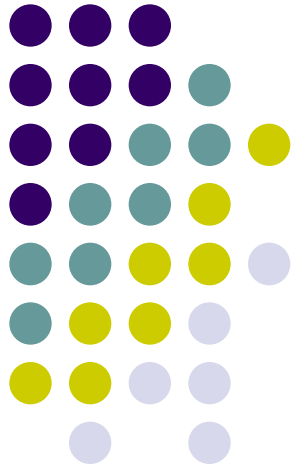
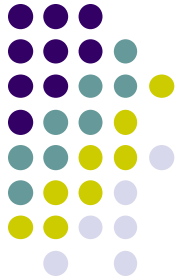
ฟังก์ชันในภาษา C

วิชา ภาษาซี : ครูไบนารี



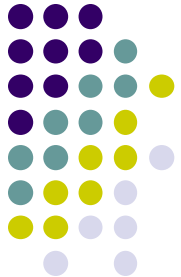
ฟังก์ชันในภาษา C



ในการทำงานบางอย่างจำเป็นต้องใช้คำสั่งมากกว่า 1 คำสั่งเพื่อทำงานนั้นให้สำเร็จ ซึ่งคำสั่งที่เขียนรวมกันไว้ใช้งานจะเรียกว่าฟังก์ชัน (Function)

ฟังก์ชัน (Function) คือ การเขียนคำสั่งรวมกันไว้เป็นกลุ่มของคำสั่งเพื่อทำงานให้สำเร็จ โดยกลุ่มของคำสั่งที่เราเขียนจะอยู่ภายในเครื่องหมาย { } เพื่อบอกขอบเขต และมีการตั้งชื่อให้กับกลุ่มคำสั่งนั้นเพื่อความสะดวกในการเรียกใช้งาน

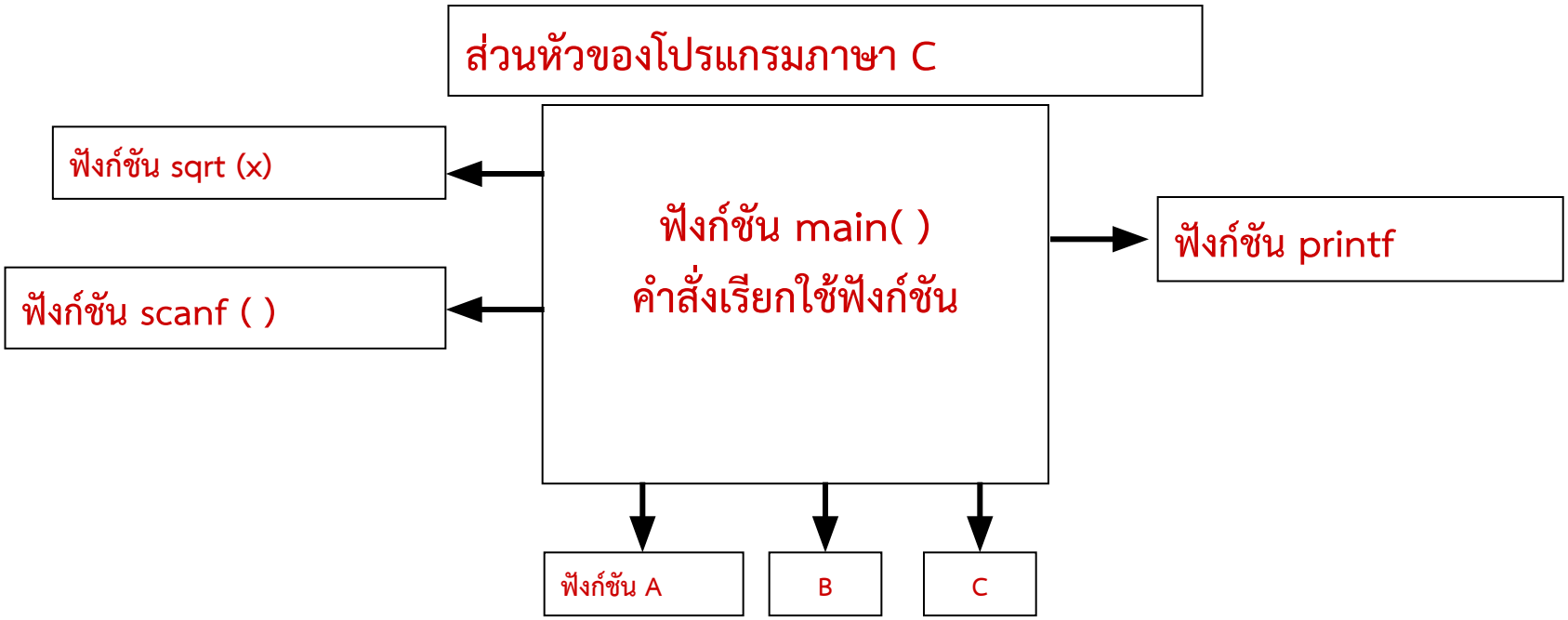
ฟังก์ชันในภาษา C

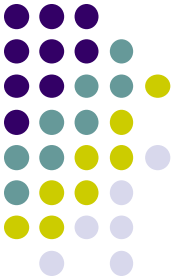


ข้อดีของการสร้างฟังก์ชันขึ้นมาใช้งาน คือ ถ้าเราต้องการทำงานที่ซ้ำซ้อน หรือทำงานใดซ้ำกันหลายครั้ง เช่น หากต้องการหาพื้นที่ของรูปสี่เหลี่ยมทั้งหมด 10 รูป เราต้องเขียนคำสั่งหาพื้นที่ทั้งหมด 10 ครั้ง ดังนั้นหากเราสร้างฟังก์ชันหาพื้นที่รูปสามเหลี่ยมก็จะสามารถเรียกใช้ฟังก์ชันดังกล่าวเมื่อใดก็ได้

- โครงสร้างของโปรแกรมที่เขียนด้วยภาษา C ภายในโปรแกรมจะประกอบด้วยฟังก์ชันต่างๆ อย่างน้อยหนึ่งฟังก์ชันเสมอนั่นคือ ฟังก์ชัน main() ซึ่งเป็นฟังก์ชันหลักที่โปรแกรมภาษา C จะเริ่มทำงานจากจุดนี้ โดยฟังก์ชัน main() อาจจะมีการใช้ฟังก์ชันย่อยๆ ต่อไปอีก

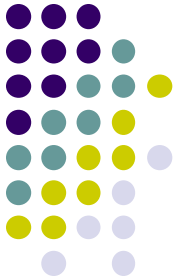
โครงสร้างโปรแกรมภาษา C





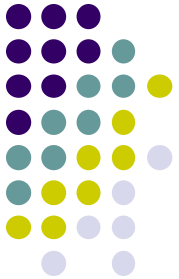
- ฟังก์ชัน ถือได้ว่าเป็นโปรแกรมย่อย (Sub Program)
- โดยจะทำงานเฉพาะอย่างของแต่ละฟังก์ชัน
- ภาษาซี แบ่งฟังก์ชันออกเป็น 2 ประเภท
  - ฟังก์ชันมาตรฐาน (Standard Function) โดยเรียกใช้จาก Library
  - ฟังก์ชันที่กำหนดขึ้นใช้เอง (User Defined Function)

ฟังก์ชันมาตรฐานหรือไลบรารีฟังก์ชัน



- เป็นฟังก์ชันที่มีมาให้พร้อมกับตัวแปลภาษา C เพื่อใช้งานได้ทันที และใช้ในงานด้านต่างๆ โดยเน้นงานพื้นฐาน เช่น ฟังก์ชันคำนวณทางคณิตศาสตร์ ฟังก์ชันสำหรับจัดการข้อความ ฟังก์ชันเวลา เป็นต้น เพื่อให้ผู้เขียนภาษา C มีความสะดวกมากขึ้น
- **ไลบรารีฟังก์ชันภาษา C จะเก็บอยู่ในไฟล์นามสกุล .h** หรือที่เรียกว่า header file ยกตัวอย่างเช่น ฟังก์ชันเกี่ยวกับการคำนวณจะเก็บอยู่ในไฟล์ชื่อ math.h หรือฟังก์ชันเกี่ยวกับการจัดการข้อความอยู่ในไฟล์ชื่อ string.h เป็นต้น

1. ฟังก์ชันมาตรฐานหรือไลบรารีฟังก์ชัน



- ในการเรียกใช้งานฟังก์ชันต้องเขียนรูปแบบการใช้คำสั่ง คือ

  #include<header file>

  และเขียนไว้ในส่วนหัวของโปรแกรม เพื่อให้ตัวแปลภาษา C เข้าใจว่าภายในโปรแกรมของเรามีการเรียกใช้ไลบรารีฟังก์ชัน

ฟังก์ชันมาตรฐาน (Standard Function)

ตัวอย่าง Library ต่างๆ ที่มีอยู่ในภาษาซี

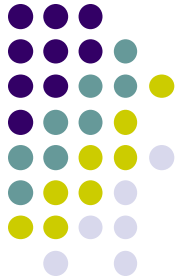
Library	Function	การทำงานของฟังก์ชัน
stdio.h	printf( )	ใช้ในการแสดงผลข้อมูล
	scanf( )	ใช้ในการรับข้อมูล
conio.h	getchar( )	ใช้ในการรับข้อมูล 1 อักขระ โดยการกด Enter
	getche( )	ใช้ในการรับข้อมูล 1 อักขระ โดยไม่ต้องกด Enter
	getch( )	ใช้ในการรับข้อมูล 1 อักขระ ไม่ปรากฏให้เห็นในการรับข้อมูล
	putchar( )	ใช้ในการแสดงอักขระ 1 อักขระ ออกทางจอภาพ
	clrscr( )	ใช้ในการลดจอภาพ
string.h	strlen( )	ใช้ในการนับความยาวของอักขระที่รับเข้ามา
	strcpy( )	ใช้ในการทำสำเนาข้อความ จากข้อความหนึ่งไปยังอีกข้อความหนึ่ง
	strcmp( )	ใช้ในการเปรียบเทียบข้อความ 2 ข้อความ
	strcat( )	ใช้ในการเชื่อมตั้งแต่ 2 ข้อความเข้าด้วยกัน

ฟังก์ชันมาตรฐาน (Standard Function)

ตัวอย่าง Library ต่างๆ ที่มีอยู่ในภาษาซี

Library	Function	รูปแบบ	การทำงานของฟังก์ชัน
math.h	sqrt( )	sqrt(parameter)	ใช้ในการหาค่าราก (root) ที่สองของเลขจำนวนเต็ม
	exp( )	exp(x)	ใช้ในการหาค่า e^x (Exponential)
	sin( )	sin(x)	เป็นฟังก์ชันหาค่า xy
	cos( )	cos(x)	เป็นฟังก์ชันหาค่า sine ของ x
	tan( )	tan(x)	เป็นฟังก์ชันหา tan ของ x
	log( )	log(n)	เป็นฟังก์ชันหาค่า log ของ x
	log10( )	log10(x)	เป็นฟังก์ชันหา log ฐาน 10 ของ x
	ceil( )	ceil(x)	เป็นฟังก์ชันหาค่าปัดเศษทศนิยมของตัวแปร x
	floor( )	floor(x)	เป็นฟังก์ชันหาค่าตัดเศษทศนิยมทิ้งของตัวแปร x
	fabs( )	fabs(x)	เป็นฟังก์ชันหาค่าสัมบูรณ์ (absolute value) ของ x

ฟังก์ชันสตริง

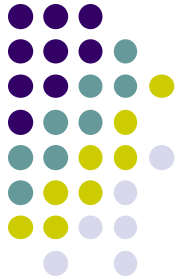


- **สตริง (string) หรืออะเรย์ตัวอักษร** คือ ข้อมูลที่ประกอบไปด้วยตัวอักษรที่มีการเรียงต่อเนื่องกันไป โดยมีจุดสิ้นสุดของข้อมูลสตริงที่ตัวอักษร NULL character เขียนด้วย ‘\0’
- ในภาษาซีรูปแบบข้อมูลประเภทสตริงไม่มีการกำหนดไว้ การประกาศตัวแปรแบบสตริงทำได้ 2 วิธี คือ ในรูปของอะเรย์ กับในรูปของพอยน์เตอร์
- **ตัวอย่างการประกาศตัวแปรสตริง**

  char p[9] = “ I think !” ;

p[0]	p[1]	p[2]	p[3]	p[4]	p[5]	p[6]	p[7]	p[8]
‘I’	‘ ’	‘t’	‘h’	‘i’	‘n’	‘k’	‘!’	‘\0’

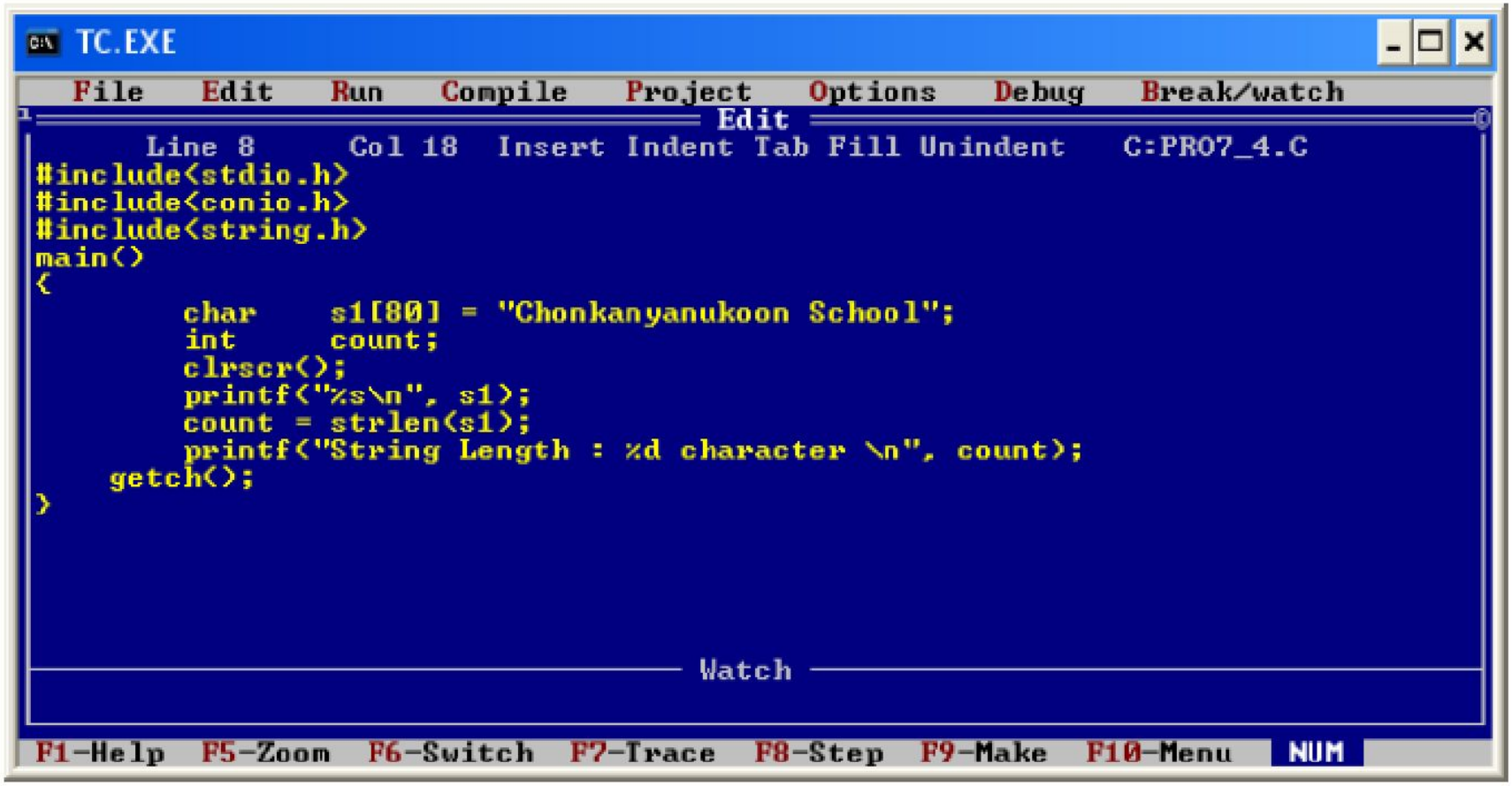
ฟังก์ชันสตริง

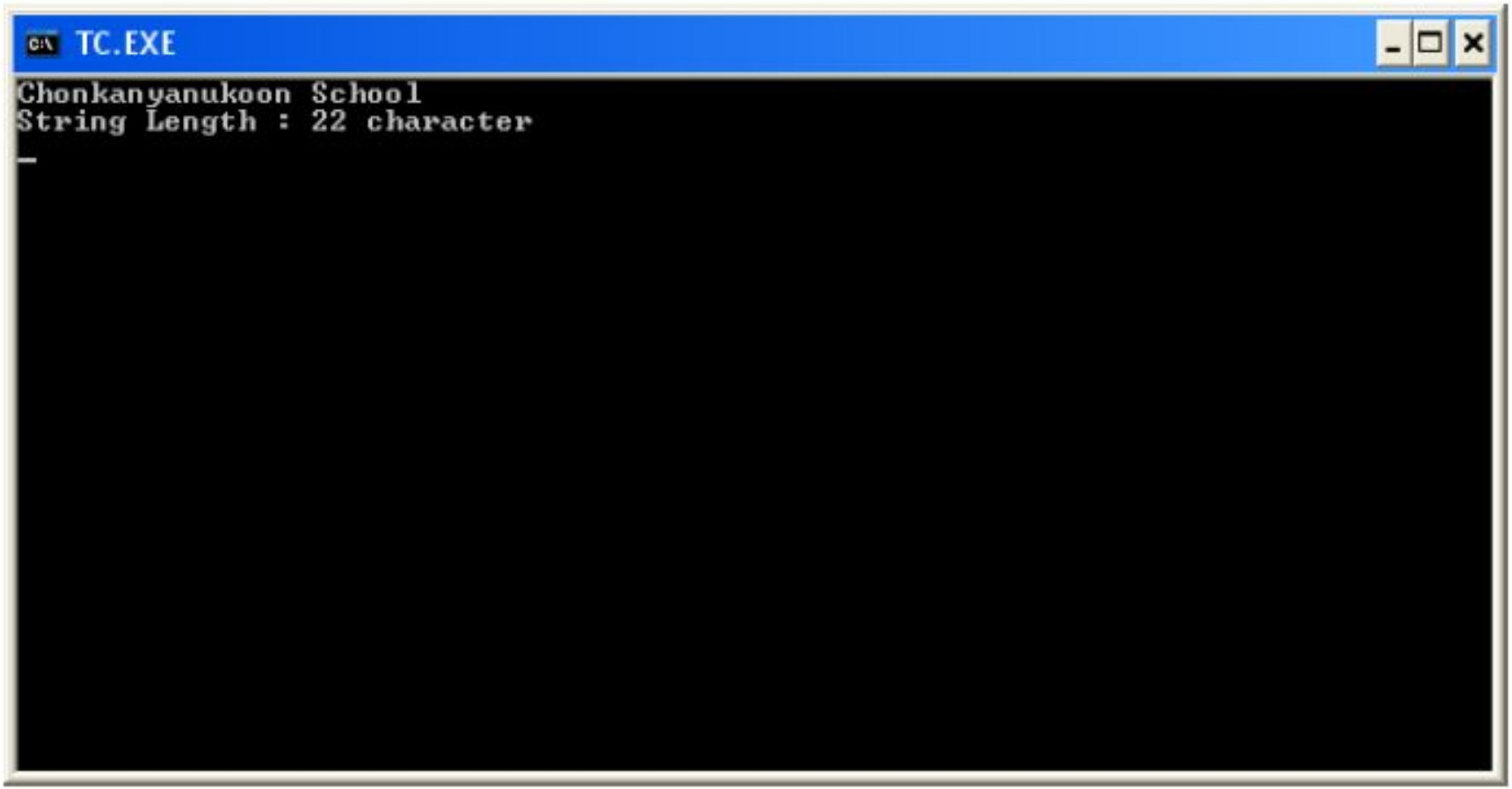


ฟังก์ชันมาตรฐานที่เกี่ยวกับสตริงที่ภาษาซีเตรียมไว้ให้เรียกใช้ ดังนี้

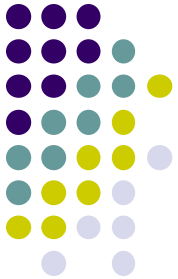
- gets( ) เป็นฟังก์ชันใช้รับค่าสตริง
- puts( ) เป็นฟังก์ชันที่ใช้แสดงผลสตริง
- strcat( ) เป็นฟังก์ชันที่ใช้ต่อสตริง 2 ตัวเข้าด้วยกัน
- strcmp( ) เป็นฟังก์ชันที่ใช้เปรียบเทียบสตริง 2 ตัว
- strcpy( ) เป็นฟังก์ชันที่ใช้ก๊อปปี้สตริง
- strlen( ) เป็นฟังก์ชันที่ใช้เพื่อหาความยาวของสตริง

ตัวอย่างโปรแกรมที่เรียกใช้ฟังก์ชันสตริง



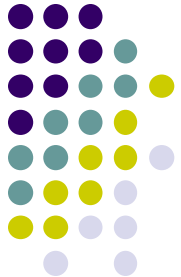

TC.EXE
Chonkanyanukoon School
String Length : 22 character

ตัวอย่างโปรแกรมที่เรียกใช้ฟังก์ชันสตริง



```
#include<stdio.h>
#include<string.h>
main( )
{     char s1[50] ={"C is easy "};
      char s2[50] = {"Uttaradit Rajabhat University"};
      char s3[50];
      int count,count2;
      count = strlen(s1); printf("String S1 Length = %d\n",count);
      strcat(s1,s2); printf("String s1 + String s2 = %s\n",s1);
      strcpy(s3,s2); printf("String s3 = %s\n",s3);
      count2=strcmp(s3,s2); printf("s3 like s2 = %d",count2)
}
```

แบบฝึกหัด



- สร้าง String ขึ้นมาสองตัว ชื่อ name ให้เก็บชื่อของนักศึกษา และ String ชื่อ surname เก็บนามสกุลของนักศึกษา จากนั้นให้นับตัวอักษรทั้งหมด
- ให้นำชื่อและนามสกุลของนักศึกษามาต่อกัน จากนั้นให้แสดงผล
- ให้เปรียบเทียบชื่อและนามสกุลของนักศึกษาว่ามีจำนวนตัวอักษรเท่ากันหรือไม่ โดยมีเงื่อนไขคือ
  - ถ้าชื่อและนามสกุลจำนวนเท่ากัน ให้แสดงคำว่า “Name equal Surname”
  - ถ้าชื่อยาวกว่านามสกุล ให้แสดงคำว่า “Name more than Surname”
  - ถ้านามสกุลยาวกว่าชื่อ ให้แสดงคำว่า “Name less than Surname”

*

แบบฝึกหัด

- เขียนโปรแกรมส่วนของการ Login เข้าสู่โปรแกรม โดยให้เขียนรับค่า Username และ Password โดยกำหนด username เป็น admin และ password เป็น admintest โดยถ้าถูกต้องให้แสดงชื่อนักศึกษาเป็นผู้เข้าใช้ระบบ

ตัวอย่างหน้าจอ

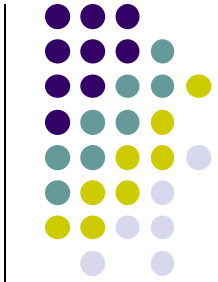
```
*** Please Login ***
Username :
Password :
------------------------
```

```
Hello!  Miss Chanida Kumpeng
Nice to meet you.
```

```
You not Miss Chanida Kumpeng.
```

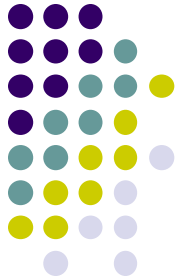
*

ตัวอย่างโปรแกรมที่เรียกใช้ฟังก์ชันคณิตศาสตร์



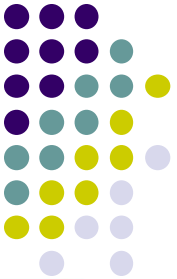
```
#include<stdio.h>
#include<math.h>
main( )
{     int x,y;
      printf(“Enter number :”);
      scanf(“%d”,&x);
      y = sqrt(x);
      printf(“square root of x = %d”,y);
}
```

2. ฟังก์ชันที่ผู้ใช้สร้างขึ้นเอง

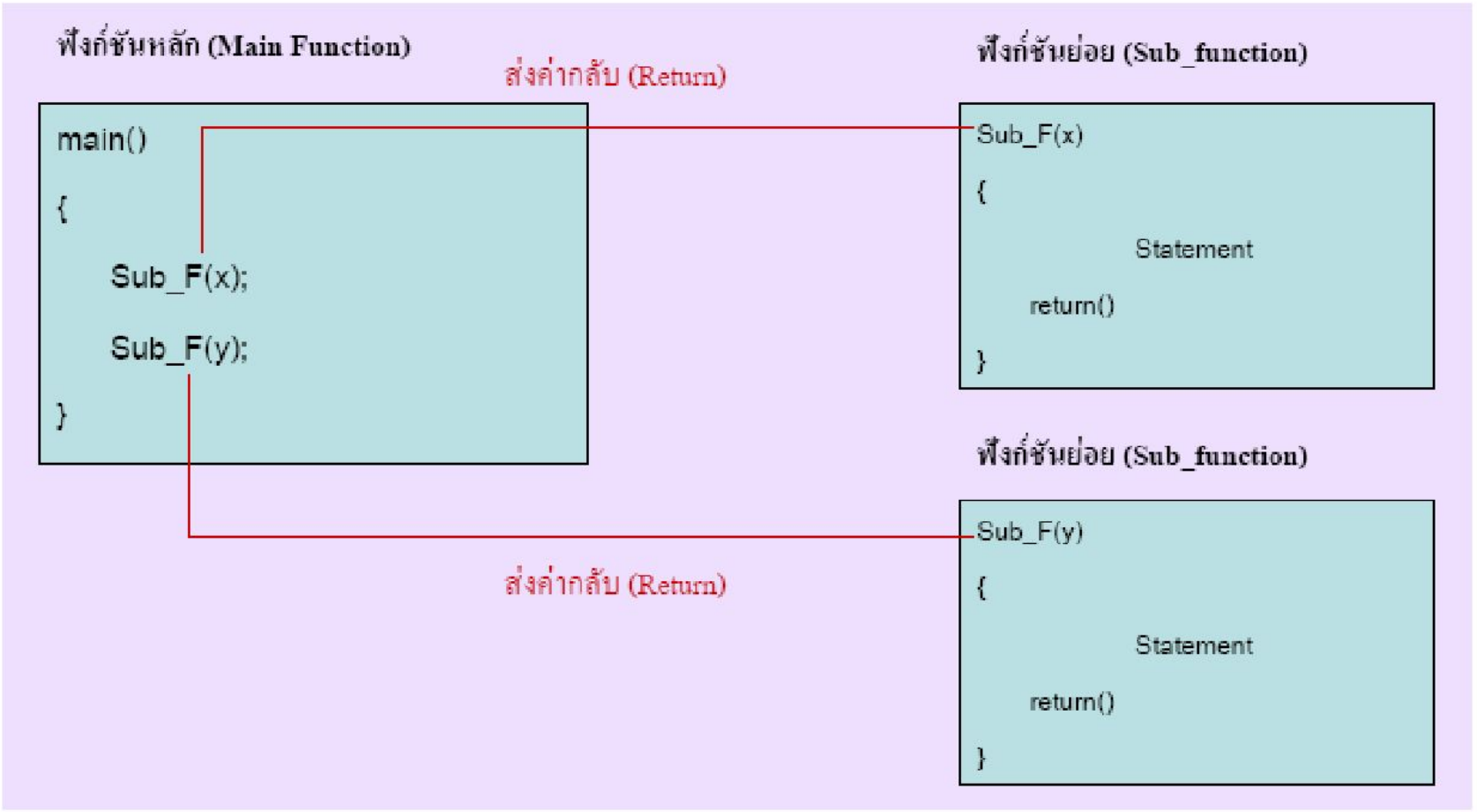


- ไลบรารีฟังก์ชันในภาษา C เป็นฟังก์ชันที่สร้างขึ้นเพื่อทำงานพื้นฐานทั่วไป ซึ่งบางครั้งอาจไม่มีไลบรารีฟังก์ชันที่ทำงานได้อย่างที่เราต้องการ ดังนั้น ภาษา C จึงให้เราสร้างฟังก์ชันขึ้นมาใช้งานได้เอง
- การสร้างฟังก์ชันขึ้นมาใช้งานโปรแกรม เป็นการแบ่งการทำงานเป็นส่วนเล็กๆ ทำให้ง่ายและสะดวกต่อการตรวจสอบและแก้ไขโปรแกรม โดยเฉพาะโปรแกรมที่มีขนาดใหญ่

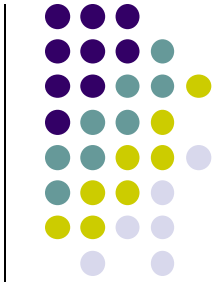
ฟังก์ชันที่ผู้ใช้สร้างขึ้นเอง



ฟังก์ชันหลัก (Main Function)

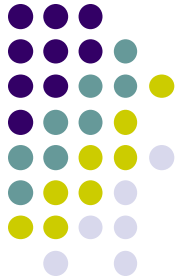


ทดลองเขียนฟังก์ชัน



```
#include<stdio.h>
    message()
    {
      printf("************ C Language *********** \n");
    }

    void main()
    {     printf("First Message\n");
      message();
      printf("Second Message\n");
      message();
      printf("Third Message\n");
      message();
    }
```

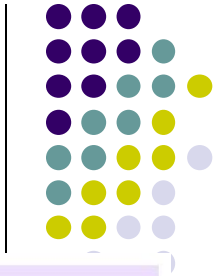
ประเภทของฟังก์ชัน



- สำหรับภาษา C ใช้การรับ/ส่งค่าของฟังก์ชันเป็นเกณฑ์ สามารถแบ่งได้ 3 ประเภทด้วยกัน ดังนี้

  1. ฟังก์ชันที่ไม่มีการรับ/ส่งค่า
  2. ฟังก์ชันที่มีการรับค่าเข้าไปในฟังก์ชัน
  3. ฟังก์ชันที่มีการส่งค่ากลับออกจากฟังก์ชัน

ฟังก์ชันที่ไม่มีการรับ/ส่งค่า



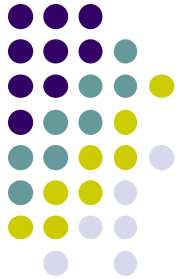
- ฟังก์ชันนี้ จะไม่มีการส่งผ่านค่าพารามิเตอร์ (parameter) ไปกลับ
- จะมีคำว่า void นำหน้าชื่อฟังก์ชัน
- การเรียกใช้ฟังก์ชันจะต้องมีการประกาศฟังก์ชันก่อน
- รูปแบบการประกาศฟังก์ชัน

```
void  ชื่อฟังก์ชัน ()
```

- รูปแบบฟังก์ชัน

```
void  ชื่อฟังก์ชัน () {
        statement
}
```

ฟังก์ชันที่ไม่มีการรับ/ส่งค่า



- โดยฟังก์ชันไม่มีการรับค่าใดๆ เข้าไปในฟังก์ชัน และฟังก์ชันก็ไม่มีผลลัพธ์ใดๆ ส่งออกมา เป็นฟังก์ชันที่ทำงานจบภายในฟังก์ชัน เช่น ฟังก์ชันการแสดงผล

```
void message( )
{
  printf(“C Programming \n”);
  printf(“is easy”);
}
```

กำหนดฟังก์ชันชื่อ message โดยไม่มีการรับค่า และไม่มีผลลัพธ์ส่งกลับ

ตัวอย่างโปรแกรมที่มีฟังก์ชันที่ไม่มีการส่ง/รับค่า

- Ex เป็นฟังก์ชันแสดงรายชื่อภาพยนตร์ที่กำลังเข้าฉาย โดยให้ผู้ใช้เลือกว่าต้องการตรวจสอบรายการภาพยนตร์หรือไม่

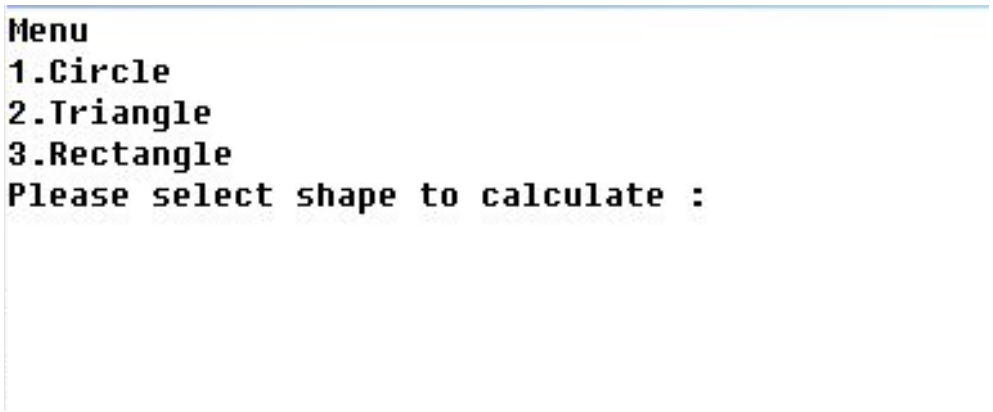
```
#include<stdio.h>
void movie_program()
{    printf("Now showing \n");
      printf("1.Spider man \n");
      printf("2.Harry Potter \n");
      printf("3.Resident Evil \n");
}

void main()
{ char ans;
      printf("Do you want to check movies program? (y/n)");
      scanf("%c",&ans);
      if(ans=='y')
        movie_program();
      else
        printf("Thank you");
}
```

แบบฝึกหัด

- สร้างเมนูและฟังก์ชันในการคำนวณพื้นที่ของรูปร่างต่อไปนี้
  - วงกลม
  - สามเหลี่ยม
  - สี่เหลี่ยม

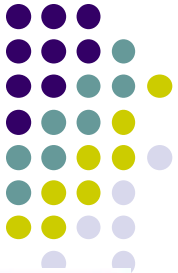
  โดยให้เลือกรูปร่างที่ต้องการคำนวณพื้นที่ จากนั้นให้นักศึกษาสร้างฟังก์ชันในการคำนวณพื้นที่ของแต่ละรูปร่าง

ตัวอย่างหน้าจอผลลัพธ์



*

ฟังก์ชันที่มีการรับค่า



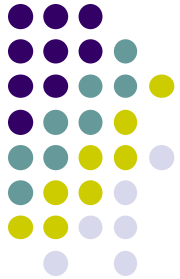
- ฟังก์ชันนี้ จะเป็นการส่งผ่านค่าพารามิเตอร์ (parameter) ให้แก่ฟังก์ชัน โดยไม่มีการส่งค่ากลับ
- จะมีคำว่า void นำหน้าชื่อฟังก์ชัน แล้วข้อความในวงเล็บจะประกอบด้วยตัวแปรและชนิดตัวแปร ถ้ามีตัวแปรหลายตัวให้คั่นด้วยเครื่องหมายคอมม่า , (Common)
- การเรียกใช้ฟังก์ชันจะต้องมีการประกาศฟังก์ชันก่อน
- รูปแบบการประกาศฟังก์ชัน

```
void  ชื่อฟังก์ชัน (ชนิดตัวแปร , ...)
```

- รูปแบบฟังก์ชัน

```
void  ชื่อฟังก์ชัน (ชนิดตัวแปร ตัวแปร, ...) {
        statement
}
```

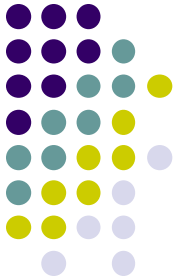
ฟังก์ชันที่มีการรับค่า



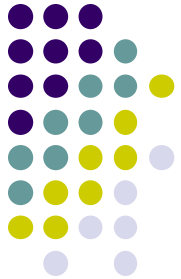
- ฟังก์ชันประเภทรับค่าอาร์กิวเมนต์ และต้องส่งอาร์กิวเมนต์ไปให้กับฟังก์ชันอื่นในการเรียกใช้งานด้วย ตัวอย่างเช่น ฟังก์ชันกำหนดอาร์กิวเมนต์ 3 ตัว เป็น int,int,char เวลาเรียกใช้งานจะต้องส่งค่า int,int และ char ไป 3 ค่าด้วย

```
void summary(int x,int y)
{
    int sum;
    sum = x+y;
    printf("Sum of 2 numbers = %d",sum);
}
```

ตัวอย่างโปรแกรม



```
#include<stdio.h>
void change_number(int num)
{
	printf("num = %d \n",num);
	num=num*2;
	printf("Values of Num before change = %d \n",num);
}

main()
{
	int number=1000;
	printf("Number before call function = %d \n",number);
	change_number(number);

}
```

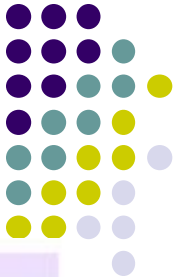


- void หมายความว่า “ไม่มีการคืนค่ากลับ” ฟังก์ชันที่มี void นำหน้าเป็นฟังก์ชันที่ทำงานทางเดียว คือ เมื่อเรียกใช้ฟังก์ชันจะทำงานตั้งแต่ต้นจนจบ จากนั้นจะกลับมายังจุดเดิมที่ถูกเรียกเพื่อให้ทำงานต่อไป

แบบฝึกหัด

- เขียนโปรแกรมคำนวณเกรดจากคะแนนสอบ โดยโปรแกรมจะรับค่าคะแนนมาจากผู้ใช้จากฟังก์ชัน main() และจะเรียกใช้ฟังก์ชันคำนวณเกรดโดยส่งค่าคะแนนไปในฟังก์ชัน โดยกำหนดฟังก์ชันคือ
  - ฟังก์ชัน gradeEng() ในการคำนวณเกรดคือ A,B,C,D และ F
  - ฟังก์ชัน gradeThai() ในการคำนวณเกรดคือ 4,3,2,1 และ 0
    - คะแนนสอบอยู่ในช่วง 90-100 เกรด A หรือ เกรด 4
    - คะแนนสอบอยู่ในช่วง 80-89 เกรด B หรือ เกรด 3
    - คะแนนสอบอยู่ในช่วง 70-79 เกรด C หรือ เกรด 2
    - คะแนนสอบอยู่ในช่วง 60-69 เกรด D หรือ เกรด 1
    - คะแนนน้อยกว่า 60 เกรด F หรือ เกรด 0

ฟังก์ชันที่มีการส่งค่ากลับออกจากฟังก์ชัน



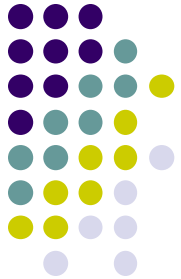
- ฟังก์ชันนี้ จะเป็นการส่งผ่านค่าพารามิเตอร์ (parameter) ไปกลับระหว่างฟังก์ชัน
- จะมีชนิดตัวแปรนำหน้าชื่อฟังก์ชัน แล้วตามด้วยชนิดตัวแปร และตัวแปร ถ้ามีหลายตัวแปร ต้องคั่นด้วยเครื่องหมายคอมม่า , (comma)
- รูปแบบการประกาศฟังก์ชัน

```
ชนิดตัวแปร  ชื่อฟังก์ชัน (ชนิดตัวแปร , ...)
```

- รูปแบบฟังก์ชัน

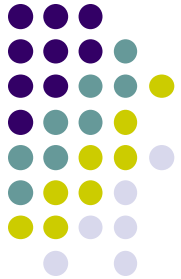
```
ชนิดตัวแปร  ชื่อฟังก์ชัน (ชนิดตัวแปร ตัวแปร, ...) {
        statement
}
```

ฟังก์ชันที่มีการส่งค่ากลับออกจากฟังก์ชัน



- เป็นฟังก์ชันที่ต้องส่งผลลัพธ์การใช้งานออกไปนอกฟังก์ชันด้วย ส่วนใหญ่จะเป็นฟังก์ชันที่ต้องคำนวณ โดยจะต้องสร้างตัวแปรเพื่อรอรับค่าผลลัพธ์จากการเรียกใช้ฟังก์ชันด้วย
- การสร้างฟังก์ชันแบบส่งค่ากลับออกจากฟังก์ชัน จะต้องอยู่ก่อนหน้า main() เพื่อให้ตัวแปลภาษา C เรียกฟังก์ชันเหล่านั้นทำงานได้อย่างถูกต้อง

ตัวอย่างฟังก์ชันที่มีการส่งค่ากลับออกจากฟังก์ชัน

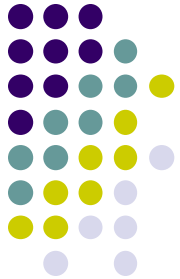


Ex ตัวอย่างโปรแกรมคำนวณเงินทอนลูกค้า โดยเมื่อลูกค้า

ซื้อสินค้าโปรแกรมจะทำการรับค่าราคาสินค้า และจำนวนเงินลูกค้า และสร้างฟังก์ชันคิดเงินทอนขึ้น

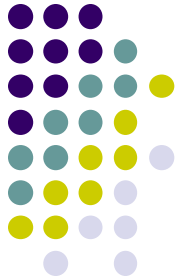
```
#include<stdio.h>
float cal_change(float price,float money)
{
	float z;
	z = money-price;
	return(z);
}
main()
{
	float price,money,change;
	printf("Enter product price :"); scanf("%f",&price);
	printf("Enter receive money :"); scanf("%f",&money);
	change = cal_change(price,money);
	printf("Customer change =%.2f \n",change);
}
```

ฟังก์ชันที่มีการส่งค่ากลับออกจากฟังก์ชัน



- จากฟังก์ชันตัวอย่าง cal_change() หน้าฟังก์ชันจาก void จะเปลี่ยนเป็นชนิดข้อมูลแทน คือฟังก์ชัน cal_change() มีข้อมูลชนิด float และมีการ return ค่าตัวแปร โดยจะตัดการแสดงผล(printf) ออกไป หมายความว่าฟังก์ชัน cal_change() จะทำการประมวลผลและส่งค่าผลลัพธ์กลับไปฟังก์ชัน main()
- จะเห็นว่าเวลาเรียกใช้ฟังก์ชันจะมีการส่งค่าไปตามปกติ และในขณะเดียวกัน เราก็จะให้มีการรับค่าที่คืนจากฟังก์ชันด้วย (return) โดยค่าที่คืนกลับมาคือค่าผลลัพธ์ที่ฟังก์ชันได้ประมวลผลนั่นเอง

แบบฝึกหัด



- เขียนโปรแกรมรับค่าสองค่าจากฟังก์ชัน main() และให้สร้างเมนูและฟังก์ชันในการคำนวณต่อไปนี้
  - ฟังก์ชัน Add() ในการบวกค่า
  - ฟังก์ชัน Difference() ในการลบค่า
  - ฟังก์ชัน Multiply() ในการคูณ
  - ฟังก์ชัน Division() ในการหาร

  โดยให้ส่งค่าผลลัพธ์การคำนวณกลับมาแสดงผลที่ฟังก์ชัน main()

  ตัวอย่างหน้าจอผลลัพธ์

```
Please Enter Number 1 : 25
Please Enter Number 2 : 20
Menu
1.Addition
2.Differnce
3.Multiply
4.Division
Please select Menu :
```