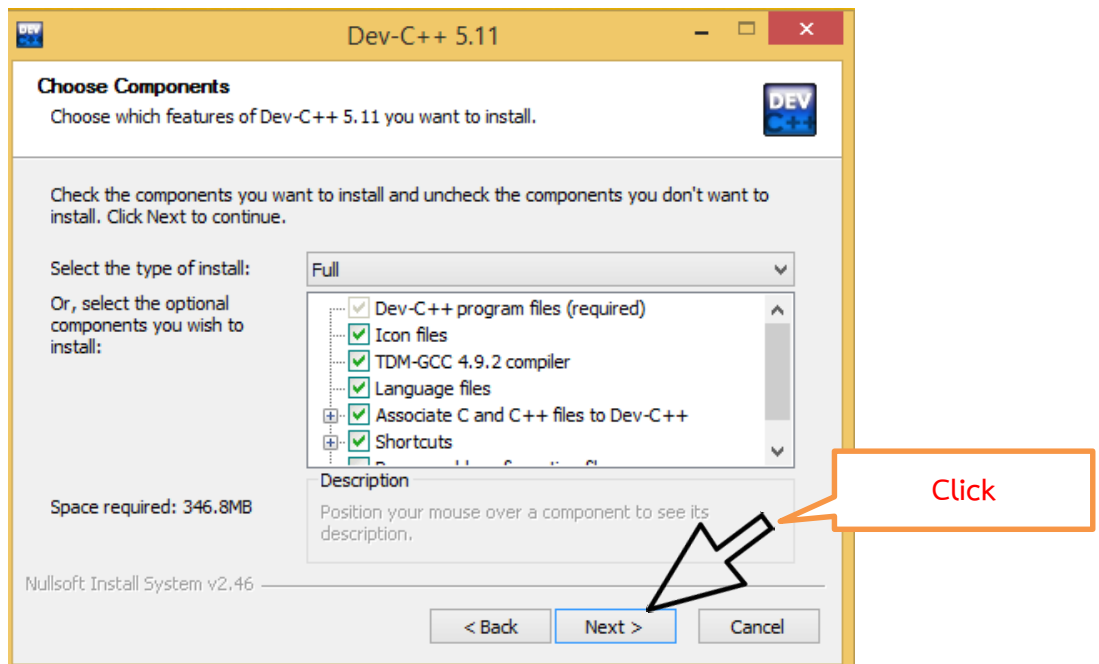
ภาพที่ 1.6 แสดงหน้าต่างการเลือกภาษาในการติดตั้ง

3. จะเข้าสู่หน้าต่างเงื่อนไขการใช้งานโปรแกรมของบริษัทผู้ผลิต คลิก I Agree เพื่อยอมรับเงื่อนไข



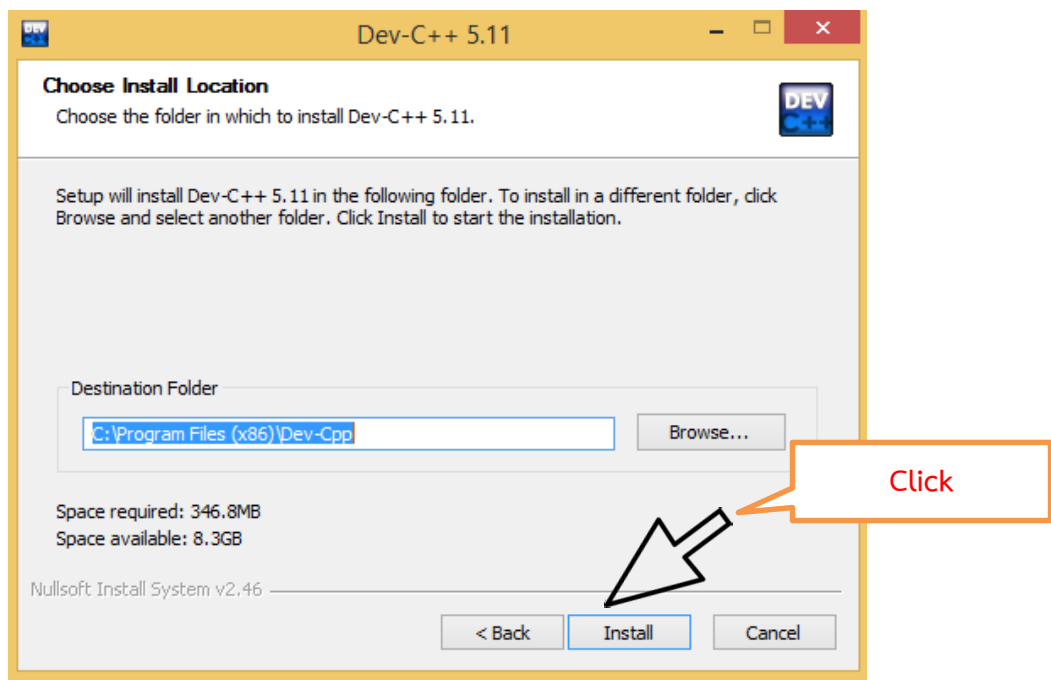
ภาพที่ 1.7 แสดงหน้าต่างเงื่อนไขการใช้งานโปรแกรม

4. จะเข้าสู่หน้าต่างการติดตั้งส่วนเสริมต่างๆ เพิ่มเติมให้กับโปรแกรม คลิก Next> เพื่อดำเนินการต่อ



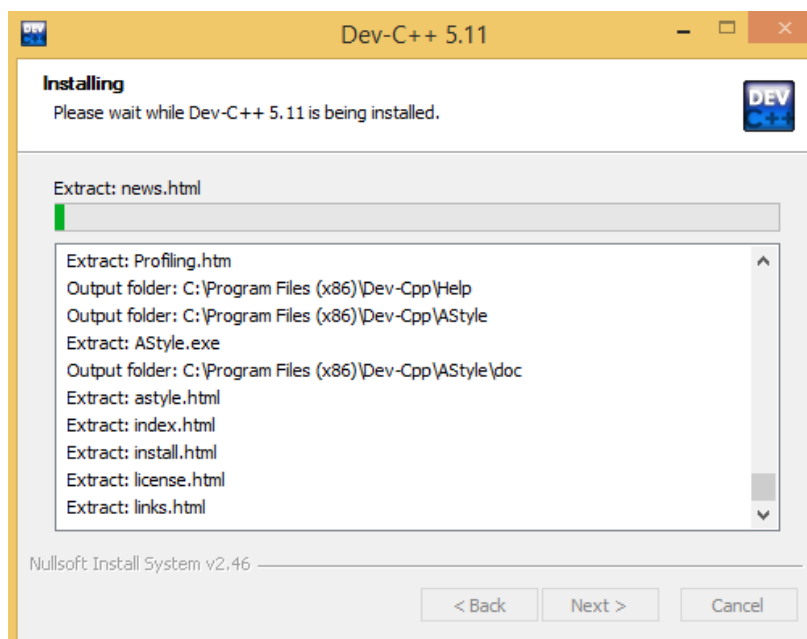
ภาพที่ 1.8 แสดงหน้าต่างการติดตั้งส่วนเสริมต่างๆของโปรแกรม

5. เข้าสู่หน้าต่างการเลือกตำแหน่งแฟ้มเพื่อการติดตั้ง คลิก Install



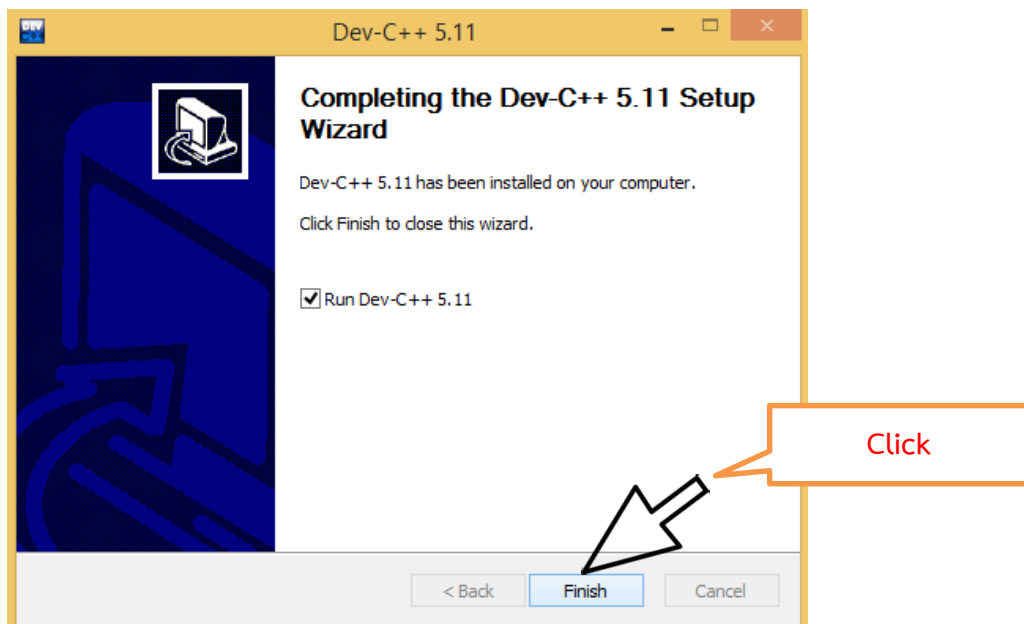
ภาพที่ 1.9 แสดงหน้าต่างการเลือกตำแหน่งแฟ้มเพื่อการติดตั้ง

6. โปรแกรมกำลังติดตั้งลงเครื่อง รอจนกระทั่งเสร็จสมบูรณ์



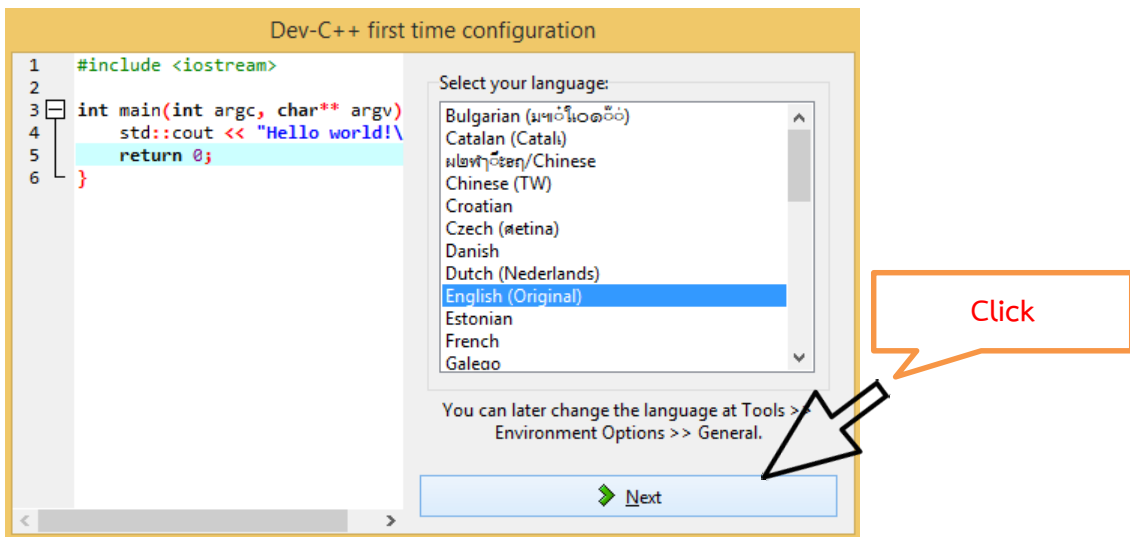
ภาพที่ 1.10 แสดงหน้าต่างความคืบหน้าของการติดตั้งโปรแกรม

7. หน้าต่างแสดงการติดตั้งเรียบร้อยแล้ว คลิก Finish



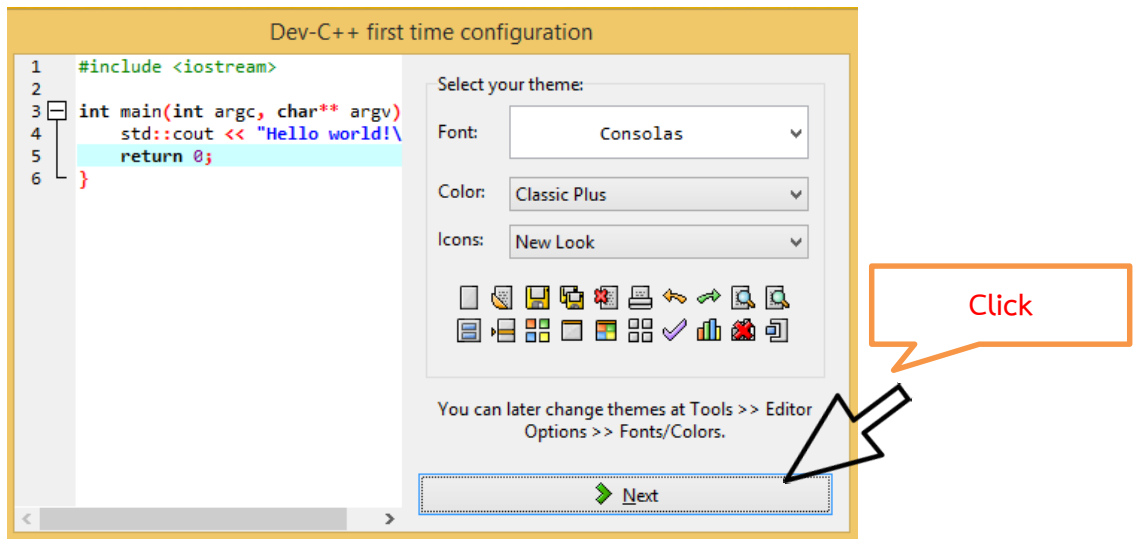
ภาพที่ 1.11 แสดงหน้าต่างสิ้นสุดการติดตั้งโปรแกรม

8. หน้าต่างเลือกภาษาในการใช้งานเริ่มต้น คลิก Next



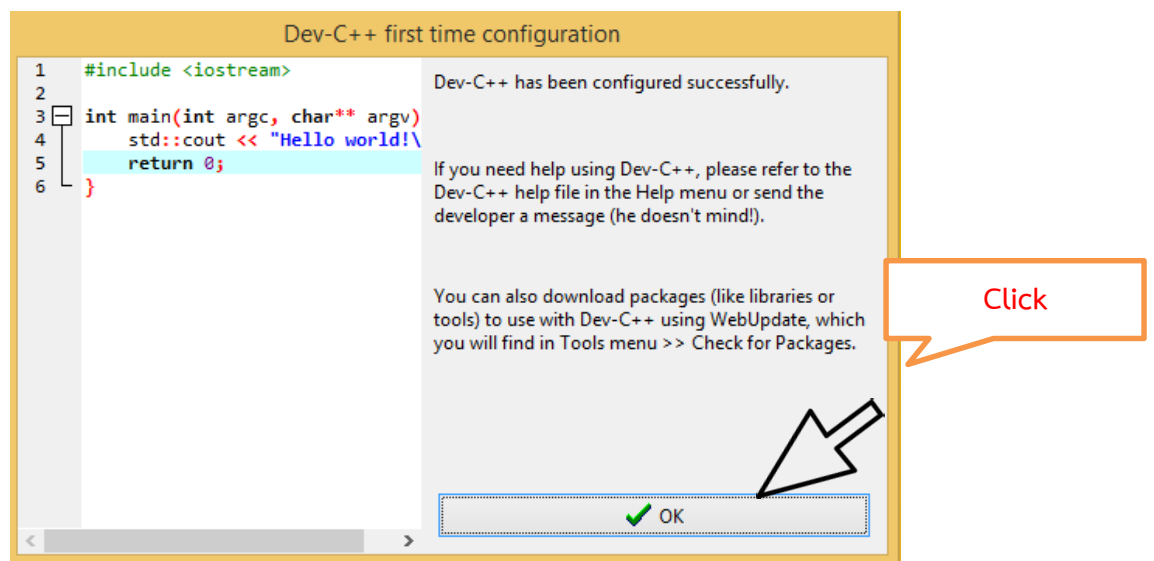
ภาพที่ 1.12 แสดงหน้าต่างการเลือกภาษาในการใช้งานเริ่มต้น

9. หน้าต่างเลือก คุณสมบัติของแบบอักษร สี และไอคอน คลิก Next



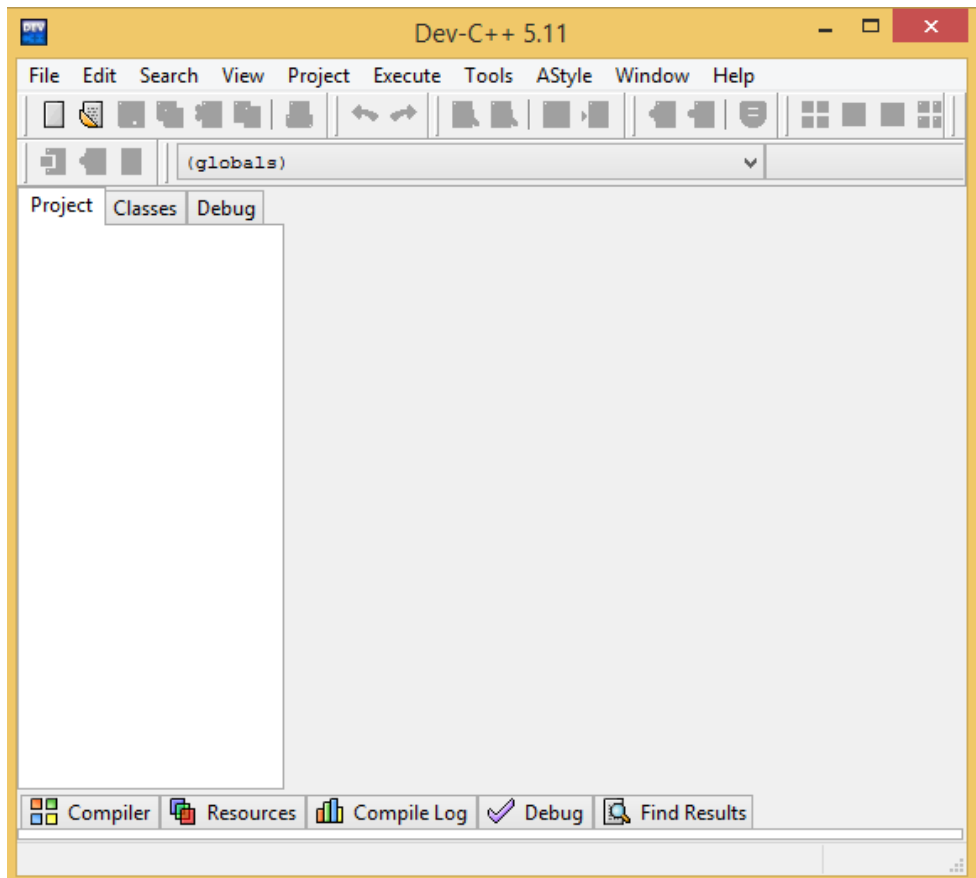
ภาพที่ 1.13 แสดงหน้าต่างการกำหนดคุณสมบัติของแบบอักษร สี และไอคอน

10. หน้าต่างแจ้งการกำหนดค่าต่างๆ ของโปรแกรมเสร็จสมบูรณ์ คลิก OK



ภาพที่ 1.14 แสดงหน้าต่างแจ้งการกำหนดค่าต่างๆ ของโปรแกรมเสร็จ

11. หน้าต่างการใช้งานโปรแกรม Dev C 5.11



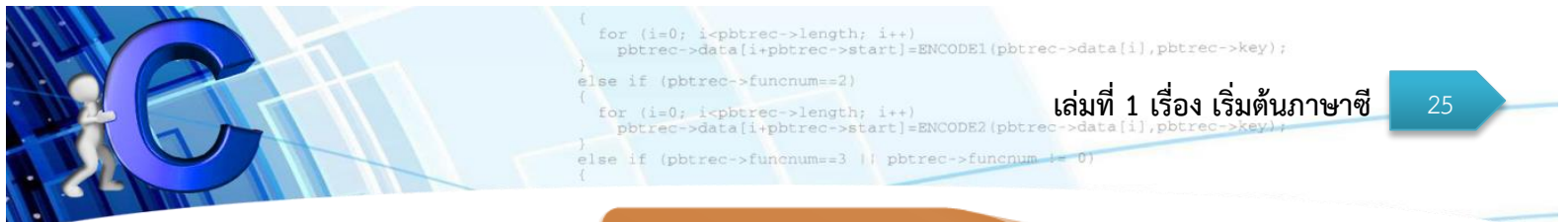
ภาพที่ 1.15 แสดงหน้าต่างโปรแกรม Dev-c ++ 5.11

12. เมื่อติดตั้งโปรแกรม Dev c 5.11 ลงในเครื่องแล้ว จะปรากฏไอคอนเข้าสู่โปรแกรมอยู่บน Desktop ของเครื่องคอมพิวเตอร์



สำหรับ





```
{ for (i=0; i<pbtrac->length; i++)  
  pbtrac->data[i+pbtrac->start]=ENCODE1(pbtrac->data[i],pbtrac->key);  
}  
else if (pbtrac->funcnum==2)  
{  
  for (i=0; i<pbtrac->length; i++)  
    pbtrac->data[i+pbtrac->start]=ENCODE2(pbtrac->data[i],pbtrac->key);  
}  
else if (pbtrac->funcnum==3 || pbtrac->funcnum != 0)
```

ใบงานที่ 1.1

เรื่อง การติดตั้งโปรแกรม Dev c

จุดประสงค์การเรียนรู้

1. นักเรียนเข้าใจความรู้เบื้องต้นเกี่ยวกับการติดตั้งโปรแกรมภาษาซี
2. นักเรียนสามารถติดตั้งโปรแกรม Dev c ได้

คำชี้แจง

ให้นักเรียนปฏิบัติตามหัวข้อต่อไปนี้

1. ติดตั้งโปรแกรม Dev C 5.11 ลงเครื่องที่นักเรียนใช้อยู่
2. เมื่อติดตั้งเสร็จ ให้เรียกโปรแกรม Dev C 5.11 ขึ้นมาเพื่อมาใช้งาน
3. เขียนปัญหาที่พบในการติดตั้งโปรแกรม และวิธีแก้ปัญหา

ปัญหาที่พบ

.....

.....

.....

.....

วิธีแก้ปัญหา.....

.....

.....

.....

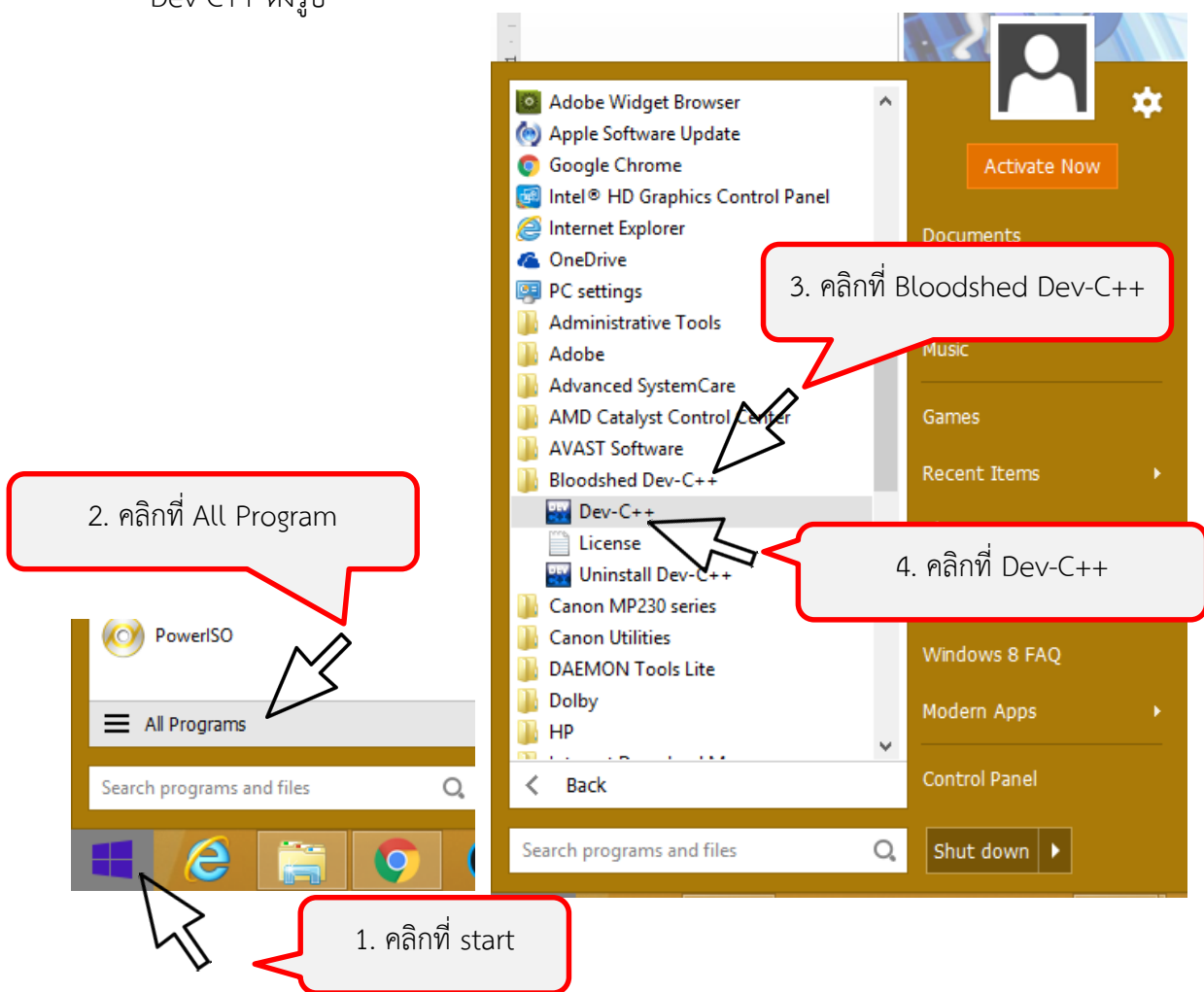
.....

ใบความรู้ที่ 5

เรื่อง การใช้งานโปรแกรม Dev C

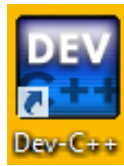
1. การเรียกใช้งานโปรแกรม Dev-C++

การเรียกใช้ Dev-C++ ไปที่ Start >All Programs> Bloodshed Dev-C++ คลิกที่ Dev-C++ ดังรูป



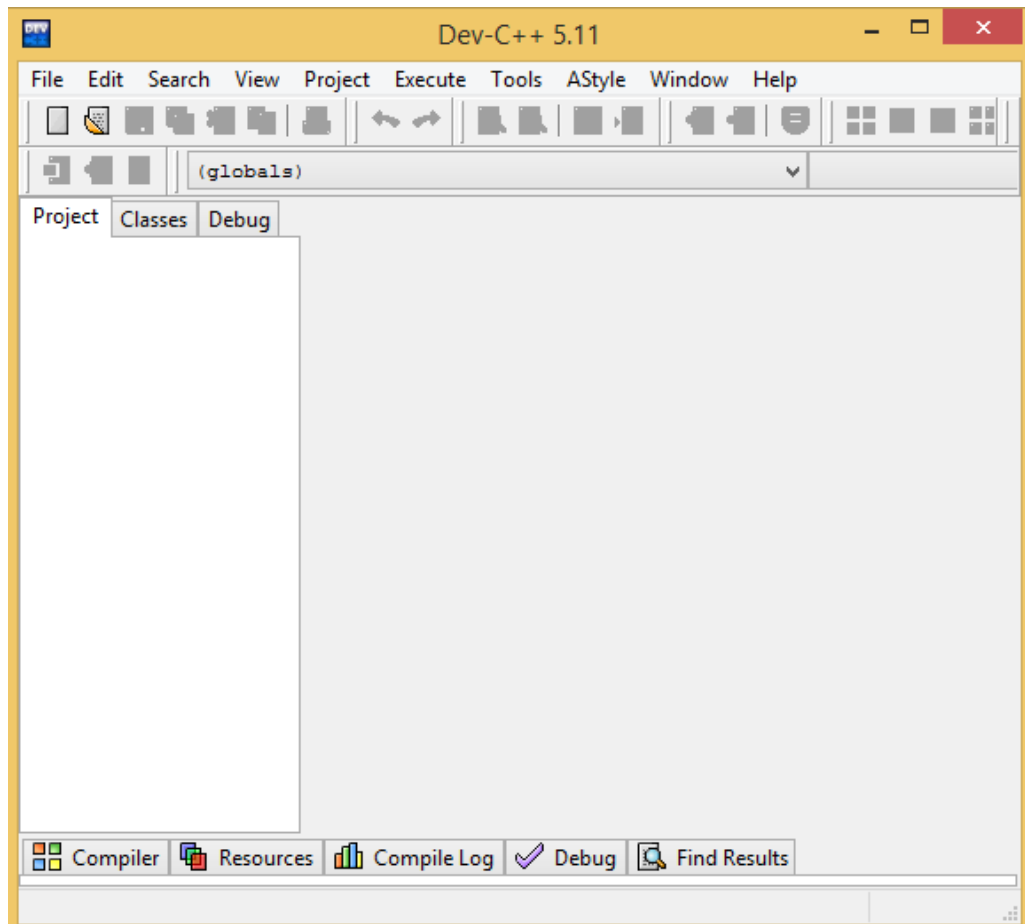
ภาพที่ 1.16 แสดงหน้าต่างการเรียกใช้โปรแกรม Dev-c ++

หรือคลิกที่ไอคอน



สำหรับเข้าสู่โปรแกรมอยู่บน Desktop ของเครื่องคอมพิวเตอร์

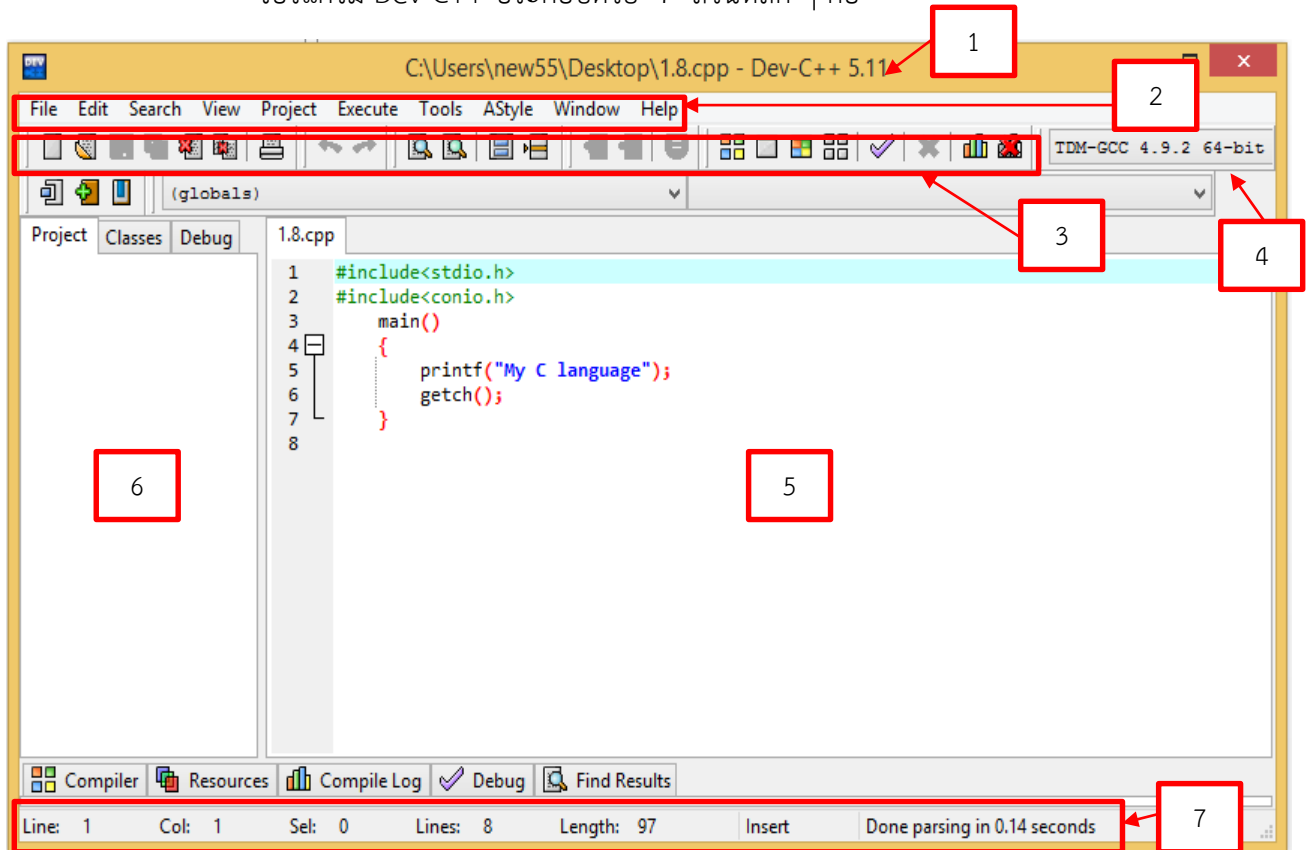
จะได้หน้าต่างโปรแกรม Dev-C++ ดังรูป



ภาพที่ 1.17 แสดงหน้าต่างโปรแกรม Dev-c ++ 5.11

2. ส่วนประกอบของโปรแกรม Dev-C++

โปรแกรม Dev-C++ ประกอบด้วย 7 ส่วนหลัก ๆ คือ



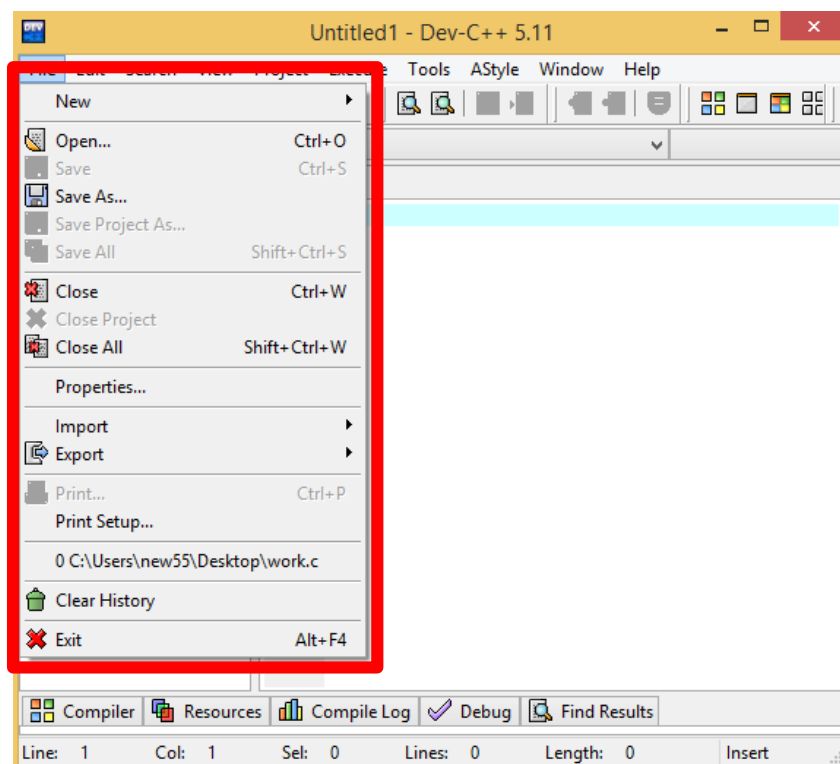
ภาพที่ 1.18 แสดงหน้าต่างส่วนประกอบโปรแกรม Dev-c ++ 5.11

- 1) ส่วนของแถบไทเทิล (Title Bar) เป็นแถบที่อยู่บนสุดของโปรแกรม จะมีชื่อโปรแกรม ภาษาซี แจ้งให้ผู้ใช้ทราบว่า กำลังทำกับแฟ้มใด เช่น Untitled1 - Dev-C++ 5.11
- 2) ส่วนของแถบเมนู (Menu Bar) ส่วนนี้เป็นส่วนของเมนูคำสั่งต่าง ๆ 10 รายการ คือ File Edit Search View Project Execute Tools AStyle Window Help
- 3) ส่วนแถบเครื่องมือ (Tool Bars) เป็นส่วนของเครื่องมือ ที่รวมเอาเมนูคำสั่งที่ใช้บ่อย ๆ เพื่ออำนวยความสะดวกในการเรียกใช้แทนเมนูที่มีขั้นตอนเข้าใช้หลายขั้นตอน
- 4) ส่วนปุ่มรายการเลือก TDM-GCC ของโปรแกรม Dev-C++ ให้รองรับ 32-bit หรือ 64-bit
- 5) ส่วนของการแสดง Project/Classes/Debug เป็นส่วนที่อยู่ทางด้านซ้ายของโปรแกรม ใช้เพื่อแสดง Project หรือ Class ต่าง ๆ ของโปรแกรม

```
for (i=0; i<pbtre->length; i++)
    pbtre->data[i+pbtre->start]=ENCODE1(pbtre->data[i],pbtre->key);
}
else if (pbtre->funcnum==2)
{
    for (i=0; i<pbtre->length; i++)
        pbtre->data[i+pbtre->start]=ENCODE2(pbtre->data[i],pbtre->key);
}
else if (pbtre->funcnum==3 || pbtre->funcnum != 0)
```

- 6) ส่วนของพื้นที่การเขียนโปรแกรม (Editor) เป็นส่วนของพื้นที่ทำงานในการเขียนรหัสโปรแกรมภาษาซี
 - 7) ส่วนแสดงสถานะของโปรแกรม (Status) อยู่ด้านล่างสุด ใช้เพื่อบอกสถานะต่าง ๆ ในขณะที่กำลังเขียนโปรแกรม เช่น จำนวนบรรทัดทั้งหมด หรือสถานะการพิมพ์แทรก/พิมพ์ทับ
3. การใช้งานเมนูต่าง ๆ ของโปรแกรม Dev-C++

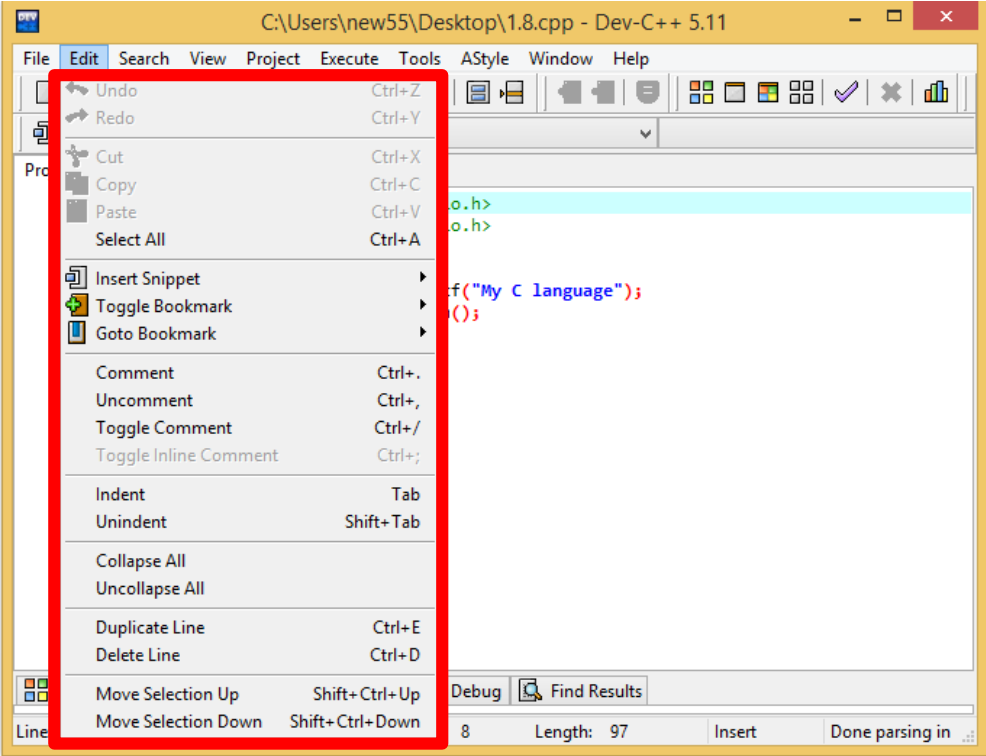
- 1) เมนู File ประกอบด้วยเมนูย่อย ๆ ที่ควรรู้จักดังนี้



ภาพที่ 1.19 แสดงหน้าต่างเมนู File โปรแกรม Dev-c ++ 5.11

New	-->	เพื่อสร้างไฟล์ใหม่
Open	-->	เพื่อเปิดไฟล์ที่บันทึกไว้
Save	-->	เพื่อบันทึกไฟล์ทับไฟล์เดิม
Save As	-->	เพื่อบันทึกไฟล์เป็นชื่อใหม่ หรือเลือกบันทึกลงในโฟลเดอร์อื่น
Save All	-->	เพื่อบันทึกไฟล์หรือโปรเจกซ์ทั้งหมด ที่เปิดทำงานอยู่
Close	-->	เพื่อปิดไฟล์ที่กำลังใช้อยู่
Close All	-->	เพื่อปิดไฟล์หรือโปรเจกซ์ทั้งหมดที่เปิดอยู่
Exit	-->	เพื่อปิดโปรแกรม Bloodshed Dev-C++

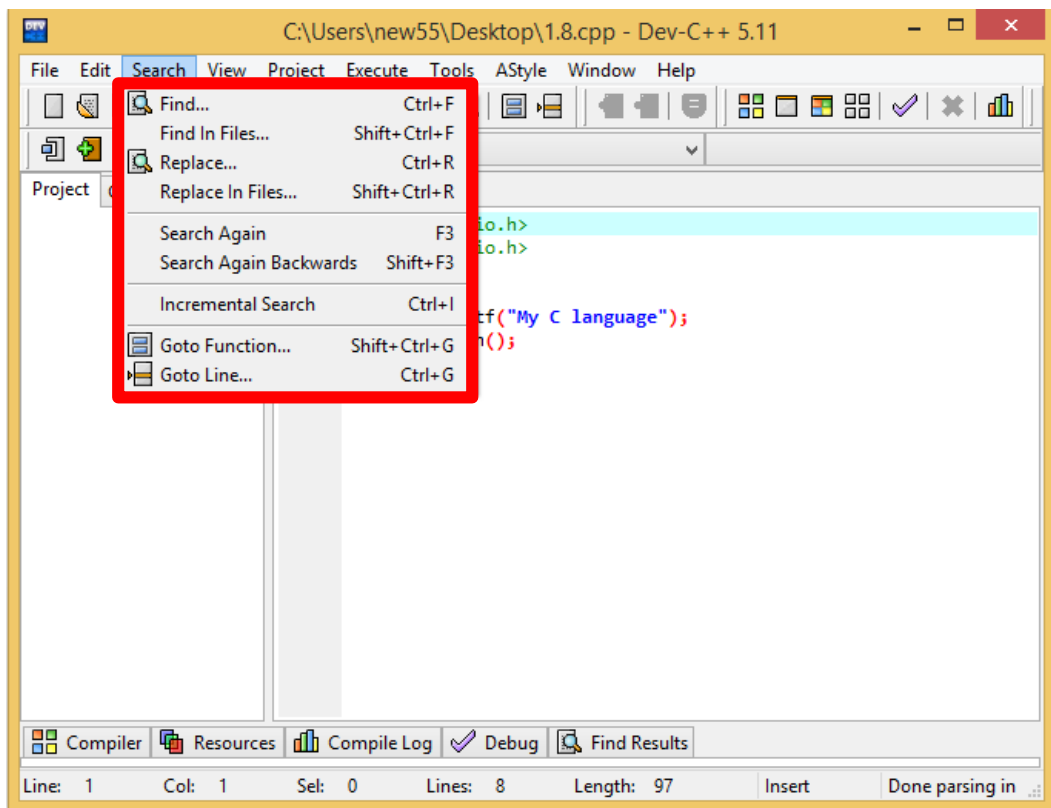
2) เมนู Edit ประกอบด้วยเมนูย่อย ๆ ที่ควรรู้จักดังนี้



ภาพที่ 1.20 แสดงหน้าต่างเมนู Edit โปรแกรม Dev-c ++ 5.11

- | | | |
|----------------|-----|---|
| Undo | --> | ยกเลิกคำสั่ง 1 ครั้ง |
| Redu | --> | ให้ทำซ้ำคำสั่ง 1 ครั้ง |
| Cut | --> | ตัดข้อความที่เลือก ไปไว้ในคลิปบอร์ด |
| Copy | --> | คัดลอกข้อความที่เลือก ไปไว้ในคลิปบอร์ด |
| Paste | --> | วางข้อความจากคลิปบอร์ด ลงในตำแหน่งเคอร์เซอร์ |
| Select All | --> | เลือกโค้ด หรือข้อความทั้งหมด ที่อยู่ใน Editor |
| Insert Snippet | --> | เป็นเครื่องมือช่วยจำรูปแบบคำสั่ง โดยสามารถเลือกแทรกรูปแบบคำสั่งต่าง ๆ ได้ |

3) เมนู Search ประกอบด้วยเมนูย่อย ๆ ที่ควรรู้จักดังนี้

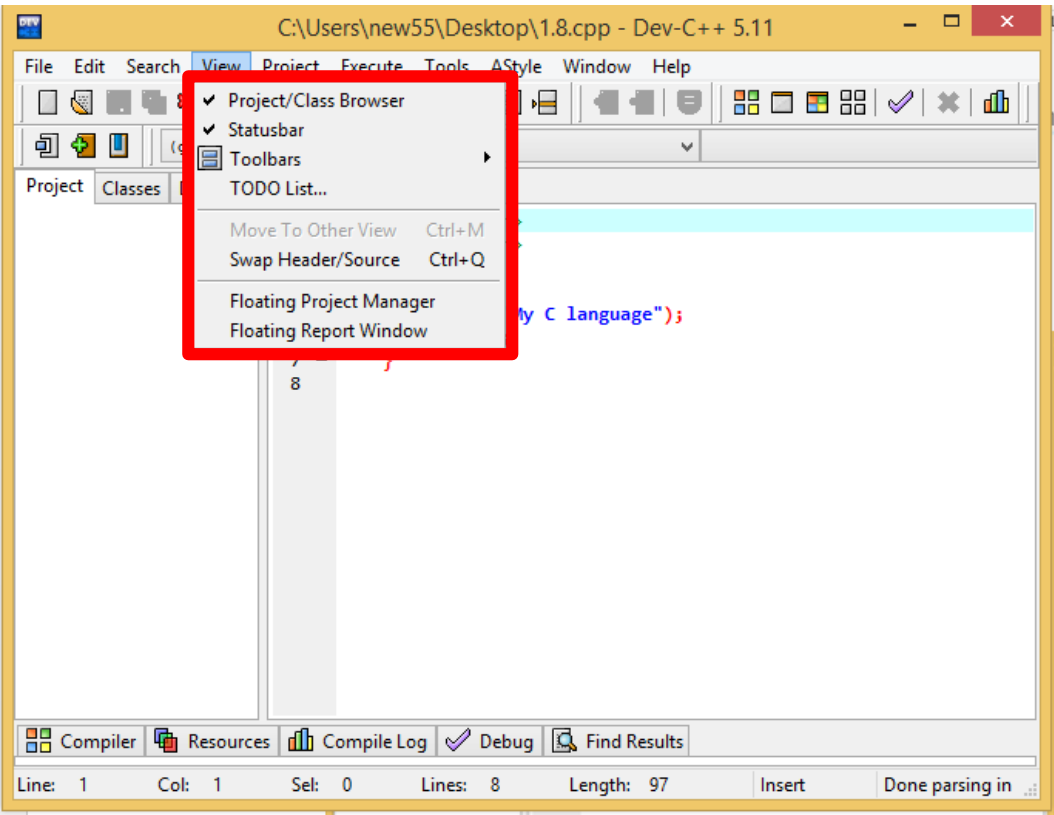


ภาพที่ 1.21 แสดงหน้าต่างเมนู Search โปรแกรม Dev-c ++ 5.11

Find	-->	ค้นหาคำ หรือข้อความใน Editor
Replace	-->	ให้แทนที่คำที่ค้นหา ด้วยคำใหม่
Search Again	-->	ให้ค้นซ้ำ หรือค้นหาต่อไป



4) เมนู View ประกอบด้วยเมนูย่อย ๆ ที่ควรรู้จักดังนี้



ภาพที่ 1.22 แสดงหน้าต่างเมนู View โปรแกรม Dev-c ++ 5.11

- Project/Class Browser

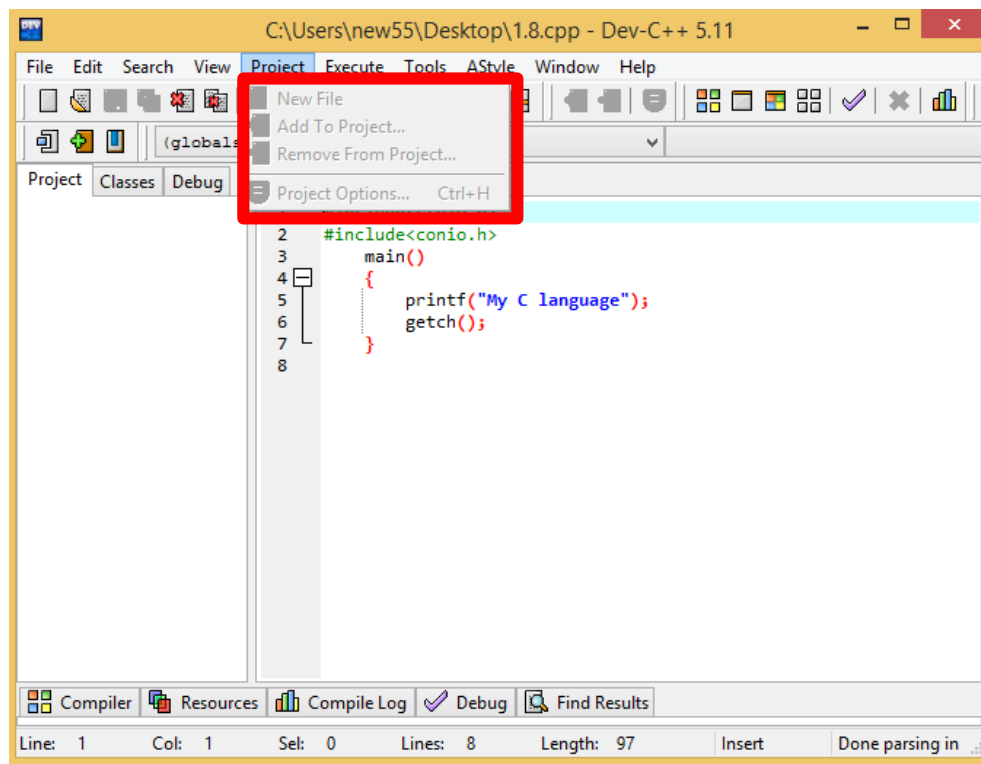
--> เพื่อแสดงหรือไม่แสดง Project/Class Browser
- Statusbar

--> เพื่อแสดงหรือไม่แสดง Statusbar
- Toolbars

--> เพื่อแสดงหรือไม่แสดง Toolbars ต่าง ๆ



5) เมนู Project ประกอบด้วยเมนูย่อย ๆ ที่ควรรู้จักดังนี้

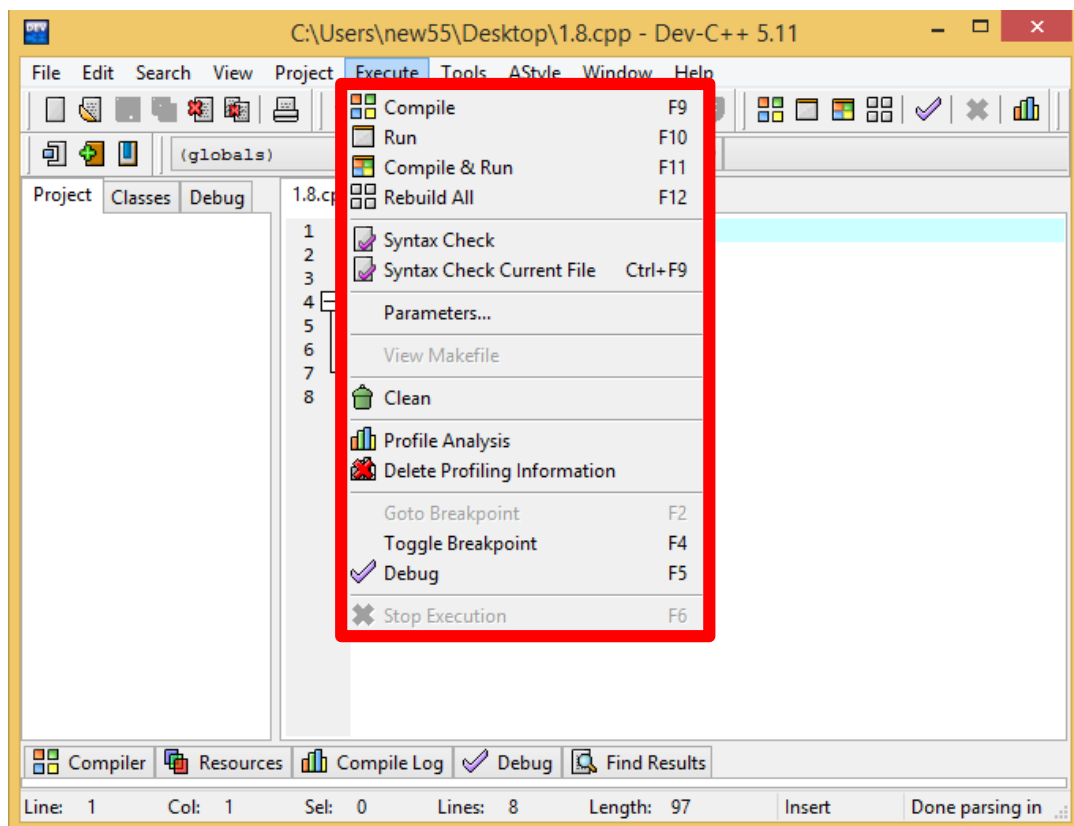


ภาพที่ 1.23 แสดงหน้าต่างเมนู Project โปรแกรม Dev-c ++ 5.11

New file	-->	เพื่อสร้างไฟล์ใหม่ในโปรเจค
Add to Project	-->	เพื่อเพิ่มไฟล์เข้าโปรเจค
Remove from Project	-->	เพื่อย้ายไฟล์ออกจากโปรเจค



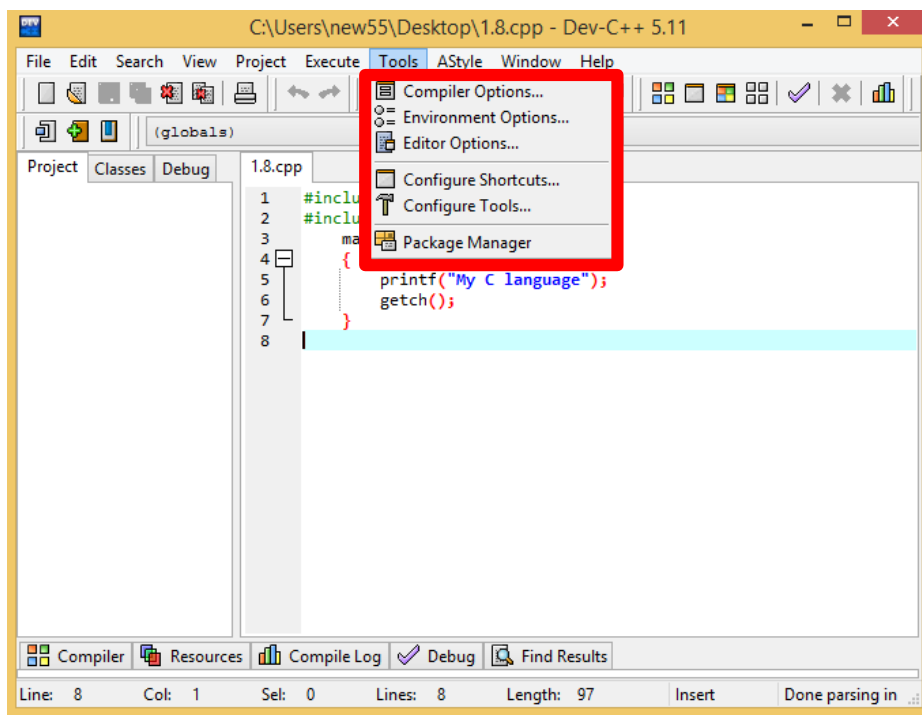
6) เมนู Execute ประกอบด้วยเมนูย่อย ๆ ที่ควรรู้จักดังนี้



ภาพที่ 1.24 แสดงหน้าต่างเมนู Execute โปรแกรม Dev-c ++ 5.11

Compile (F9)	-->	สั่งให้ Compile ซอร์สโค้ด ถ้าผ่านจะได้ไฟล์ .exe
Run (F10)	-->	สั่งให้โปรแกรม .exe ทำงาน
Compile & Run (F11)	-->	สั่งให้ Compile ถ้าผ่านให้ Run ไฟล์ .exe ต่อเลย
Rebuild All (F12)	-->	สั่งให้สร้างไฟล์ .exe ใหม่ แทนที่ไฟล์เดิม

7) เมนู tools ประกอบด้วยเมนูย่อย ๆ ที่ควรรู้จักดังนี้

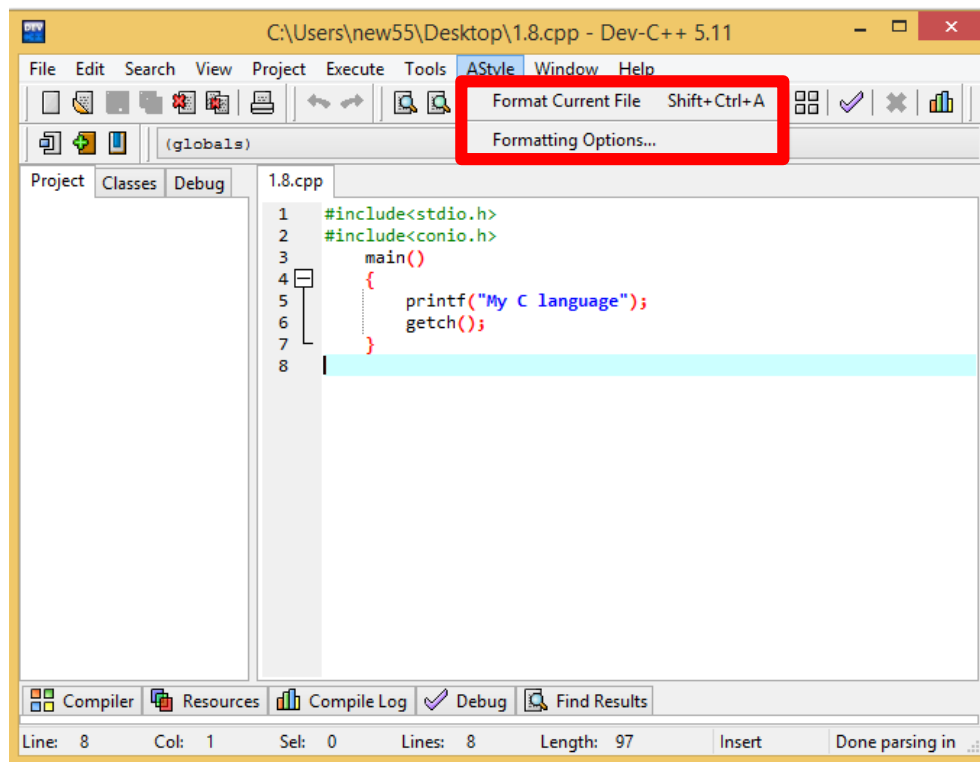


ภาพที่ 1.25 แสดงหน้าต่างเมนู tools โปรแกรม Dev-c ++ 5.11

Editor Options --> เป็นการตั้งค่าสภาพแวดล้อมให้กับ Editor เช่น กำหนด
ขนาด Tabs สี แบบอักษร เป็นต้น



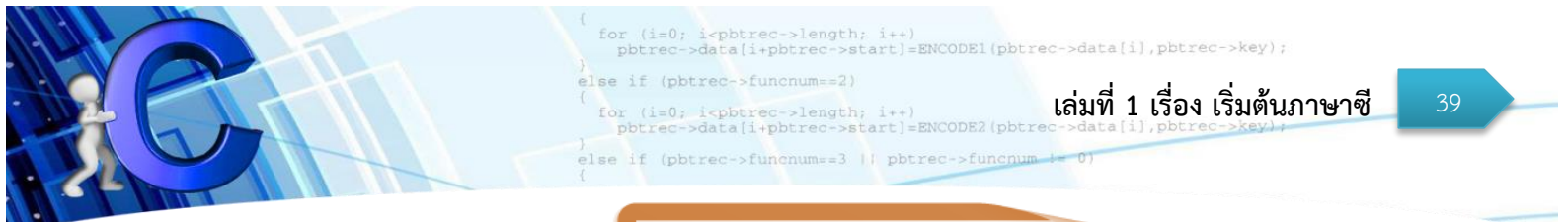
8) เมนู AStyle ประกอบด้วยเมนูย่อย ๆ ที่ควรรู้จักดังนี้



ภาพที่ 1.26 แสดงหน้าต่างเมนู ASyle โปรแกรม Dev-c ++ 5.11

- | | | |
|---------------------|-----|-----------------------|
| Format Current File | --> | จัดรูปแบบไฟล์ปัจจุบัน |
| Formatting Options | --> | ตัวเลือกการจัดรูปแบบ |





```
{  
    for (i=0; i<pbtrc->length; i++)  
        pbtrc->data[i+pbtrc->start]=ENCODE1(pbtrc->data[i],pbtrc->key);  
    else if (pbtrc->funcnum==2)  
    {  
        for (i=0; i<pbtrc->length; i++)  
            pbtrc->data[i+pbtrc->start]=ENCODE2(pbtrc->data[i],pbtrc->key);  
    }  
    else if (pbtrc->funcnum==3 || pbtrc->funcnum != 0)  
    {
```

ใบงานที่ 1.2

เรื่อง กาใช้งานโปรแกรม Dev c

จุดประสงค์การเรียนรู้

1. นักเรียนเข้าใจความรู้เบื้องต้นเกี่ยวกับการใช้โปรแกรมภาษาซี
2. นักเรียนสามารถใช้งานโปรแกรม Dev c ได้

คำชี้แจง

ให้นักเรียนปฏิบัติตามขั้นตอนดังต่อไปนี้

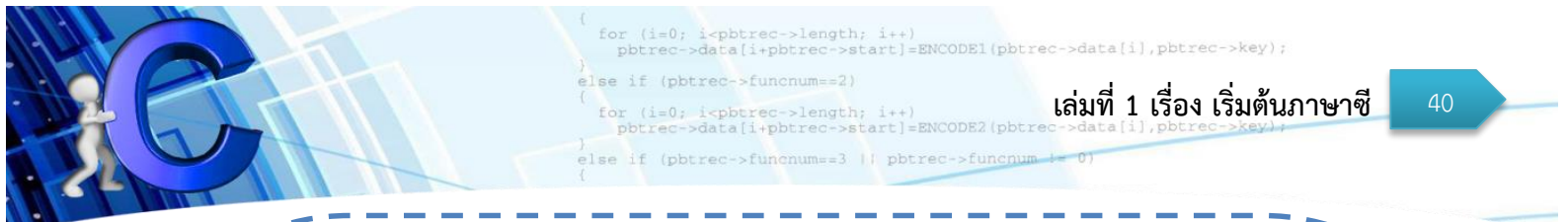
1. สร้าง Folder ไว้ใน Drive D โดยตั้งชื่อของนักเรียนเอง
2. เปิดโปรแกรม Dev-c ++ 5.11 และสร้างไฟล์งานใหม่
3. พิมพ์คำสั่งตามตัวอย่างด้านล่าง

```
work.cpp  
1  #include<stdio.h>  
2  #include<conio.h>  
3      main()  
4  {  
5      printf("My C language");  
6      getch();  
7  }  
8
```

4. บันทึกไฟล์โดยตั้งชื่อว่า work.cpp เลือกที่อยู่ไฟล์เป็นโฟลเดอร์ที่นักเรียนสร้างไว้
5. ทดสอบการเขียนโปรแกรมโดยการ Compile และ Run โปรแกรม สังเกตผลลัพธ์ที่เกิดขึ้น
6. ทดลองการเพิ่มคำว่า "Takfawichaprasit School" ดังรูปด้านล่าง

```
*) work.cpp  
1  #include<stdio.h>  
2  #include<conio.h>  
3      main()  
4  {  
5      printf("My C language");  
6      printf("Takfawichaprasit School");  
7      getch();  
8  }
```

7. ทดสอบการเขียนโปรแกรมโดยการ Compile และ Run โปรแกรม สังเกตผลลัพธ์ที่เกิดขึ้น



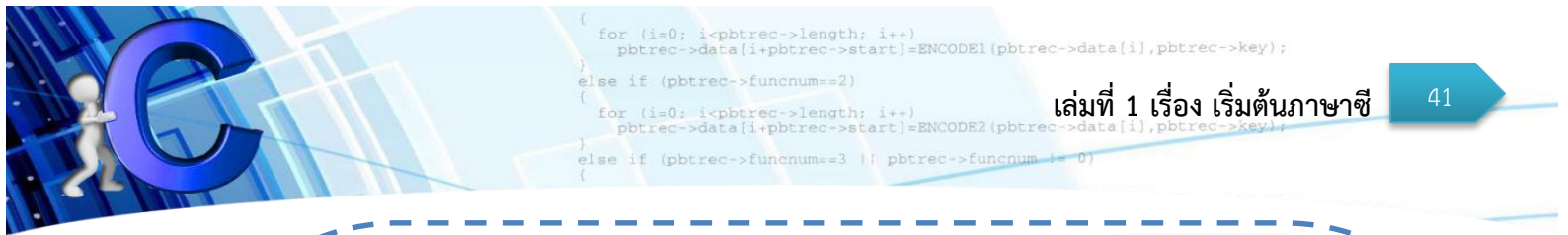
```
( for (i=0; i<pbtrac->length; i++)  
pbtrac->data[i+pbtrac->start]=ENCODE1(pbtrac->data[i],pbtrac->key);  
else if (pbtrac->funcnum==2)  
{  
for (i=0; i<pbtrac->length; i++)  
pbtrac->data[i+pbtrac->start]=ENCODE2(pbtrac->data[i],pbtrac->key);  
}  
else if (pbtrac->funcnum==3 || pbtrac->funcnum != 0)  
{
```

แบบทดสอบหลังเรียน

เรื่อง ขั้นตอนการพัฒนาโปรแกรมและประวัติภาษาซี

- คำชี้แจง 1. แบบทดสอบฉบับนี้มี 10 ข้อ ข้อละ 1 คะแนน คะแนนเต็ม 10 คะแนน เวลา 20 นาที
2. แบบทดสอบนี้เป็นแบบเลือกตอบ
3. ให้นักเรียนทำเครื่องหมาย X ลงในกระดาษคำตอบ โดยเลือกตัวอักษร ก ข ค และ ง ที่ถูกต้องที่สุดเพียงข้อเดียว

1. ข้อใดไม่ใช่ขั้นตอนการพัฒนาโปรแกรมโดยใช้ภาษา C
 - ก. การเขียนและแก้ไขโปรแกรม
 - ข. คอมไพล์โปรแกรม
 - ค. การประกาศตัวแปร
 - ง. การลิงค์โปรแกรม
2. ภาษาซีสืบสานมาจากภาษาใดบ้าง
 - ก. Pascal
 - ข. B
 - ค. Java
 - ง. HTML
3. ภาษาซีจัดเป็นภาษาระดับใด
 - ก. ระดับต่ำ
 - ข. ระดับกลาง
 - ค. ภาษาเครื่อง
 - ง. ระดับสูง
4. ข้อใดคือผู้พัฒนาภาษาซี
 - ก. เดนิส ริชี
 - ข. ชาลส์ แบบเบจ
 - ค. เอดา เลิฟเลซ
 - ง. ทิม เบอร์เนอรส์-ลี
5. โปรแกรมภาษาซีจะเริ่มทำงานจากฟังก์ชันใด
 - ก. ฟังก์ชัน main
 - ข. ฟังก์ชัน include
 - ค. ฟังก์ชัน library
 - ง. ฟังก์ชัน declare



```
(  
    for (i=0; i<pbtre->length; i++)  
        pbtre->data[i+pbtre->start]=ENCODE1(pbtre->data[i],pbtre->key);  
    else if (pbtre->funcnum==2)  
    {  
        for (i=0; i<pbtre->length; i++)  
            pbtre->data[i+pbtre->start]=ENCODE2(pbtre->data[i],pbtre->key);  
    }  
    else if (pbtre->funcnum==3 || pbtre->funcnum != 0)  
    {
```

6. เครื่องหมายใดต่อไปนี้ ที่นำมาใช้สำหรับอธิบายรายละเอียดภายในโปรแกรม
- ก. @ @
 - ข. /* */
 - ค. []
 - ง. { }
7. ประโยคคำสั่งในภาษาซีต้องลงท้ายด้วยเครื่องหมายใด
- ก. , (comma)
 - ข. \n
 - ค. ; (semi-colon)
 - ง. : (colon)
8. ภาษาซีจัดเป็นภาษาที่ใช้ตัวแปลภาษาชนิดใด
- ก. Compiler
 - ข. Translator
 - ค. Interpreter
 - ง. Assembler
9. หากต้องการนำฟังก์ชันมาตรฐาน (stdio.h) มาใช้งาน จะต้องประกาศอย่างไร
- ก. main(stdio.h)
 - ข. { stdio.h }
 - ค. printf("<stdio.h>")
 - ง. #include "stdio.h"
10. ข้อใดไม่ใช่ส่วนประกอบหลักของโครงสร้างภาษาซี
- ก. ส่วนหัวของโปรแกรม (Preprocessor Directive)
 - ข. ส่วนฟังก์ชันหลัก (Main Function)
 - ค. ส่วนของคำอธิบายโปรแกรม (Program Comment)
 - ง. ส่วนคอมไพเลอร์ (Compiler)

