ครั้งที่ 2

การออกแบบโปรแกรมด้วยฟังก์ชันและคลาส

Top down design

- เทคนิคที่มีการแบ่งปัญหาออกเป็นปัญหาย่อยๆให้แยกออกจากกัน ลักษณะบนลงล่าง
- โดยโปรแกรมจะประกอบด้วยโปรแกรมย่อยๆต่างๆที่มีการปฏิสัมพันธ์ต่อกันเป็นลำดับชั้น
- ผลดีของการออกแบบเทคนิคนี้คือ ปัญหาได้ถูกแบ่งเป็นปัญหาย่อยๆ และโปรแกรมย่อย 1 โปรแกรมจะแก้ปัญหาย่อยได้เพียง 1 ปัญหา ทำให้ง่ายต่อความเข้าใจและการแก้ไข

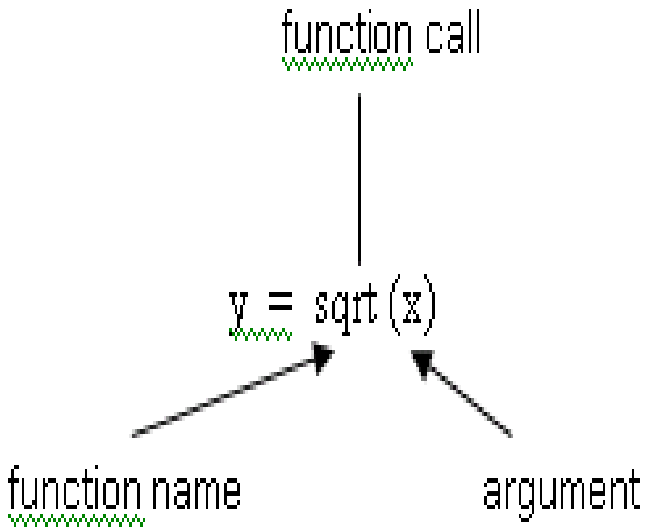
Function

Library Function

C++ ได้เตรียมคลาสที่กำหนดไว้(predefined class) และฟังก์ชันในไลบรารีมาตรฐาน อทิ เช่น ไลบรารีมาตรฐาน cmath

User defined Function

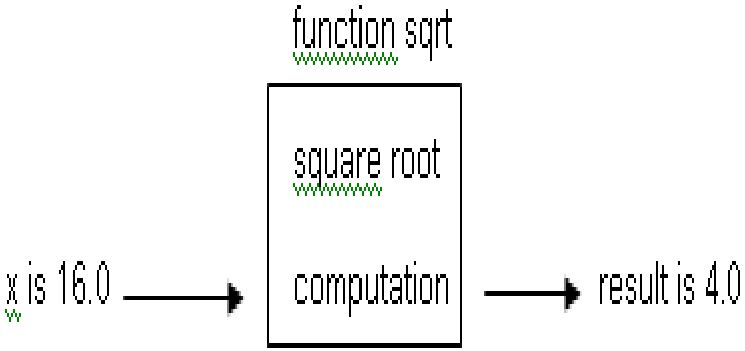
ไลบรารีมาตรฐาน cmath



- การทำงานจะมีการส่งผ่านอากิวเมนต์ x ไปยังฟังก์ชัน sqrt เมื่อถูกเรียกใช้ (function call)
- ถ้า x มีค่าเท่ากับ 16.0 การทำงานเป็นลำดับขั้นตอนดังนี้
- ดังนั้นฟังก์ชัน sqrt จะทำการคำนวณหาค่ารากที่สองของ 16.0 ซึ่งคือค่า 4.0
- ซึ่งผลลัพธ์ของฟังก์ชันมีค่าเท่ากับ 4.0 ซึ่งจะมีผลให้ y มีค่าเท่ากับ 4.0

การทำงานของฟังก์ชัน

- เรียกว่า"black box" กล่าวคือมีการส่งผ่านค่าซึ่งเป็นข้อมูลเข้าไปยังฟังก์ชัน และมีการส่งผ่านค่าผลลัพธ์กลับมาโดยอัตโนมัติ ซึ่งผู้เรียกใช้ไม่ต้องทราบว่ามีวิธีการหรือกระบวนการทำงานอย่างไร



```
/ / File: squareRoot.cpp
/ / Performs three square root computations
#include <cmath>          / / sqrt function
#include <iostream>       / / i/o functions
using namespace std;
int main ( )
{
   float first;           / / input : one of two data values
   float second;          / / input : second of two data values
   float answer:          / / output : a square root value
   / / Get first number and display its square root.
   cout << “Enter the first number:  “ ;
   cin >> first ;
   answer = sqrt ( first );
   cout  << “The square root of the first number is  “ << answer << endl ;
   / / Get second number and display its square root.
   cout << “Enter the second number :  “;
   cin >> second;
   answer = sqrt (second);
   cout << “The square root of the second number is  “  << answer  << endl;
   / / Display the square root of the sum of first and second.
   answer = sqrt ( first + second );
   cout << “The square root of the sum of both numbers is  “ << andwer  << endl;
   return 0;
}
```

```
Enter the first number: 9
The square root of the first number is 3
Enter the second number : 25
The square root of the second number is 5
The square root of the sum of both numbers is 5.83095
```

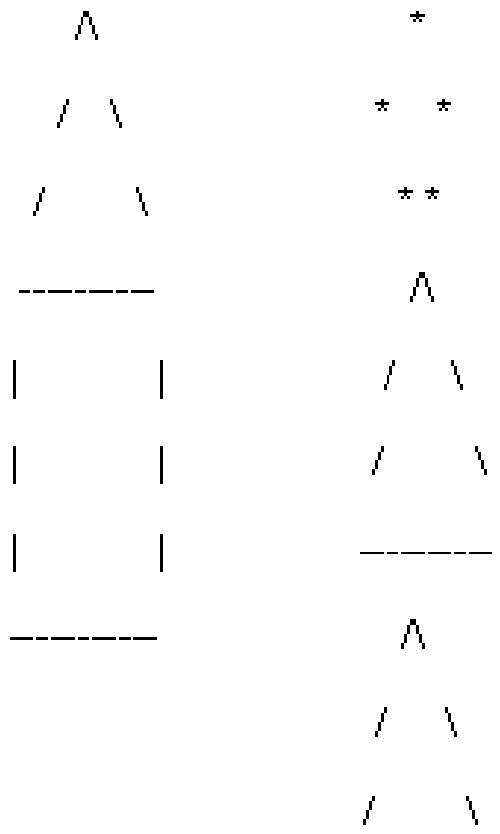
ตารางที่ 3.1 แสดงไลบรารีฟังก์ชันคณิตศาสตร์

Function	Standard Library	Argument	Result	Purpose:Example
abs(x)	<cstdlib>	int	int	หาค่าสัมบูรณ์ของตัวเลขจำนวนเต็ม เช่น abs(-5) มีค่าเท่ากับ 5
ceil(x)	<cmath>	double	double	หาค่าตัวเลขที่น้อยที่สุดแต่มากกว่า x เช่น ceil(45.23) มีค่าเท่ากับ 46.0
cos(x)	<cmath>	double (radians)	double	หาค่า cosine ของมุม x เช่น cos(0.0) มีค่าเท่ากับ 1.0
exp(x)	<cmath>	double	double	หาค่า e^x โดย e = 2.71828... เช่น exp(1.0) มีค่าเท่ากับ 2.71828
fabs(x)	<cmath>	double	double	หาค่าสัมบูรณ์ของเลขจำนวนจริง เช่น fabs(-8.43) มีค่าเท่ากับ 8.43
floor(x)	<cmath>	double	double	หาค่าที่มากที่สุดแต่น้อยกว่า x เช่น floor(45.23) มีค่าเท่ากับ 45.0
log(x)	<cmath>	double	double	หาค่าลอการิทึม ของ x โดย x>0.0 เช่น log(2.71828) มีค่าเท่ากับ 1.0
log10(x)	<cmath>	double	double	หาค่าลอการิทึมฐาน10 ของ x โดย x>0.0 เช่น log10(100.0) มีค่าเท่ากับ 2
pow(x,y)	<cmath>	double	double	หาค่า x^y เช่น pow(0.16,0.5) มีค่าเท่ากับ 0.4
sin(x)	<cmath>	double (radians)	double	หาค่า sine ของมุม x เช่น sin(1.5708) มีค่าเท่ากับ 1.0
sqrt(x)	<cmath>	double	double	หาค่ารากที่สองของ x โดย x >= 0.0 เช่น sqrt(2.25) มีค่าเท่ากับ 1.5
tan(x)	<cmath>	double	double	หาค่ามุม tangent ของมุม x

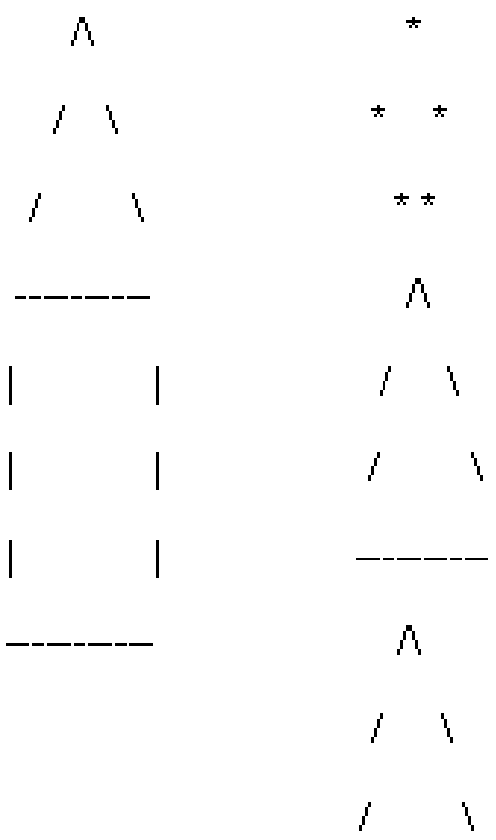
Top-Down Design และ Structure Charts

ปัญหา

ต้องการวาดภาพบ้าน และ คน



การวิเคราะห์



รูปบ้านนั้น เป็นการพิมพ์

-รูปสามเหลี่ยมที่ไม่มีฐาน

-และพิมพ์ รูปของ สี่เหลี่ยม

รูปคนนั้น

-พิมพ์ รูปวงกลม

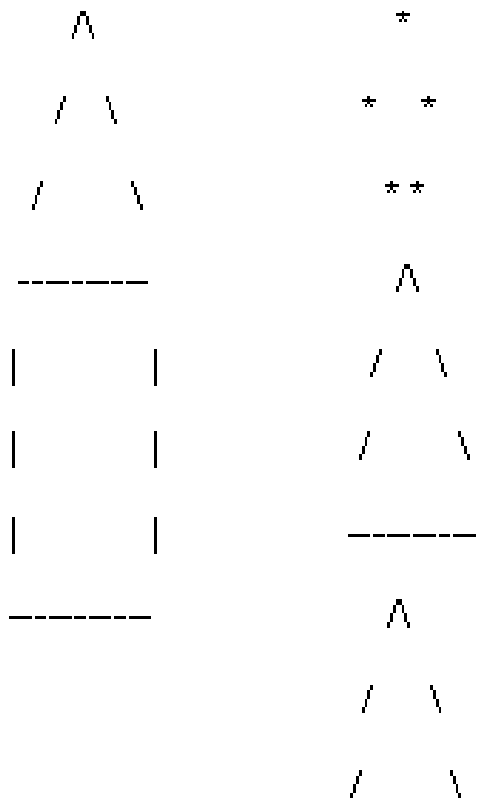
-พิมพ์ สามเหลี่ยมที่ไม่มีฐาน

-พิมพ์รูปเส้นตรง

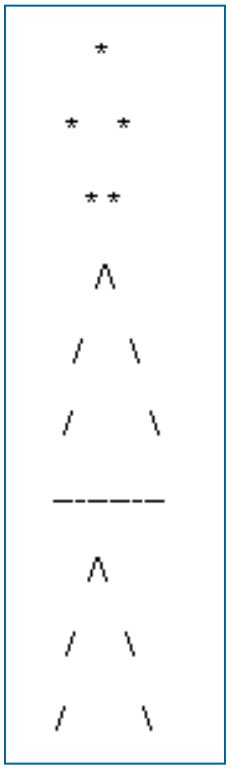
-และพิมพ์รูปสามเหลี่ยมที่ไม่มีฐาน อีกครั้งหนึ่ง ตามลำดับ

การวิเคราะห์

- a circle เป็นภาพวงกลม
- a base line เป็นเส้นตรง
- parallel lines เป็นเส้นขนาน
- intersecting lines เป็นสามเหลี่ยมที่ไม่มีฐาน



การออกแบบ



INITIAL ALGORITHM

- วาดวงกลม
- วาดสามเหลี่ยม
- วาดสามเหลี่ยมไม่มีฐาน

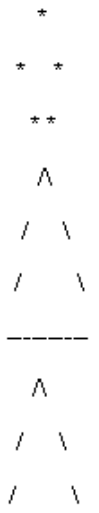
ALGORITHM REFINEMENTS

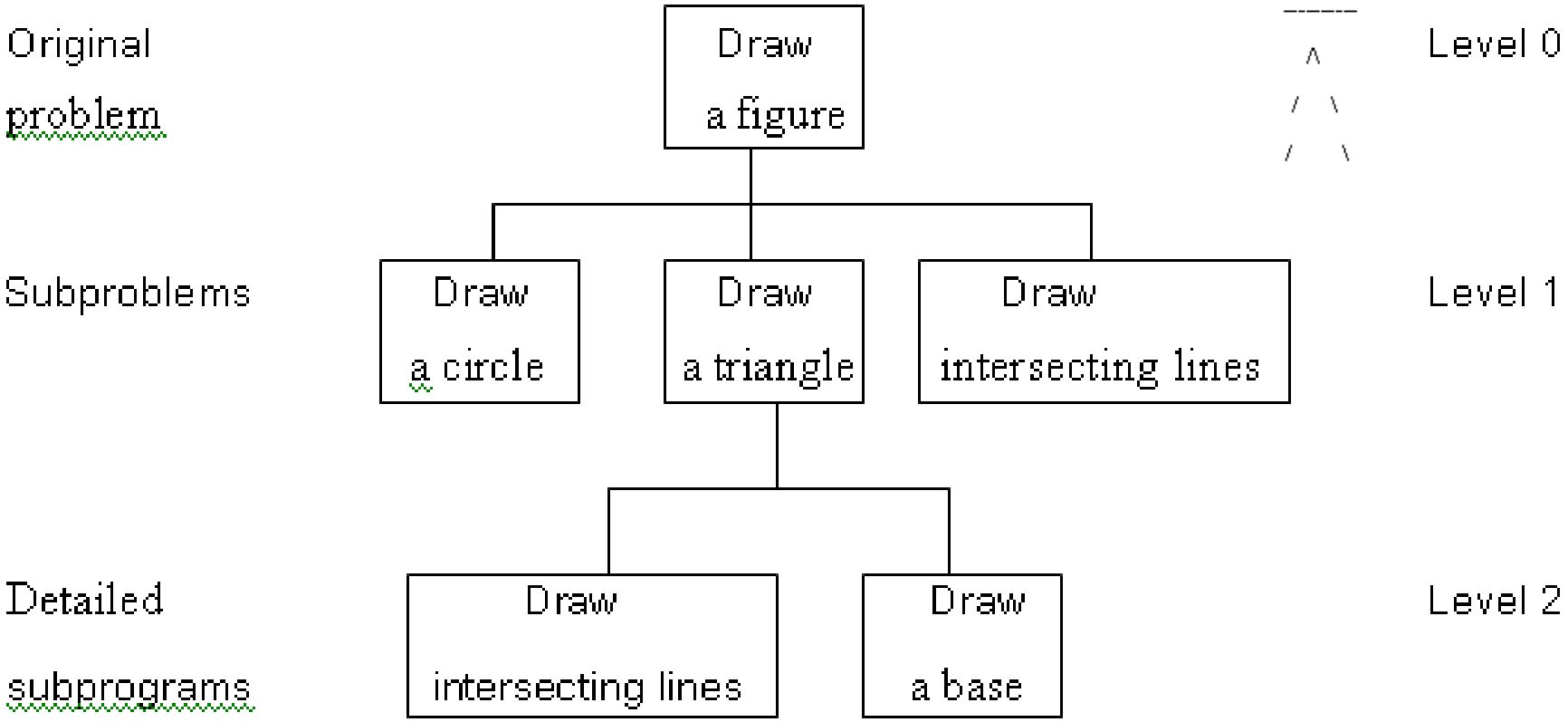
เนื่องจากรูปสามเหลี่ยมไม่มี เราสามารถกำหนดรูปนี้ได้จากองค์ประกอบอื่นๆ

Step 2 Refinement

- วาดสามเหลี่ยมไม่มีฐาน
- วาดเส้นตรง

structure chart





Function

- **Function without Arguments**
- **Functions with Input Arguments**
- **Function with Multiple Arguments**
- **Functions with Input Arguments and Single Result**
- **Function Pass by Value and Reference Parameters**

Function without Arguments

Function Call Statement(Function Without Arguments)
รูปแบบ: fname( );
ตัวอย่าง: drawCircle( ) ;

Function Prototypes

Function Prototype(Function Without Arguments)
รูปแบบ: ftype fname( ) ; **ตัวอย่าง**: void drawCircle( ) ;

Function prototypes and main function for drawing a stick figure

```
/ / Draws a stick figure (main function only)
#include <iostream>
using namespace std;
/ / Functions used  …
void drawCircle ( ) ;        / / Draws a circle
void drawTriangle ( );       / / Draws a triangle
void drawIntersect ( );      / / Draws intersecting lines
void drawBase ( );           / / Draws a horizontal line
int main ( )
{
   / / Draw a circle.
   drawCircle ( );
   / / Draw a triangle.
   drawTriangle ( );
   / / Draw intersecting lines.
   drawIntersect ( );
   return 0;
}
```

Function Definitions

Function Definition (Function Without Arguments)

รูปแบบ : ftype fname( )
{
local declarations
executable statements
}

```
//Draws a circle
void  drawCircle( )
{  cout << "   *   " << endl ;
   cout << "*     *" << endl ;
   cout << "  * *  " << endl ;
}
```

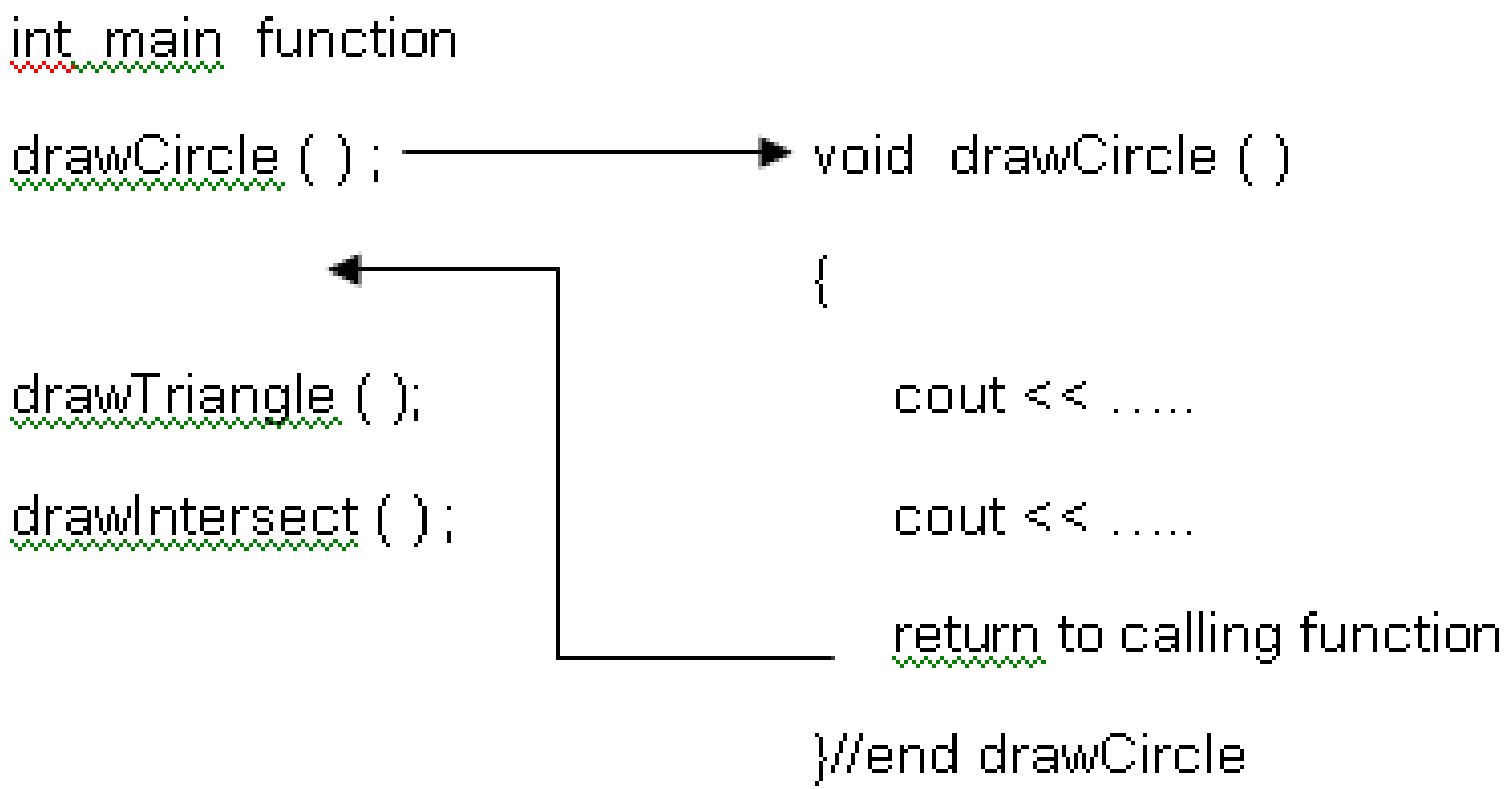
```
// Draws a triangle
void drawTriangle ( )
{
   drawIntersect ( ) ;
   drawBase ( ) ;
}
```

Program to draw a stick figure

```
/ / File: stickFigure.cpp
/ / Drawa a stick figure
#include <iostream>
using namespace std;
/ / Functions used …
void drawCircle ( ) ;
void drawTriangle ( ) ;
void drawIntersect ( ) ;
void drawBase ( ) ;
int main ( )
{
    / / Draw a circle.
    drawCircle ( ) ;
    / / Draw a triangle.
    drawTriangle ( ) ;
    / / Draw intersecting lines.
    drawIntersect ( ) ;
    return 0 ;
}
```

```
/ / Draws a circle
void drawCircle ( )
{
    cout << "       *     " << endl;
    cout << "   *       *  " << endl;
    cout << "     *   *    " << endl;
}          / / end drawCircle
/ / Draws a triangle
void drawTriangle ( )
{
    drawIntersect ( );
    drawBase( );
}   / / end drawTriangle
/ /  Draws intersecting lines
void drawIntersect ( )
{
    cout << "         / \ \       " << endl;
    cout << "       /    \ \      " << endl;
    cout << "     /       \ \     " << endl;
}   / / end drawIntersect
/ / Draws a horizontal line
void drawBase ( )
{
    cout <<  " _ _ _ _ _ _ _ "  << endl;
}   / / end drawBase
```

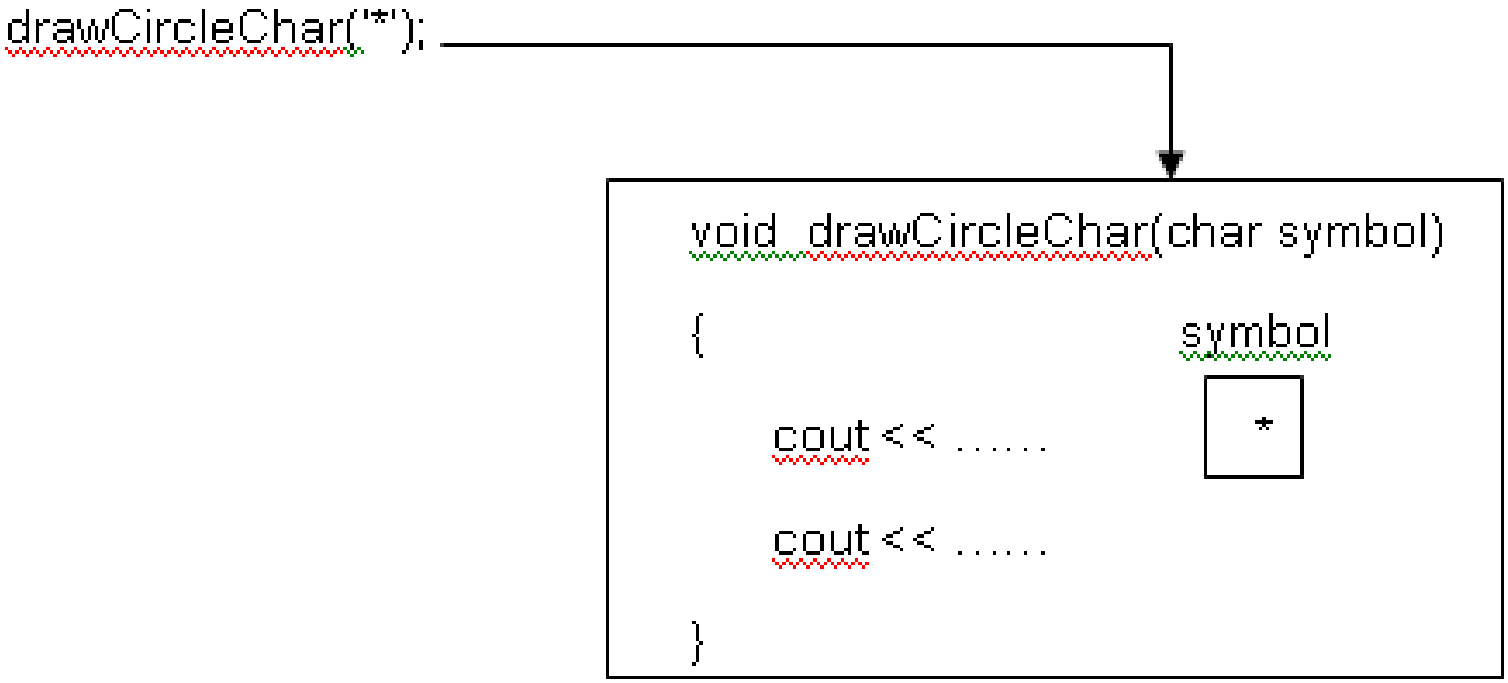
แสดงการเรียกใช้ระหว่างฟังก์ชันหลักและโปรแกรมย่อย



Functions with Input Arguments

```
// Draws a circle using the character specified by symbol

void drawCircleChar (char  symbol)

{   cout << "      " << symbol << endl;

    cout << "   " << symbol << "      " << symbol << endl;

    cout << "    " << symbol << " " << symbol <<  endl;

}

// end  drawCircle
```

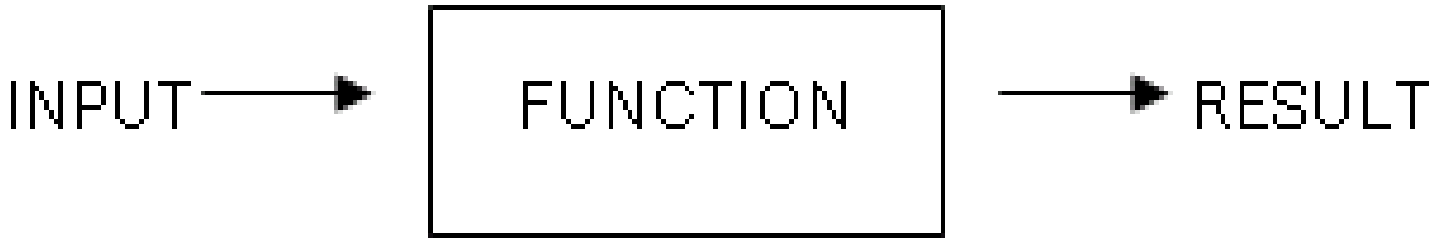
Call Functions with Input Arguments



Prototype

```
void  drawCircleChar (char) ;
```

Functions with Input Arguments and Single Result



Functions with Input Arguments and Single Result

Function Definition (Input Arguments and One Result)

รูปแบบ:

```
// function interface comment
ftype fname (formal-parameter-declaration-list)
{
    local variable declarations
    executable statements
}
```

ตัวอย่าง :

```
// Finds the cube of its argument.
// Pre : n is defined .
int cube (int n)
{
    return (n*n*n) ;
} // end cube
```

Functions with Input Arguments and Single Result

Function Prototype (With Parameters)
รูปแบบ : ftype fname (formal-parameter-type-list) ; **ตัวอย่าง :** int cube(int) ;

Functions findCircum and findArea

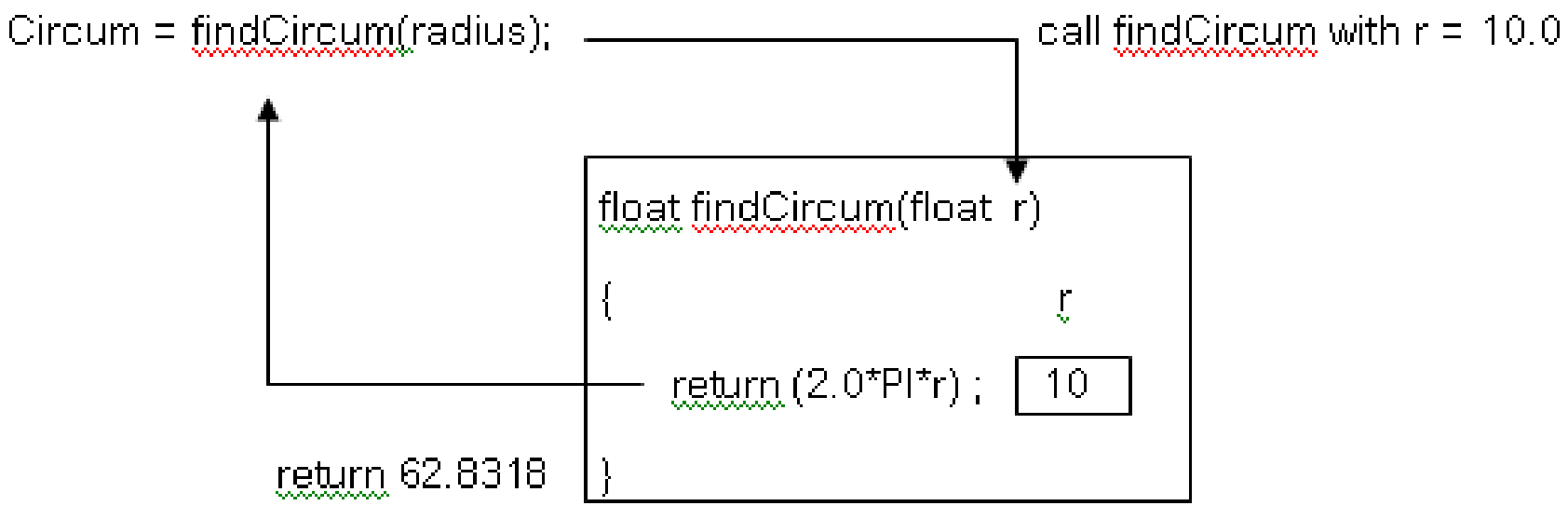
```
float findCircum (float r)
{
   return (2.0 * PI * r );
}
```

```
float findArea (float r )
{
   return  ( PI  *  r * r );
}
```

Functions findCircum

การเรียกใช้เป็นดังนี้

```
radius  = 10.0 ;
circum = findCircum(radius) ;
```



Function with Multiple Arguments

```
/ / specified by its second argument.
/ / Pre  :  x and n are defined and library
// cmath is included.
float scale ( float  x,  int  n )
{
  float  scaleFactor ;       / / local variable
  scaleFactor = pow(10 , n) ;
  return  ( x *  scaleFactor ) ;
}
```

Scope of Names

```
void one (int anArg,  double second) ;      / /  prototype 1
int  funTwo (int one, char anArg);   / / prototype 2
const  int MAX = 950;
const  int LIMIT = 200;
int main ( )
{
   int  localVar ;
   . . .
}  / / end main
void one (int  anArg,  double second)       / / header 1
{
   int oneLocal ;                           / / local 1
   . . .
}  / / end one
int  funTwo (int one,  char anArg)   / / header 2
{
   int localVar ;                           / / local 2
   . . .
}  / / end funTwo
```

Value and Reference Parameters

- ในการแก้ปัญหาบางอย่างนั้น เราต้องการให้การทำงานของฟังก์ชันนั้นสามารถส่งผ่านข้อมูลได้มากกว่า 1 ค่า
- เราใช้เครื่องหมาย &(ampersand) เป็นตัวกำหนด โดยพารามิเตอร์ตัวใดมีการระบุ & จะแสดงให้ทราบว่ามีการส่งผ่านแบบอ้างอิง โดยจะไม่กำหนดหรือจองเนื้อที่ใหม่ในการปฏิบัติงานในฟังก์ชัน แต่จะใช้เนื้อที่เดียวกับพารามิเตอร์ที่ส่งผ่านมาจากจุดเรียกใช้

โปรแกรมหาค่าผลรวมและค่าเฉลี่ย

กำหนดชื่อฟังก์ชัน และการเรียกใช้ได้ดังนี้

computeSumAve(x ,y , sum ,mean) ;

การเรียกใช้ฟังก์ชันนี้มีการส่งผ่านพารามิเตอร์ 4 ตัวด้วยกันโดย x, y , sum ,mean เป็น Actual Argument

ซึ่ง x , y เป็นค่าของข้อมูลที่เป็นเลขจำนวนจริงใดๆซึ่งทำหน้าที่เป็นข้อมูลเข้าให้แก่ฟังก์ชัน โดยก่อนการเรียกใช้ค่า x ,y ต้องมีค่าเก็บอยู่ก่อนแล้ว

แต่ sum และ mean ยังไม่มีค่าใดๆ แต่เมื่อมีการเรียกใช้ฟังก์ชันนี้ผลลัพธ์จากการทำงานจะให้ค่าแก่ sum และ mean

```
#include <iostream>
using namespace std;
// Function prototype
void computeSumAve(float , float ,
    float& , float&);
int main ()
{
   float x ,   // input – first number
         y ,  // input – second
    number
      sum,   // output – their sum
      mean ;// output – their
    average
    cout << "Enter 2 numbers: " ;
    cin >> x >> y ;
    computeSumAve(x , y , sum , mean);
   // Display results
    cout << "Sum is " << sum <<
          endl ;
    cout << "Average is " << mean
          << endl ;
    return 0 ;
}
```

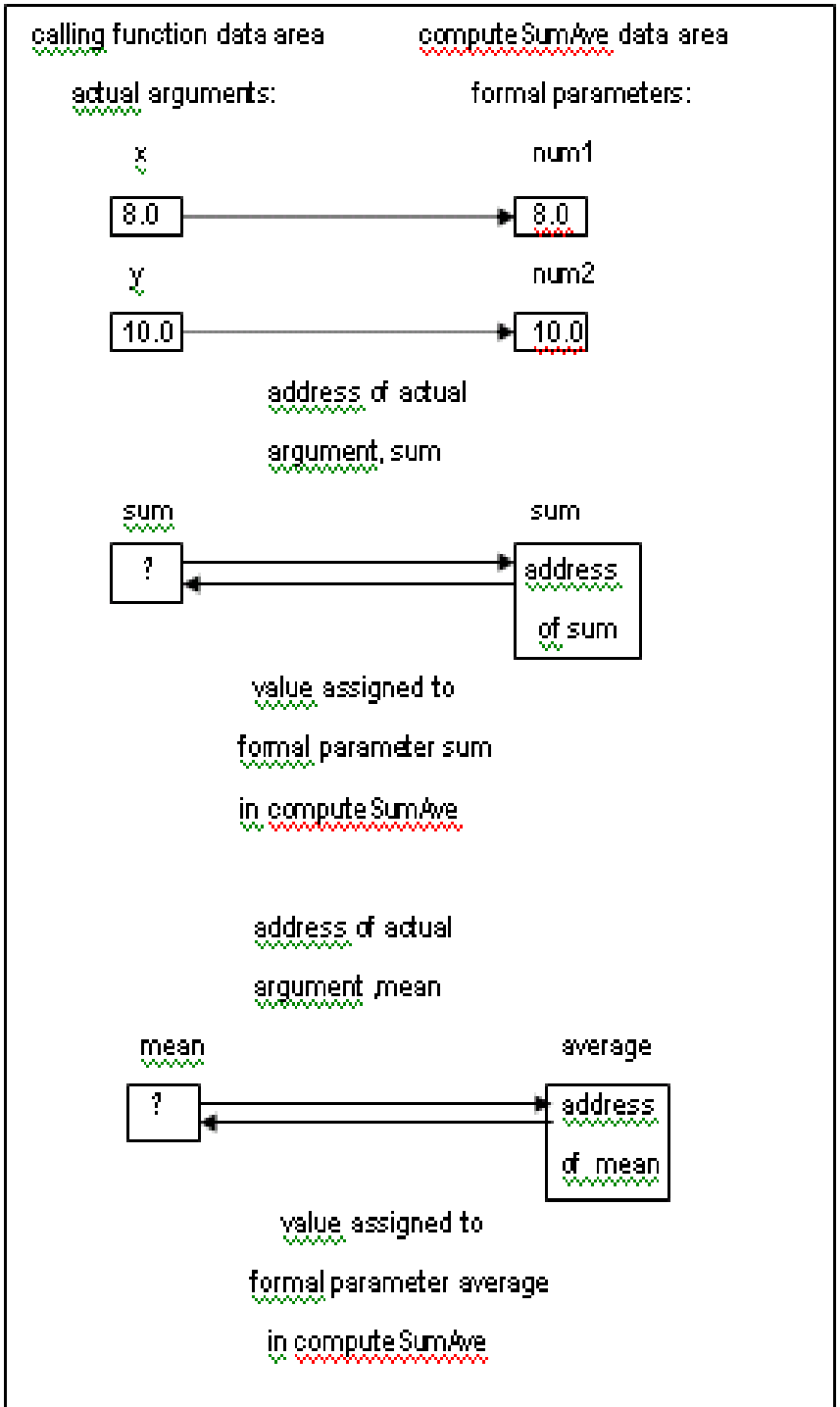
```
void computeSumAve
          (float num1 ,
           float num2 ,
           float& sum ,
          float& average)
{
   sum = num1 + num2 ;
   average = sum / 2.0 ;
}
```

```
Enter 2 numbers : 8.0  10.0

Sum is 18

Average is 9
```

```
Enter 2 numbers : 8.0  10.0
Sum is 18
Average is 9
```

```
computeSumAve(x ,y , sum ,mean) ;
```

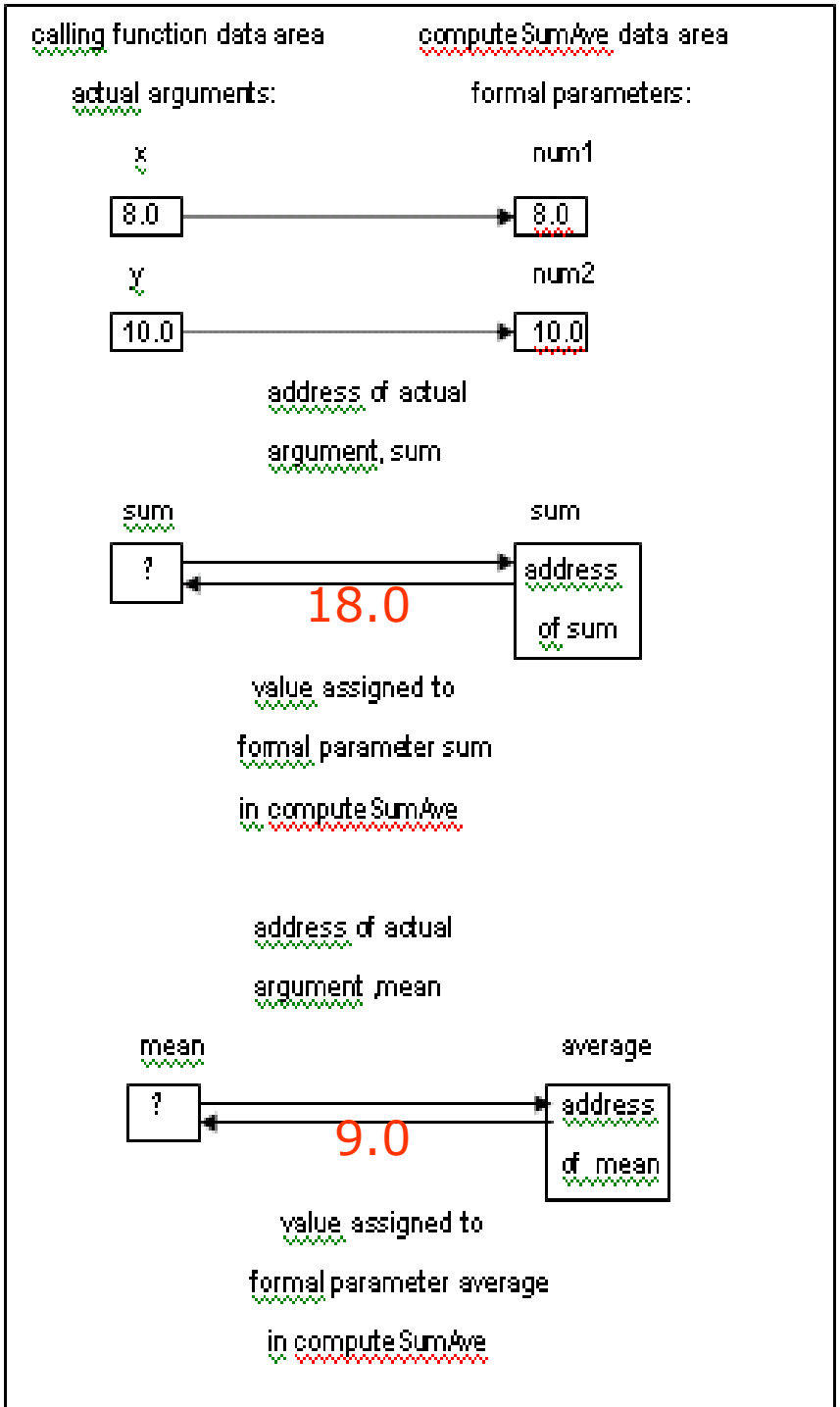
```
void computeSumAve
        (float num1 , float num2
        ,float& sum ,float& average)
{
   sum = num1 + num2 ;
   average = sum / 2.0 ;
}
```



```
Enter 2 numbers : 8.0  10.0
Sum is 18
Average is 9
```

computeSumAve(x ,y , sum ,mean) ;

```
void computeSumAve
        (float num1 , float num2
        ,float& sum ,float& average)
{
   sum = num1 + num2 ;
   average = sum / 2.0 ;
}
```



ตัวอย่างที่ 1

จงเขียนโปรแกรม รับเลขจำนวนเต็ม 5 จำนวน จงหา

1. ผลรวมของตัวเลขเหล่านี้
2. หาค่าเฉลี่ยของตัวเลขเหล่านี้

โดยสร้างเป็นฟังก์ชัน

```
int sum(int.int.int.int,int);
    double average(int);
```

ตัวอย่างที่ 2

จงเขียนโปรแกรมหาจำนวนชั่วโมง , นาที และวินาที โดยสร้างเป็นฟังก์ชันดังนี้

```
void  Find_time(int,int&,int&,int&);
```

ตัวอย่างที่ 3

จงเขียนฟังก์ชันในการสลับค่าข้อมูล 2 ตัว

ตัวแปร	X	Y
ก่อนทำงาน	6	3
หลังทำงาน	3	6

เขียนอย่างไร

```
#include <iostream>
using namespace std;
void Test1(int& ,int);
int main()
{
   int  a,b,c ;
    a = 3 ; b = 9; c = 5 ;
   Test1(b , c) ;
   return 0;
}
void Test1(int& a ,int  b)
{
     int c=4 ;
   a = b ;  b= c ; c = a ;
}
```

แบบฝึกหัด

หลังจากทำงาน

ตัวแปร a , b , c มีค่าเท่าใด

```
#include <iostream>
using namespace std;
void Test2(int ,int&);
int main()
{
   int  a,b,c ;
    a = 3 ; b = 9; c = 5 ;
   Test2(c , a) ;
   return 0;
}
void Test2(int a ,int&  b )
{
   int c=4 ;
   a = b ;  b= c ; c = a ;
}
```

แบบฝึกหัด

หลังจากทำงาน

ตัวแปร a , b , c มีค่าเท่าใด

```
#include <iostream>
using namespace std;
void Test3(int& ,int&);
int main()
{
   int  a,b,c ;
    a = 3 ; b = 9; c = 5 ;
   Test3(a , b) ;
   return 0;
}
void Test3(int& b,int& a)
{
     int c=4 ;
   a = b ;  b= c ; c = a ;
}
```

แบบฝึกหัด

หลังจากทำงาน

ตัวแปร a , b , c มีค่าเท่าใด

```
#include <iostream>
using namespace std;
void Test4(int ,int);
int main()
{
   int  a,b,c ;
    a = 3 ; b = 9; c = 5 ;
   Test4(b , c) ;
   return 0;
}
void Test4(int a ,int  b)
{
     int c=4 ;
   a = b ;  b= c ; c = a ;
}
```

แบบฝึกหัด

หลังจากทำงาน

ตัวแปร a , b , c มีค่าเท่าใด

```
#include <iostream>
using namespace std;
int Test5(int ,int) ;
int main()
{
   int  a,b,c ;
    a = 3 ; b = 9; c = 5 ;
    a = Test5(c , b) ;
   return 0;
}
int Test5(int a  ,int b)
{
   int c = 4 ;
   C = C+B;
    return c ;
}
```

แบบฝึกหัด

หลังจากทำงาน

ตัวแปร a , b , c มีค่าเท่าใด

การบ้านข้อที่ 1

จงเขียนโปรแกรมหาพื้นที่ของสามเหลี่ยม ,พื้นที่ของสี่เหลี่ยม โดยผู้ใช้ป้อน ความยาวฐาน และส่วนสูงใดๆทางแป้นพิมพ์

จงเสร้างฟังก์ชันในการหาพื้นที่ดังนี้

```
float Triangle(float , float);
void Rectangle(float ,float ,float&);
```

การบ้านข้อที่ 2

จงเขียนโปรแกรมในการคิดดอกเบี้ย โดยสร้างฟังก์ชันในการคิดดอกเบี้ย ประเภท 1 ปี 6 เดือน และ 3 เดือน

โปรแกรมนี้ให้ผู้ใช้ป้อนเงินต้น และ อัตราดอกเบี้ยต่อปี การทำงานจะพิมพ์ว่าถ้าฝากประเภท 1 ปีได้เงินเท่าใด

ฝากประเภท 6 เดือนได้เงินเท่าใดใน 1 ปี

ฝากประเภท 3 เดือนได้เงินเท่าใดใน 1 ปี