



โครงสร้างภาษาซีเบื้องต้น



วัตถุประสงค์การเรียนรู้



- อธิบายโครงสร้างภาษาซีได้
- เขียนโปรแกรมภาษาซีเบื้องต้นได้
- ใช้ฟังก์ชันมาตรฐานในการรับและแสดงข้อมูลเบื้องต้นได้
- สามารถประกาศตัวแปรและใช้งานตัวแปรได้



โครงสร้างโปรแกรม



- Preprocessor directives
- Global declarations
- The main() function
- Uses-defined function
- Program comments



Preprocessor directives



- ใช้สำหรับเรียกไฟล์ที่โปรแกรมต้องการใช้ในการทำงาน
- คอมไพเลอร์จะกระทำตามคำสั่งก่อนที่จะคอมไพล์โปรแกรม
- # คือเครื่องหมาย directives

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

```
รูปแบบ #define name value  
ตัวอย่าง #define a 100;
```



Global declarations



- ส่วนที่ใช้ประกาศตัวแปรที่ต้องใช้ในโปรแกรม
- ทุกส่วนของโปรแกรมสามารถเรียกใช้งานตัวแปรนี้ได้
- ส่วนนี้อาจ มี หรือ ไม่มีก็ได้



Main () function



- ส่วนนี้ทุกโปรแกรมจะต้องมี
- แต่ละประโยคคำสั่งจะจบด้วยเครื่องหมาย (; เซมิโคลอน)
- เริ่มต้นด้วย main()
- ตามด้วยเครื่องหมาย ({ } ปิดกาเปิดและปิด)



Uses-defined functions



- การกำหนดฟังก์ชันขึ้นมาใช้งานเอง
- ต้องอยู่ในเครื่องหมายปีกกา
- สามารถให้โปรแกรมหลักเรียกใช้งานได้

```
#include<stdio.h>
main()
{
    function();
}
```

```
function()
{
    return
}
```



Program comments



- เป็นส่วนอธิบายโปรแกรม
- ทำให้อ่านโปรแกรมได้ง่ายยิ่งขึ้น
- จะไม่ถูกคอมไพเลอร์นำไปประมวลผล



โปรแกรมภาษาซีเบื้องต้น



- ส่วนหัวคือส่วนเรียกโมดูลอื่น โดยโมดูลเหล่านี้จะเก็บคำสั่งพื้นฐาน
- ส่วนฟังก์ชันหลัก คือ เป็นส่วนของคำสั่ง

```
#include<stdio.h>
main()
{
    .....
}
```

```
#include<stdio.h>
main()
{
    printf("COMPUTER");
}
```



โปรแกรมภาษาซีเบื้องต้น



- คำอธิบายโปรแกรม

```
/* ..... */
```

```
/* program by Komvech dangprapai  
#include<stdio.h>  
main()  
{  
    printf("COMPUTER");  
    printf("KMITL");  
}
```



โปรแกรมภาษาซีเบื้องต้น



- การขึ้นบรรทัดใหม่

```
/* program by Komvech dangprapai
#include<stdio.h>
main()
{
    printf("COMPUTER\n");
    printf("KMITL\n");
}
```





```
#include<stdio.h>
int feet, inches;
main()
{
    feet = 6;
    inches = feet * 12;
    printf("Height in inches is %d",inches)
}
```



ผลลัพธ์ Height in inches is 72



คำสั่ง printf



```
printf("Hello");
```



Hello

```
X=7;  
printf("%d\n",x);  
printf("%d\n",x+9);
```



7
16

```
Yard =8;  
Feet = yards * 3;  
printf("%d yards is",yard);
```



8 yard is 24 feet



คำสั่ง printf



```
printf("Yes \n");  
printf("No \n");
```



Yes
No

```
printf("Yes \n");  
printf("\n");  
printf("No \n");
```



Yes

No

```
printf("Yes");  
printf("No");
```



YesNo



คำสั่ง printf



```
#include<stdio.h>
int nickels, dimes, TotCenter;
main()
{
    nickels = 3;
    dimes = 7;
    TotCenter = (nickels * 5) + (dimes*10);
    printf(“%d nickels and %d dimes \n”,nickels,dimes)
    printf(“= %d cents \n”,TotCenter);
}
```



3 nickels and 7 dimes
= 85 cents



การเก็บข้อมูลของตัวแปร



```
#include<stdio.h>
int x;
main()
{
    x = 6;
    x = 8;
    printf("X is %d",x);
}
```



6	✘
8	



การเก็บข้อมูลของตัวแปร



```
num1 = 5;  
num2 = 14;  
num1 = num2;  
printf("%d",num1);  
printf("%d",num2);
```



14

14



การเก็บข้อมูลของตัวแปร



$count = count + 1$ {เพิ่มค่าในตัวแปร **count** ขึ้นหนึ่ง}

$sum = sum + x$ {นำค่าใน **sum** บวกกับ **x** โดยค่าใน **x** จะไม่เปลี่ยนแปลง}

$num = 3 * num$ {นำค่าใน **num** คูณกับ **3** และเก็บไว้ที่เดิม}



การรับและพิมพ์ข้อมูลเบื้องต้น (printf)



```
printf("ข้อความ"); หรือ  
printf("รหัสควบคุมรูปแบบ", ตัวแปร) หรือ  
printf("Control string", variable list...);
```



การรับและพิมพ์ข้อมูลเบื้องต้น (printf)



```
#include<stdio.h>
main()
{
    printf("COMPUTER\n");
    printf("\n\nCOMPUTER");
    printf("%d\n",20);
}
```



รหัสแบ็คสแล็ช



\n

ให้ขึ้นบรรทัดใหม่

\t

ให้เว้นแท็บ เป็นระยะ 8 ช่วง

\xhh

แสดงตัวอักษรเมื่อ hh เป็นเลขฐาน 16

\a

ส่งเสียงบีป

\\

เครื่องหมาย \\



Format code



- %d** ให้พิมพ์เลขจำนวนเต็มฐานสิบ
- %u** ให้พิมพ์เลขจำนวนเต็มไม่มีเครื่องหมาย
- %f** ให้พิมพ์เลขทศนิยม
- %e** ให้พิมพ์เลขจำนวนจริงแบบยกกำลัง
- %c** ให้พิมพ์อักขรตัวเดียว
- %s** ให้พิมพ์ชุดตัวอักษร (ข้อความ)
- %%** ให้พิมพ์เครื่องหมาย %
- %o** ให้พิมพ์เลขฐานแปด
- %x** ให้พิมพ์เลขฐานสิบหก



ฟังก์ชัน printf



```
printf(“%s %d %f %c \n”, “Sam”, 14, -8.76, 'x');
```



Sam 14 -8.760000 X

```
printf(“%f %.3f %.2f  
%.1f”, 4.5678, 4.5678, 4.5678, 4.5678
```



4.567800 4.568 4.57 4.6



printf



```
#include<stdio.h>
main()
{
    int x,y;
    x =5;
    y =6;
    printf(“%d\n”,x);
    printf(“%c\n”,x);
    printf(“%d %d\n”,x,y);
    printf(“%d\n”,125);
```

```
printf(“%c\n”,125);
printf(“The total is $ %6.2f\n”,12.5);
printf(“The total is $ %6.3f\n”,12.5);
}
```



scanf



รูปแบบ

```
scanf("Control string", &variable list,.....); หรือ  
scanf("รหัสรับข้อมูล", &ตัวแปรเก็บข้อมูล);
```

ตัวอย่าง

```
int num;  
scanf("%d",&num);
```

ตัวอย่าง

```
int i, j;  
scanf("%o %x", &i, &j);  
printf("%o %x", i, j);
```

Interactive Program



โปรแกรมแปลงหน่วยฟุตเป็นหน่วยนิ้ว

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

```
main()
```

```
{
```

```
    int feet, inches;
```

```
    printf("Enter number of feet");
```

```
    scanf("%d", &feet);
```

```
    inches = feet * 12;
```

```
    printf("%d inches",inches);
```

```
}
```



Interactive Program



โปรแกรมคำนวณหาเส้นรอบวง

```
main()
#define TWOPI (3.1415926 * 2.0)
{
    double radius, circumf;
    printf("Enter radius:");
    scanf("%f",&radius);
    circumf = TWOPI * radius;
    printf("Circumference: %1.2f\n", circumf);
}
```



คำสั่งพื้นฐานการรับและแสดงผลข้อมูล



`putchar()` ใช้สำหรับแสดงตัวอักขระตัวเดียวออกทางหน้าจอ

```
putchar(ch);
```

```
char m='T';  
putchar(m);  
putchar('z');
```



คำสั่งพื้นฐานการรับและแสดงผลข้อมูล



`puts()` ใช้แสดงข้อความออกทางจอภาพ

```
put(str);
```

```
char name[10] = "Computer";  
puts(name);  
puts("THAI");
```



คำสั่งพื้นฐานการรับและแสดงผลข้อมูล



`getchar()` รับข้อมูลตัวอักขระหนึ่งตัวเมื่อป้อนข้อมูลแล้วต้องกด **Enter**



```
ch = getchar();
```

`getch();` รับข้อมูลตัวอักขระหนึ่งตัว แต่ไม่ต้องกดคีย์ **Enter**



```
ch = getch();
```



คำสั่งพื้นฐานการรับและแสดงผลข้อมูล



`get()` ใช้รับข้อความสตริง



```
gets(str);
```





Thank You!

