

ใบปฏิบัติการการเขียนโปรแกรมภาษาซี

การทดลองที่ 1

การเขียนโปรแกรมภาษาซีเบื้องต้น

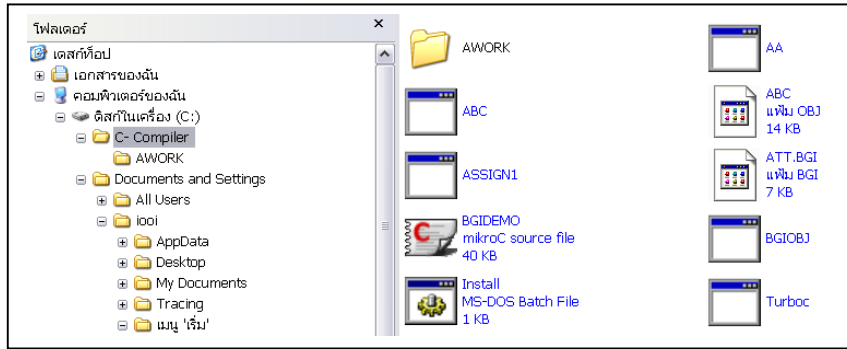
วัตถุประสงค์

1. เพื่อให้ผู้อ่านสามารถติดตั้งโปรแกรมภาษาซีได้
2. เพื่อให้ผู้อ่านสามารถใช้งานภาษาซีได้
3. เพื่อให้ผู้อ่านสามารถพิมพ์โปรแกรมตามที่กำหนดได้

ความรู้พื้นฐาน

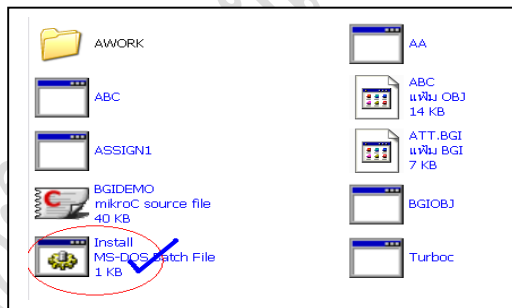
การติดตั้งโปรแกรม Turbo C

- ทำการสำเนาเพิ่มข้อมูลตัวโปรแกรมเทอร์โบซี ของ **Borland Turbo C** (Dennis R., 1971). มาเก็บไว้ใน Drive C: ใน Folder C-Compiler แสดงรูปที่ Lab1.1



รูปที่ Lab1.1 แสดงที่อยู่โปรแกรมภาษาซี

- ทำการสั่งติดตั้ง โปรแกรมภาษาซี โดย ดับเบิลคลิกที่แฟ้มข้อมูล Install
รูปที่ Lab1.2



รูปที่ Lab1.2 แสดงแฟ้มข้อมูล Install สำหรับการติดตั้งโปรแกรม

- โปรแกรมภาษาซีจะถูกติดตั้งลงในเครื่องคอมพิวเตอร์ใน Driver C:
รูปที่ Lab1.3

```

Command Prompt - install
Inflating: C:\TC\BIN\ABC.BAK
Inflating: C:\TC\BIN\ASSIGN1.CPP
Inflating: C:\TC\BIN\ASSIGN1.EXE
Inflating: C:\TC\BIN\ASSIGN1.OBJ
Inflating: C:\TC\BIN\ASSIGN1.BAK
Inflating: C:\TC\BIN\BBB
Inflating: C:\TC\BIN\BUILTINS.MAK
Inflating: C:\TC\BIN\CCC.TXT
Inflating: C:\TC\BIN\CH24_2.BAK
Inflating: C:\TC\BIN\CH24_2.OBJ
Inflating: C:\TC\BIN\CH24_2.CPP
Inflating: C:\TC\BIN\CH24_25.C
Inflating: C:\TC\BIN\CHKLIST.PS
Inflating: C:\TC\BIN\CPP.EXE
Inflating: C:\TC\BIN\DPMI16BI.OWL
Inflating: C:\TC\BIN\DPMIINST.EXE
Inflating: C:\TC\BIN\DPMILOAD.EXE
Inflating: C:\TC\BIN\DPMI.MEM.DLL
Inflating: C:\TC\BIN\DPMIRES.EXE
Inflating: C:\TC\BIN\EMSTEST.COM
Inflating: C:\TC\BIN\EX1.EXE
Inflating: C:\TC\BIN\EX1.OBJ
Inflating: C:\TC\BIN\FFF.TXT
Inflating: C:\TC\BIN\GGG.TXT
Inflating: C:\TC\BIN\GREF.COM

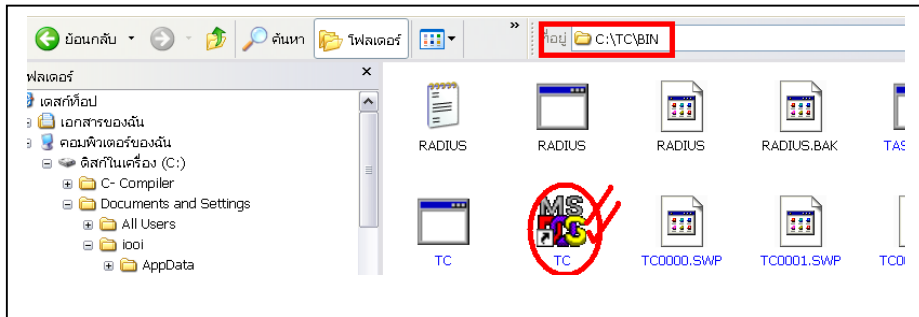
```

รูปที่ Lab1.3 แสดงการติดตั้งโปรแกรมภาษาซีลงใน C:\TC\BIN

การเรียกโปรแกรม Turbo C มาใช้งาน

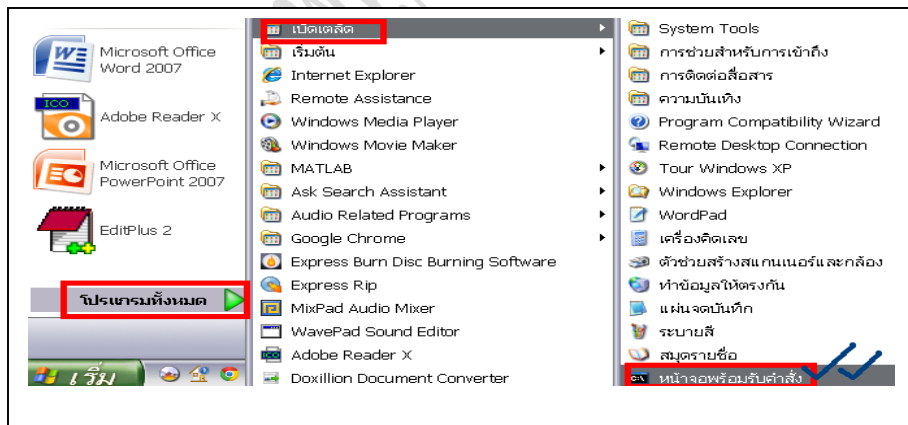
การเรียกโปรแกรมภาษาซีมาใช้งานสามารถทำได้ 2 วิธีคือ สามารถเรียกผ่านทางระบบปฏิบัติการ windows และเรียกผ่านระบบ Command Line

- การเรียกโปรแกรมภาษาซีโดยผ่านทางระบบปฏิบัติการ windows สามารถทำได้โดยทำการย้าย Folder มาที่ C:\TC\BIN และทำการ ดับเบิ้ลคลิกที่เพิ่มข้อมูล TC รูปที่ Lab1.4



รูปที่ Lab1.4 แสดงการเรียกโปรแกรมภาษาซีมาใช้งานด้วยระบบ windows

- การเรียกโปรแกรมภาษาซีโดยเรียกผ่านระบบ Command Line สามารถทำได้โดยการย้ายระบบการสั่งการไปยัง ระบบปฏิบัติการ DOS ทำการคลิก รูปที่ Lab1.5



รูปที่ Lab1.5 แสดงการเรียกระบบการสั่งการด้วยระบบปฏิบัติการ DOS

- ทำการย้าย Directory ไปยัง C:\TC\BIN ด้วยคำสั่ง `cd\tc\bin` และทำการกดคีย์ Enter เมื่อย้ายได้สำเร็จให้ทำการเรียกโปรแกรมภาษาซี ด้วยการพิมพ์คำสั่ง `tc` และกด Enter รูปที่ Lab1.6

```

c:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\iooi>cd\tc\bin
C:\TC\BIN>tc
  
```

รูปที่ Lab1.6 แสดงการเรียกโปรแกรมภาษาซีโดยเรียกผ่านระบบ DOS

- ได้ผลการเรียกโปรแกรมภาษาซีได้ รูปที่ Lab1.7

```

c:\ Command Prompt - tc
- File Edit Search Run Compile Debug Project Options Window Help
CH24_2.CPP 2
CH24_25.C 3
===== Program Description =====
/* program name : ch24_25.c */
/* getpixel() application. */
#include <graphics.h>

void main()
{
    int driver = DETECT, mode;
    int i;

    initgraph(&driver, &mode, "c:\\windows\\system32\\");
    line(100, 100, 500, 100);
    for ( i = 20; i < 300; i++ )
        if ( getpixel(300, i) == WHITE )
        {
            setcolor(BLUE);
            circle(300, i, 5);
        }
    else
    12:35
  
```

รูปที่ Lab1.7 แสดงโปรแกรมภาษาซี

การทดลองการเขียนโปรแกรมภาษาซี

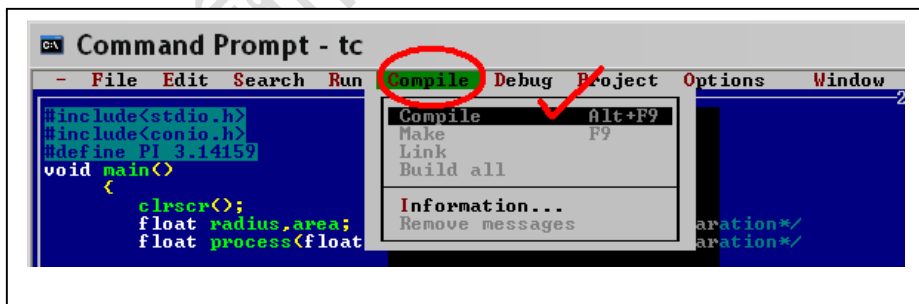
ในการเขียนโปรแกรมด้านล่างมุ่งเน้นให้ผู้อ่านทราบถึงหลักการการเขียนโปรแกรมภาษาซีเท่านั้น ผู้อ่านไม่ต้องคำนึงถึงไวยากรณ์ (จะกล่าวในภายหลัง) จากโปรแกรมด้านล่าง ให้ผู้อ่านฝึกทักษะในการเขียนโปรแกรมตามโครงสร้างของภาษาซีให้มีความเข้าใจในเบื้องต้นเป็นสำคัญก่อน โปรแกรมด้านล่างเป็นโปรแกรมการคำนวณหาพื้นที่วงกลมที่สามารถเขียนได้หลายรูปแบบ

ตัวอย่าง 1.3 โปรแกรมคำนวณหาพื้นที่วงกลมรูปแบบหนึ่ง ให้ผู้อ่านพิมพ์โปรแกรมลงในโปรแกรมภาษาซี

```
#include<stdio.h>
#include<conio.h>
#define PI 3.14159
void main()
{
    clrscr();
    float radius,area;           /*variable declaration*/
    float process(float radius); /*function declaration*/
    printf(" Radius = ? ");
    scanf("%f",&radius);
    area = process(radius);
    printf(" Area = %f",area);
    getch();
}
```

```
// ต่อจากด้านบน
float process(float r)           //function declaration
{
    float x;                    //local variable declaration
    x = PI * r * r;
    return (x);
}
```

การแปลภาษา (Compiler) เมื่อผู้อ่านพิมพ์โปรแกรมจากตัวอย่างที่ 1.3 แล้วให้ผู้อ่านทำการสั่งให้โปรแกรมภาษาซีทำการแปลภาษา โปรแกรมที่เขียนและตรวจสอบความถูกต้องและทำการแก้ไขข้อผิดพลาดตามการแสดงผล Error โดยเลือกคำสั่งรูปที่ Lab1.8



รูปที่ Lab1.8 แสดงการสั่งแปลภาษาโปรแกรมที่สร้างขึ้น

โดยผลการทำการแปลภาษาซี ถ้าไม่มีข้อผิดพลาดตัวคอมไพเลอร์จะแสดงกล่องข้อความ (Dialog) แจ้งว่าผลของ Warnings เป็น 0 และผลการตรวจสอบความผิดพลาดหรือ errors เป็น 0 รูปที่ Lab1.9

```

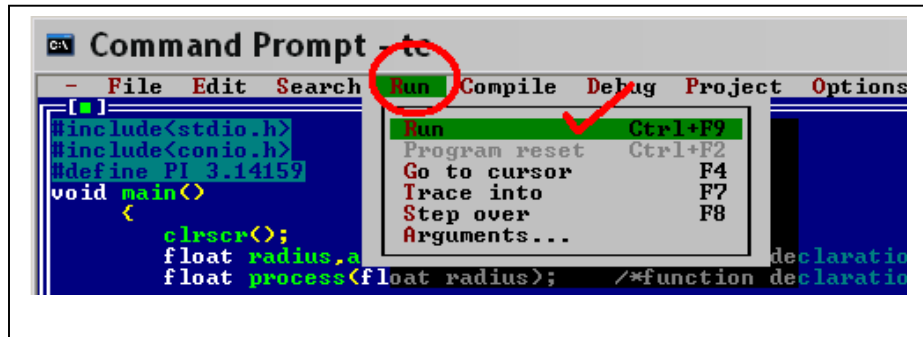
Compiling
Main file: GIRCLE1.CPP
Compiling: EDITOR  GIRCLE1.CPP

          Total      File
Lines compiled: 475  475
Warnings: 0         0
Errors: 0           0

Available memory: 1945K
Success : Press any key
  
```

รูปที่ Lab1.9 แสดงผลการแปลผลของโปรแกรม

การสั่งโปรแกรมภาษาซีทำงาน (RUN) เมื่อผลการแปลภาษาไม่มีข้อผิดพลาดใดๆ ขั้นตอนต่อไปเป็นการสั่งให้โปรแกรมภาษาซีทำงาน โดยแสดงผลการทำงานตามที่โปรแกรมเขียน โปรแกรมได้สร้างขึ้น โดยการสั่งงาน รูปที่ Lab1.10



รูปที่ Lab1.10 แสดงขั้นตอนการสั่งให้โปรแกรมทำงาน

จากตัวอย่างโปรแกรม 1.3 เป็นโปรแกรมที่ผู้เขียนโปรแกรมได้กำหนดค่าคงที่สัญลักษณ์ (symbolic constant) ของ PI ให้มีค่าเท่ากับ 3.14159 ไว้ในส่วนของ หัวโปรแกรม (header) เมื่อมีการเรียกใช้ PI ตัวแปลโปรแกรมจะทราบว่ามีการเรียกค่า 3.14159 มาทำการคำนวณ ในโปรแกรมนี้มีการกำหนดคำอธิบาย (comment) ที่ให้ผู้อ่านโปรแกรมเข้าใจวัตถุประสงค์ของการเขียนคำสั่งเพื่ออะไร ซึ่งมีการใช้ทาสองลักษณะคือ /*คำอธิบาย*/ และอีกรูปแบบใช้เครื่องหมาย //คำอธิบาย ในโปรแกรมนี้มีการเขียนโปรแกรมเป็นแบบโครงสร้างคือ มีการนิยามโปรแกรมย่อย (define sub-program) หรือการกำหนดฟังก์ชัน (define function) ที่มีชื่อว่า process มาใช้งานร่วมกับโปรแกรมหลักหรือฟังก์ชันหลักที่มีชื่อว่า void main โดยฟังก์ชัน process มีการกำหนดอาร์กิวเมนต์ หรือ พามามีเตอร์ (Argument : parameter) ชื่อ r ทำหน้าที่รับค่ารัศมีจากการเรียกใช้ฟังก์ชันด้วยคำสั่ง $area = process(radius)$ และในฟังก์ชัน process มีการคือค่าที่ได้ทำการคำนวณแล้วตัวแปร x มาให้ฟังก์ชันหลักนำค่ามาแสดงผลการทำงานของโปรแกรม รูปที่ Lab1.11



```

Output
Radius = ? 12
Area = 452.388947_

```

รูปที่ Lab1.11 แสดงผลการทำงานโปรแกรมคำนวณหาพื้นที่วงกลม

จากผลการทำงานของโปรแกรมจะแสดง Radius = ? เป็นการรอรับค่าจากผู้ใช้งาน โปรแกรมในที่นี่ได้ทำการกรอกค่ารัศมีที่ใช้ในการคำนวณเป็น 12 ทำให้ผลการคำนวณพื้นที่วงกลมมีผลลัพธ์เป็น Area = 452.388947

ให้ผู้อ่านทำการพิมพ์โปรแกรมตามตัวอย่างดังตัวอย่างที่ 1.4 แล้วทำการสั่งแปลภาษาโปรแกรม พร้อมทั้งสั่งให้โปรแกรมทำงาน เพื่อนำผลการทำงานของโปรแกรมทำการบันทึกผล ลงในหัวข้อผลการทดลองซึ่งเป็นโปรแกรมการคำนวณหาพื้นที่วงกลม โดยโปรแกรมนี้ผู้ใช้งานโปรแกรมสามารถกำหนดจำนวนครั้งในการทำงานของโปรแกรมได้

ตัวอย่างที่ 1.4 โปรแกรมคำนวณพื้นที่วงกลม สามารถให้คำนวณตามความต้องการ
ของผู้ใช้งาน โปรแกรม

```
#include<stdio.h>
#include<conio.h>
#define PI 3.14159
void main()
{
    clrscr();

    float radius,area;           /*variable declaration*/
    int count, n;                /*variable declaration*/
    float process(float radius); /*function declaration*/
    printf("\nHow many circles ?");
    scanf("%d",&n);
    for (count = 1; count<= n;count++)
    {
        printf("Circles no. %d: Radius = ? ",count);
        scanf("%f",&radius);
        area = process(radius);
        printf(" Area = %f",area);
        getch();
    }
}
```

ต่อจากด้านบน

```
float process(float r)                //function declaration
{
    float x;                          //local variable declaration
    x = PI * r * r;
    return (x);
}
```

บันทึกผลการทดลอง ให้ผู้อ่านเขียนผลการทำงานของโปรแกรม พร้อมอธิบายในการกรอกข้อมูลให้โปรแกรมตามตัวอย่าง 1.4 นำค่าไปคำนวณ

สรุปผลการทดลอง ให้ผู้อ่านทำการสรุปผลการทำงานของโปรแกรมว่ามีข้อดี/ข้อเสีย หรือสามารถนำผลของตัวอย่างโปรแกรมไปใช้ประโยชน์ด้านใด ประยุกต์พัฒนาโปรแกรมต่อด้านใดได้บ้าง

อุปกรณ์

1. เครื่องคอมพิวเตอร์
2. ระบบอินเทอร์เน็ต สำหรับการค้นคว้า
3. ตัวโปรแกรมภาษาซี
4. ตัวโปรแกรม CodeBlock
5. หน่วยความจำเคลื่อนที่ (Handy Drive, External Drive)

วิธีการทดลอง

1. ให้ผู้อ่านดำเนินการติดตั้งโปรแกรมที่ใช้ในการเขียนโปรแกรมภาษาซี
 - 1.1 สามารถติดตั้งโปรแกรมภาษาซีจาก Turbo C ตามขั้นตอนภาคผนวก ก. หรือ
 - 1.2 สามารถติดตั้งโปรแกรมภาษาซีจากโปรแกรม CodeBlock ตามขั้นตอน ภาคผนวก ข.
2. ให้ผู้อ่านฝึกทดสอบการใช้ เมนุรายการคำสั่งของโปรแกรมที่ผู้อ่านได้เลือกทำการติดตั้งในเครื่องคอมพิวเตอร์
 - 2.1 ทดสอบการเขียนโค้ดสั้น ๆ
 - 2.2 ทดสอบการสั่งให้โปรแกรมทำการแปลคำสั่ง (Compiler)
 - 2.3 ทดลองทำการสั่งให้โปรแกรมเริ่มทำงาน (Run)
3. ให้ผู้อ่านทดสอบหาจุดผิดพลาดของโปรแกรมและทำการแก้ไขจุดผิดพลาดโดยยึดทฤษฎีมาเป็นหลักในการแก้ปัญหา
4. ให้ผู้อ่านได้ทดลองทำการป้อนตามที่เนื้อหาได้กำหนด ทั้งส่วนทฤษฎีและการทดลอง ให้ผลการทำงานโปรแกรม ตรงตามที่เนื้อหาที่กำหนด
5. ให้ผู้อ่านได้ทำการเขียนผลการทดลองตามกำหนด