

## บทที่ 2

### ภาพรวมของ C++

#### วัตถุประสงค์

1. เพื่อให้ นักศึกษาทราบถึงองค์ประกอบของภาษา C++
2. เพื่อให้ นักศึกษาสามารถประกาศตัวแปรและกำหนดค่าคงที่ได้
3. เพื่อให้ นักศึกษาทราบชนิดข้อมูลประเภทต่างๆ
4. เพื่อให้ นักศึกษาสามารถแก้ปัญหาโปรแกรมทางคณิตศาสตร์ได้
5. เพื่อให้ นักศึกษาสามารถประมวลผลและทราบถึงข้อผิดพลาดในการเขียนโปรแกรม

สำหรับบทนี้จะอธิบายถึงภาพรวมของภาษา C++ ซึ่งเป็นภาษาระดับสูงที่พัฒนาขึ้นในกลางปี ค.ศ. 1980 โดย Bjarne Stroustrup โดย AT&T's Bell Laboratories ภาษานี้นิยมนำไปใช้ในการแก้ปัญหาเพราะเป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ และรวบรวมจุดเด่นต่างๆในการเขียนโปรแกรมไว้มากมาย มารู้จักภาษานี้กันดีกว่า

## 2.1 องค์ประกอบของภาษา C++

ภาษา C++ เป็นภาษาที่ประกอบด้วยคำภาษาอังกฤษและอักขระต่างๆ ที่ประกอบกันเป็นประโยคหรือคำสั่ง ที่สั่งให้เครื่องคอมพิวเตอร์ปฏิบัติตามได้ จากตัวอย่างที่ 2.1 เป็นการแปลงระยะทางไมล์เป็นกิโลเมตรนั้น สามารถอธิบายรายละเอียดได้ดังนี้

### ตัวอย่างที่ 2.1 การแปลงระยะทางไมล์เป็นกิโลเมตร

```
//miles.cpp
//Converts distance in miles to kilometers.
#include <iostream>
using namespace std;
int main() //start of main function
{
    const float KM_PER_MILE=1.609 //1.609 km in a mile
    float miles, //input : distance in miles
          kms; //output: distance in kilometers

    //Get the distance in miles.
    cout << "Enter the distance to miles: ";
    cin >> miles;

    // Convert the distance to kilometers.
    kms = KM_PER_MILE * miles;
```

```

// Display the distance to kilometers.
cout << " The distance in kilometers is " << kms << endl;
return 0;
}

```

### คำอธิบาย(Comments)

สองบรรทัดแรกของโปรแกรมตัวอย่างที่ 2.1 ซึ่งขึ้นต้นด้วย // เป็นคำอธิบายโปรแกรม ซึ่งเป็นข้อความที่โปรแกรมเมอร์ใช้เป็นเอกสารภายในโปรแกรมโดยอธิบายความหมายของตัวแปรและการปฏิบัติงานในโปรแกรม เพื่อช่วยให้โปรแกรมเมอร์เข้าใจและสามารถแก้ไขคำสั่งเมื่อเกิดข้อผิดพลาดได้ง่ายขึ้น คำอธิบายนี้เป็นข้อความไม่ใช่คำสั่งของโปรแกรมภาษาจึงไม่ถูกแปลเป็นภาษาเครื่องแต่อย่างใด

Comment	
<b>รูปแบบ:</b>	// comment /* comment */
<b>ตัวอย่าง:</b>	// miles.cpp /* Converts distance in miles to kilometers.*/
<b>ความหมาย :</b>	โดย // เป็นจุดเริ่มต้นของคำอธิบายในกรณีที่มีคำอธิบายเพียงบรรทัดเดียว โดยอาจใช้ /* เป็นจุดเริ่มต้น และ */ เป็นจุดสิ้นสุดก็ได้ ในกรณีที่มีคำอธิบายหลายบรรทัด

### Compiler Directive #include

จากตัวอย่างที่ 2.1 #include <iostream> เป็นคำสั่งที่ปฏิบัติการก่อนการแปลโปรแกรม ซึ่งทำหน้าที่ในการนำฟังก์ชันต่างๆจากคลาส iostream แทรกเข้าไปในโปรแกรม โดยฟังก์ชันต่างๆ

เหล่านี้เกี่ยวข้องกับการอ่านข้อมูลจากแป้นพิมพ์ การแสดงผลทางจอภาพ โดยในโปรแกรมที่พัฒนาขึ้นส่วนมากมีการเรียกใช้ฟังก์ชันต่างๆเหล่านี้ ภาษา C++ ได้เตรียม standard library ต่างๆนอกเหนือจากคลาส iostream อีกมากมายและผ่านการทดสอบความถูกต้องเรียบร้อยแล้ว โปรแกรมเมอร์สามารถเรียกใช้ได้เพียงแต่ทำการ #include และตามด้วยชื่อคลาสที่ต้องการใช้เท่านั้น ทำให้เขียนโปรแกรมสั้นลงและประหยัดเวลาและค่าใช้จ่ายในการพัฒนาอีกด้วย

### Compiler Directive #include

**รูปแบบ:** #include <filename>

**ตัวอย่าง:** #include <iostream>

**ความหมาย :** ก่อนการแปล บรรทัดนี้จะถูกแทนที่ด้วย standard library filename โดยทั่วไปคำสั่งนี้จะอยู่ต้นโปรแกรม และต้องกำหนดก่อนการเรียกใช้ฟังก์ชันภายในคลาสนั้น

**หมายเหตุ :** คำสั่ง compiler directive จะไม่จบด้วยเครื่องหมาย ;

### Namespace std

จากตัวอย่างที่ 2.1 คำสั่ง using namespace std; เป็นการกำหนดการใช้ขอบเขต (object) ในส่วนพิเศษที่เรียกว่า namespace std(short for standard) โดยคำสั่งนี้จะต้องมีการกำหนดเมื่อโปรแกรมเมอร์มีการเรียกใช้ C++ standard library

## using namespace

**รูปแบบ:** `using namespace region ;`

**ตัวอย่าง:** `using namespace std;`

**ความหมาย :** เป็นคำสั่งที่แสดงว่าโปรแกรมมีการใช้อุปกรณ์ที่กำหนดใน namespace ที่ระบุโดย region โดยทั่วไปบรรทัดนี้จะตามจากบรรทัดที่ #include เสมอ

### Function main

ในตัวอย่างที่ 2.1 `int main() //start of main function`

ภาษา C++ ทุกโปรแกรมต้องมีฟังก์ชัน main ฟังก์ชันนี้เป็นจุดเริ่มต้นในการปฏิบัติงานของโปรแกรม คำว่า int เป็นคำเฉพาะที่บ่งบอกว่าฟังก์ชัน main นี้ต้องมีการส่งผ่านค่าจำนวนเต็มกลับไปให้ระบบปฏิบัติงานเมื่อการดำเนินงานต่างๆภายในฟังก์ชันที่อยู่ภายใต้เครื่องหมาย ( ) เสร็จสิ้นแล้ว และคำว่า main ต้องตามด้วยเครื่องหมาย () เสมอ สำหรับ // เป็นคำอธิบายเพื่อให้เข้าใจความหมายของคำสั่งได้ชัดเจนขึ้น

### การประกาศตัวแปร(Declaration Statements)

ในส่วนของ function body สามารถแบ่งคำสั่งได้เป็น 2 ชนิดคือ declaration statement และ executable statement สำหรับ declaration statement เป็นคำสั่งในการประกาศตัวแปรทั้งหมดที่ใช้ในฟังก์ชันให้กับคอมพิวเตอร์ได้รับรู้

```
const float KM_PER_MILE=1.609           //1.609 km in a mile
float miles,                             //input : distance in miles
      kms;                                //output: distance in kilometers
```

ในฟังก์ชันนี้มีตัวแปรทั้งหมด 3 ตัวได้แก่ KM\_PER\_MILE, miles และ kms โดย KM\_PER\_MILES เป็นตัวแปรที่เก็บค่าคงที่มีค่าเท่ากับ 1.609 สำหรับ miles และ kms เป็นตัวแปรที่เก็บค่าจำนวนจริงใดๆในการปฏิบัติงานโปรแกรม

main Function definition	
<b>รูปแบบ:</b>	<pre>int main() {     function body }</pre>
<b>ตัวอย่าง:</b>	<pre>int main() {     cout &lt;&lt; "Enjoy C++ " &lt;&lt; endl;     return 0; }</pre>
<b>ความหมาย :</b>	เป็นจุดเริ่มต้นของการปฏิบัติงาน โดยกระทำงานตามคำสั่งในเครื่องหมาย {} และต้องมีคำสั่ง return 0 เพื่อส่งค่ากลับไปให้ระบบปฏิบัติการเมื่อการปฏิบัติการสิ้นสุดลง

### คำสั่งประมวลผล(Executable Statement)

เป็นคำสั่งในการประมวลผลโปรแกรม

```
//Get the distance in miles.
```

```
cout << "Enter the distance to miles: ";
```

เป็นคำสั่งในการแสดงข้อความในเครื่องหมาย " " ออกทางจอภาพ

```
cin >> miles;
```

เป็นคำสั่งที่รอรับค่าจำนวนจริงจากผู้ใช้งานแบบพิมพ์ ซึ่งค่าที่ป้อนคือระยะทางไมล์ใดๆที่ต้องการแปลงเป็นกิโลเมตร ดังนี้

```
Enter the distance to miles: 10.0
```

ในที่นี้ผู้ใช้เห็นข้อความปรากฏ จะมี prompt กระหิวรอให้ผู้ป้อนข้อมูล ผู้ใช้ป้อน 10.0 และ กด Enter การทำงานของฟังก์ชัน main จะกระทำในคำสั่งถัดไป

```
// Convert the distance to kilometers.
```

```
kms = KM_PER_MILE * miles;
```

คำสั่งนี้เป็นการคำนวณเพื่อแปลงค่าที่ผู้ใช้ป้อน โดยนำค่า 10.0 \* 16.09 นำผลลัพธ์เก็บในตัวแปร kms โดย 10.0 เป็นค่าที่ผู้ใช้ป้อนเก็บในตัวแปร miles ส่วน 16.09 เป็นค่าที่เก็บในตัวแปร KM\_PER\_MILE ซึ่งกำหนดไว้ตั้งแต่ตอนต้น

```
// Display the distance to kilometers.
```

```
cout << "The distance in kilometers is " << kms << endl;
```

คำสั่งนี้เป็นคำสั่งในการแสดงผลออกมาทางจอภาพ โดยมีการพิมพ์ข้อความในเครื่องหมาย " " ออกทางจอภาพตามด้วย ค่าของข้อมูลที่เก็บในตัวแปร kms ดังนี้

```
The distance in kilometers is 16.09
```

ท้ายสุดคำสั่ง

```
return 0;
```

เป็นคำสั่งในการส่งค่า 0 กลับไปให้กับระบบปฏิบัติการ เป็นจุดสิ้นสุดการปฏิบัติการของฟังก์ชัน

## 2.2 คำเฉพาะ(Reserved Word) และ ชื่อ(Identifiers)

คำต่างๆที่ใช้ในโปรแกรมแบ่งออกเป็น 2 ชนิดคือคำเฉพาะและคำที่ใช้กำหนดขึ้นมาใหม่เพื่อใช้งาน ในคำสั่งหนึ่งของภาษา C++ จะประกอบด้วยคำทั้งสองชนิดนี้ปะปนกัน เรามารู้จักคำทั้งสองชนิดกันดีกว่า

## คำเฉพาะ (Reserved Words)

เรียกอีกอย่างหนึ่งว่า keyword เป็นคำสงวนไว้ไม่ให้โปรแกรมเมอร์ไปใช้ในวัตถุประสงค์อื่น ๆ เนื่องจากมีความหมายใช้งานเฉพาะในภาษา C++ ประกอบด้วยคำต่างๆดังนี้

and	default	inline	pre_cast	typename
and_eq	delete	int	return	union
asm	do	long	short	unsigned
auto	double	mutable	signed	using
bitand	dynamic_cast	namespace	sizeof	virtual
bitor	else	new	static	void
bool	enum	not	static_cast	volatile
break	explicit	not_eq	struct	wchar_t
case	export	operator	switch	while
catch	extern	or	template	xor
char	false	or_eq	this	xor_eq
class	float	private	throw	
compl	for	protected	true	
const	friend	public	try	
const_cast	goto	register	typedef	
continue	if	reinter	typeid	

จากตัวอย่างที่ 2.1 มีคำเฉพาะดังนี้ const , float , include , int , namespace , return ,using

## ชื่อ(Identifiers)

คำสั่งต่างๆในภาษา C++ ประกอบด้วยคำต่างๆซึ่งอาจเป็นคำเฉพาะ หรือคำที่โปรแกรมเมอร์กำหนดขึ้นใหม่อาจเป็นเป็นคำที่ใช้แทนชื่อตัวแปร ชื่อค่าคงที่ ชื่อออปเจก ชื่อฟังก์ชัน หรืออื่นๆ



โดยทั่วไปแล้วการตั้งชื่อควรให้สื่อความหมายเพื่อให้ง่ายต่อการทำความเข้าใจ และการตั้งชื่อต้องมีกฎเกณฑ์ตามหลักของไวยากรณ์ภาษาดังนี้

1. เริ่มต้นด้วยตัวอักษรภาษาอังกฤษ หรือเครื่องหมาย underscore (ขีดล่าง) ภาษา C++ ถือว่าอักษรภาษาอังกฤษตัวเล็กกับตัวใหญ่เป็นคนละตัวกัน เช่น Rate , rate , RATE เป็นต้น
2. ตัวต่อไปอาจเป็นตัวเลข ตัวอักษร หรือ ขีดล่างก็ได้ เช่น Letter1 , letter1 , Letter\_1
3. ห้ามใช้คำเฉพาะ เป็นชื่อที่แทนความหมายอื่นใด

#### ตัวอย่างการตั้งชื่อที่ถูกต้อง

inches	cent	centperInch	hello	cent_per_inch
A	_A	_12	aA	Power_A

#### ตัวอย่างการตั้งชื่อที่ไม่ถูกต้อง

1Letters	เริ่มต้นด้วยตัวเลขไม่ถูกต้อง
float	คำเฉพาะ
const	คำเฉพาะ
two*four	ไม่อนุญาตให้ใช้ *
joe's	ไม่อนุญาตให้ใช้ '
two-dimensional	ไม่อนุญาตให้ใช้ -

### 2.3 ชนิดของข้อมูล(Data Types)

#### Data Types

ชนิดของข้อมูลที่ใช้ในโปรแกรมถูกแทนด้วยเซตหรือกลุ่มของค่าที่สามารถปฏิบัติการได้ สำหรับภาษา C++ ได้กำหนดข้อมูลไว้ใน standard library ให้ 4 กลุ่มได้แก่

1. Integers : เลขจำนวนเต็ม เป็นเลขที่ไม่มีเศษ เช่น 5 , -52 , 343222 โดยชนิดข้อมูลที่แทนเลขจำนวนเต็มได้แก่ short , int , long
2. Real numbers: เลขทศนิยม เช่น 2.5 , -0.00034 , 3.66666666 , 5.0 เป็นต้น ชนิดข้อมูลที่แทนเลขทศนิยมได้แก่ float , double , long double
3. Boolean : เป็นข้อมูลที่มีค่า true หรือ false ซึ่งเป็นค่าที่ได้จากการเปรียบเทียบด้วยคำสั่งเงื่อนไข ชนิดข้อมูลที่แทนค่าบูลีน คือ ข้อมูลชนิด bool
4. Character : เป็นข้อมูลที่เก็บอักขระ 1 ตัวอักษรซึ่งอาจเป็นอักขระที่เป็นตัวเลข ตัวอักษรภาษาอังกฤษ หรืออักขระพิเศษ ก็ได้ ข้อมูลชนิดนี้ได้แก่ char

### ข้อมูลชนิด int

เลขจำนวนเต็มทางคณิตศาสตร์แบ่งเป็น 3 กลุ่มคือเลขจำนวนเต็มบวก เลขจำนวนเต็มลบ และเลขจำนวนเต็มศูนย์ โปรแกรมเมอร์สามารถกำหนดชนิดข้อมูลเป็นเลขจำนวนเต็มได้ 3 ชนิดคือ short ; int และ long ขึ้นอยู่กับค่าของข้อมูลที่ใช้ในโปรแกรมโดยข้อมูลแต่ละชนิดมีขอบเขตของข้อมูลแตกต่างกัน โดยทั่วไปนิยามกำหนดชนิดข้อมูลจำนวนเต็มเป็น int ซึ่งค่าของข้อมูลที่สามารถปฏิบัติการในโปรแกรมได้อยู่ในช่วง -32768 ถึง 32767 ถ้าข้อมูลมีค่ามากกว่านี้หรือน้อยกว่านี้สามารถใช้ข้อมูลชนิด long หรือ short แทน โดยขอบเขตของการทำงานกับข้อมูลจำนวนเต็มนั้น C++ ได้กำหนดค่าคงที่เพื่อแทนค่าที่น้อยที่สุดและค่าที่มากที่สุดของชนิดข้อมูลจำนวนเต็มชนิดต่างๆไว้ดังตัวอย่างต่อไปนี้

```
cout << INT_MIN << " though " << INT_MAX << endl ;
```

เป็นการพิมพ์ช่วงของค่าจำนวนเต็มที่สามารถใช้งานได้เป็นชนิดข้อมูล int โดยทั่วไปเลขจำนวนเต็มในภาษา C++ ใช้สำหรับการอ่านข้อมูลทางแป้นพิมพ์ การพิมพ์ผลทางจอภาพ และการคำนวณทางคณิตศาสตร์ ซึ่งผลลัพธ์เป็นเลขจำนวนเต็ม

### ข้อมูลชนิด float

เลขทศนิยม ประกอบด้วยเลขจำนวนเต็มและเศษที่แยกกันด้วยจุดทศนิยม ภาษา C++ มีชนิดข้อมูลในกลุ่มนี้ 3 ชนิดได้แก่ float , double , long double ซึ่งมีขอบเขตของข้อมูลแตกต่างกัน

ขึ้นอยู่กับข้อมูลที่ใช้ในโปรแกรม การแทนค่าของจำนวนจริงในภาษา C++ นั้น ค่า 123000.0 แทนด้วย scientific notation ได้ดังนี้ 1.23e5 หรือ 1.23e+5 ซึ่งมีค่าเท่ากับ  $1.23 \times 10^5$  นั้นเอง

ถ้าสัมประสิทธิ์มีค่าเป็นจำนวนลบ จุดทศนิยมจะเลื่อนไปทางซ้าย อาทิเช่น 0.34e-4 มีค่าเท่ากับ 0.00034 เป็นต้น

### ตัวอย่าง เลขจำนวนจริงที่ถูกต้องและไม่ถูกต้อง

ถูกต้อง	ไม่ถูกต้อง
3.14259	-15e-0.3 (สัมประสิทธิ์ 0.3 ผิด)
0.0005	12.5e.3 (สัมประสิทธิ์ .3 ผิด)
.12345	.123E3 (ค่า E ต้องเป็นตัวเล็กเท่านั้น)
12345.0	e32 (ผิดต้องเริ่มต้นด้วยค่าตัวเลข)
16	a34e03 (ผิดต้องเริ่มต้นด้วยค่าตัวเลข)
15.0e-04 (ค่าคือ 0.0015)	
2.345e2 (ค่าคือ 234.5)	
1.15e-3 (ค่าคือ 0.00115)	
12e+5 (ค่าคือ 1200000.0)	

### ข้อมูลชนิด bool

เป็นชนิดของข้อมูลที่ประกอบด้วยค่าจริง(true) และค่าเท็จ(false) ซึ่งใช้แทนค่าที่ได้จากการเปรียบเทียบจากเงื่อนไขต่างๆ เพื่อประกอบการตัดสินใจในโปรแกรม

### ข้อมูลชนิด char

เป็นชนิดข้อมูลที่แทนอักขระ 1 ตัว ซึ่งอาจเป็นอักขระที่เป็นตัวเลข อักขระที่เป็นตัวอักษรภาษาอังกฤษ หรืออักขระพิเศษ อย่างเช่น 'A' 'z' "' " '' เป็นต้น อักขระที่เป็น literal ต้องอยู่ภายในเครื่องหมาย ' ' (single quotes) อักขระแต่ละตัวจะถูกเก็บในหน่วยความจำกิน

เนื้อที่ 1 ไบท์ นอกจากนี้มีกลุ่มของอักขระพิเศษที่เรียกว่า escape sequence ที่ใช้สำหรับ  
ควบคุม ข้อความเพื่อให้ปรากฏบนจอภาพที่ต้องการ ซึ่งถือว่าเป็นข้อมูลชนิด char ดังนี้

### Escape Sequence

Escape Sequence	ความหมาย
'\n'	linefeed
'\b'	backspace
'\r'	carriage return
'\t'	tab
'\"'	double quote
'\''	single quote
'\f'	form feed
'\\'	backslash

### String Class

เป็นคลาสที่ภาษา C++ เตรียมไว้ให้กับโปรแกรมเมอร์สำหรับการใช้ข้อมูลที่เป็นข้อความ โดย  
ข้อความคือการนำอักขระมาเรียงต่อกันเป็นกลุ่มคำ อาทิเช่น "A" "1234" "Enter the  
distance in miles " เป็นต้น ผู้ใช้สามารถอ่าน บันทึก เปรียบเทียบ รวม หรือค้นหา ข้อความ  
ต่างๆในการปฏิบัติงานได้ โดยต้องมีคำสั่ง #include <string> ในโปรแกรมก่อนใช้งาน

### การประกาศตัวแปร(Declarations)

จุดประสงค์ที่มีชนิดของข้อมูลแตกต่างกัน เพื่อเป็นการบอกให้ตัวแปลภาษาได้ทราบถึงการ  
ปฏิบัติการและแทนค่าต่างๆในหน่วยความจำได้อย่างถูกต้อง เมื่อผู้ใช้มีการใช้การปฏิบัติการที่  
ไม่ถูกต้องเช่นนำค่าที่เป็นเลขทศนิยม ไปเก็บในตัวแปลที่เป็นเลขจำนวนเต็ม ตัวแปลภาษาจะมี  
ข่าวสารความผิดพลาด(error message) แจ้งให้กับโปรแกรมเมอร์ได้ทราบ ซึ่งเป็นการป้องกัน  
ความผิดพลาดในการกระทำการปฏิบัติการที่ไม่ถูกต้อง เราสามารถกำหนดหน่วยความจำที่มี  
อยู่ให้สามารถเก็บข้อมูล หรือผลลัพธ์จากการทำงานได้ โดยแทนหน่วยความจำหนึ่งๆเป็น

ตัวแปร (variable) เพื่อแทนค่าของข้อมูล ดังนั้นการที่จะทำให้ตัวแปรต่างๆเหล่านี้ติดต่อกับคอมพิวเตอร์ได้หรือปฏิบัติงานตามโปรแกรมได้ ต้องมีการประกาศตัวแปรที่เรียกว่า variable declarations.

### Variable and Object Declarations

**รูปแบบ:** `type identifier-list ;`

**ตัวอย่าง:** `float x , y ;`  
`int me , you ;`  
`float week = 40.0 ;`  
`string flower = "rose" ;`

**ความหมาย :** เป็นการจัดสรรหน่วยความจำให้สามารถเก็บข้อมูลได้จากตัวอย่าง เป็นการกำหนดชื่อของตัวแปร (identifier list) ซึ่งอาจมีหลายตัวแยกด้วยเครื่องหมาย , ส่วน type เป็นชนิดของข้อมูลที่เก็บในตัวแปรนั้น สำหรับค่า 40.0 และ "rose" เป็นการกำหนดค่าเริ่มต้นให้กับตัวแปร หรือ ออปเจกต์ จุดจบของการประกาศตัวแปรต้องลงท้ายด้วยเครื่องหมาย ;

### การกำหนดค่าคงที่(Constant Declarations)

การประกาศตัวแปรนั้น ค่าของข้อมูลต่างๆที่เก็บสามารถแปรเปลี่ยนไปได้ตามการปฏิบัติการของโปรแกรม สำหรับข้อมูลบางประเภทมีค่าที่คงที่แต่ถูกใช้งานในโปรแกรมหลายๆแห่ง ถ้ามีการเปลี่ยนแปลงต้องตามแก้ไขทุกจุดทำให้ไม่สะดวกและเกิดความผิดพลาดได้ง่าย เช่น อัตราภาษี จำนวนนักศึกษาที่ลงทะเบียน หรือค่าที่ใช้ในการแปลงระยะทางไมล์เป็นกิโลเมตรตามตัวอย่างที่ 2.1 เป็นต้น ข้อมูลลักษณะนี้โปรแกรมเมอร์สามารถกำหนดให้เป็นค่าคงที่ที่เก็บในตัวแปรโดยไม่เปลี่ยนแปลงค่าได้ตามรูปแบบดังนี้

## Constant Declaration

**รูปแบบ:** `const type constant-identifier = value ;`

**ตัวอย่าง:** `const float PI = 3.14159;`

**ความหมาย :** เป็นการกำหนดค่าคงที่ (value) ให้กับชื่อคงที่ (constant-identifier) โดยไม่มีการเปลี่ยนแปลงค่า จุดจบของคำสั่งต้องลงท้ายด้วยเครื่องหมาย ; โดยทั่วไปจะกำหนดค่าคงที่ก่อนประกาศตัวแปรหรือออปเจก

### ตัวอย่างที่ 2.2 พิมพ์ข้อความทักทาย

---

```
//File.hello.cpp
//Display a user's name
#include <iostream>
#include <string>
using namespace std;
int main()
{
    char letter1,letter2 ;           //input and output :first two initials
    string  lastName;               //input and output :last name

    //Enter letters and print message.
    cout << " Enter 2 initials and a last name : " ;
    cin >>letter1 >>letter2 >> lastName;
    cout << "Hello" << letter1 << ". " << letter2 << ". "
        << lastName << " ! " ;
}
```

```

cout << " We hope you enjoy studying C++." << endl ;
return 0;
}

```

```

Enter 2 initials and a last name : EB Koffman
Hello E.B. Kottman! We hope you enjoy studying C++.

```

โปรแกรมนี้ประกาศตัวแปร letter1 และ letter2 เป็น char และตัวแปร lastName เป็น string การปฏิบัติงานในโปรแกรมเริ่มจาก การพิมพ์ข้อความเพื่อบอกผู้ใช้ให้ป้อนข้อมูลโดยมีข้อความดังนี้

```

Enter 2 initials and a last name :

```

ต่อจากนั้นโปรแกรมจะรอรับข้อมูลจากคำสั่ง

```

cin >>letter1 >>letter2 >> lastName;

```

ผู้ใช้ต้องโดยป้อนข้อมูลที่เป็นอักขระเป็นชื่อสองตัวแรก เว้นวรรคตามด้วยนามสกุล โปรแกรมจะอ่านอักขระ2 ตัวแรกจากการป้อนข้อมูลของผู้ใช้ไปเก็บในตัวแปรมีผลให้ letter1 เก็บค่า 'E' และ letter2 เก็บค่า 'B' นำข้อความที่เหลือเก็บในตัวแปร lastName มีผลให้ lastName เก็บค่า "Koffman" ต่อจากนั้นทำคำสั่งถัดไป

```

cout << "Hello" << letter1 << " ." << letter2 << " ."
<< lastName << " ! " ;

```

```

cout << " We hope you enjoy studying C++." << endl ;

```

ซึ่งเป็นคำสั่งในการแสดงข้อความและค่าของตัวแปรที่เก็บในหน่วยความจำออกมาทางจอภาพ โดย พิมพ์ข้อความที่อยู่ในเครื่องหมาย " " ออกมาพิมพ์ ถ้าเป็นตัวแปรจะนำค่าที่เก็บในหน่วยความจำพิมพ์ออกมาทางจอภาพโดยพิมพ์ต่อเนื่องกันไป เมื่อพบคำสั่ง endl จึงจะขึ้นบรรทัดใหม่ดังได้ผลลัพธ์ดังนี้

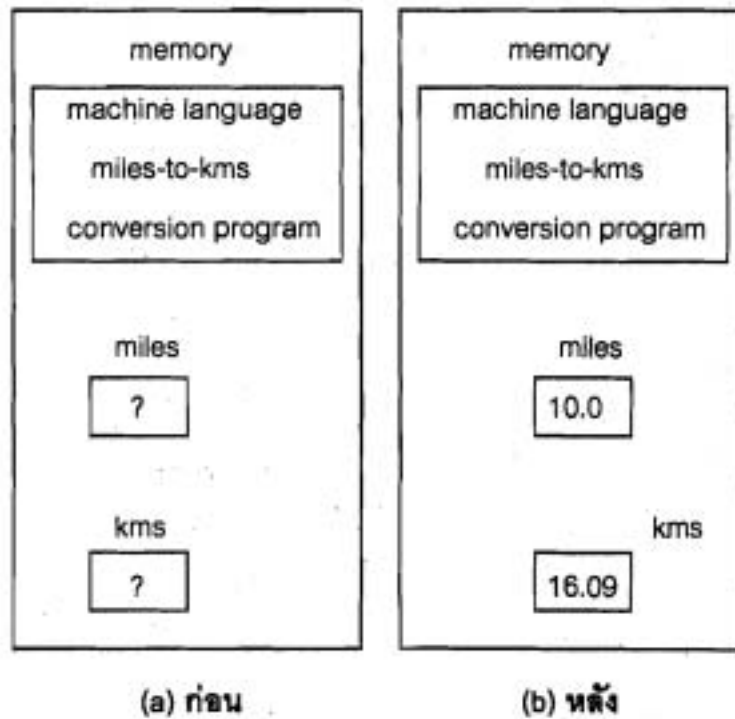
```

Hello E.B. Kottman! We hope you enjoy studying C++.

```

## 2.4 คำสั่งปฏิบัติการ(Executable Statement)

ก่อนที่โปรแกรมจะประมวลผลนั้น ต้องมีการนำโปรแกรมแปลให้เป็นภาษาเครื่องและนำไปไว้ในหน่วยความจำ มีการกำหนดเนื้อที่ในหน่วยความจำเพื่อเก็บค่าของตัวแปรและค่าคงที่ไว้ก่อนปฏิบัติการ จากรูปที่ 2.1 แสดงการจองเนื้อที่สำหรับตัวแปร miles และ kms ก่อนการประมวลผล ในระหว่างที่มีการประมวลผลมีการนำข้อมูล 10.0 จากผู้ใช้ที่ป้อนทางแป้นพิมพ์ให้ค่าแก่ ตัวแปร miles หลังการประมวลผลค่าของตัวแปร kms มีค่าเท่ากับ 16.09



รูปที่ 2.1 แสดงถึงหน่วยความจำก่อนและหลังการปฏิบัติงานโปรแกรม

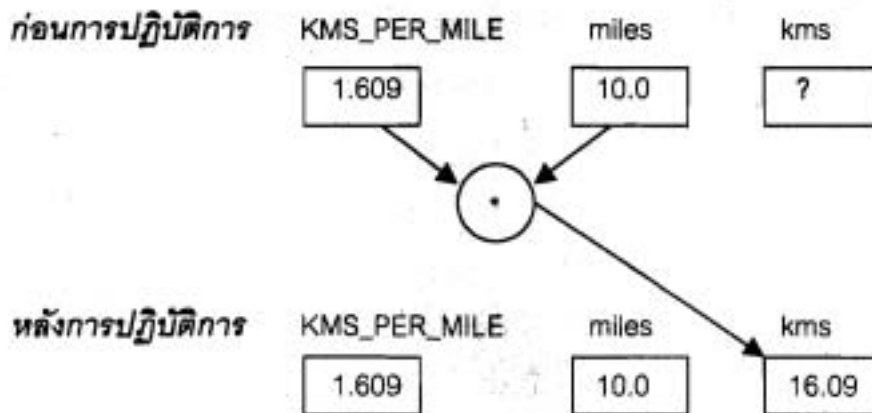
### คำสั่งกำหนดค่า(Assignment Statement)

เป็นคำสั่งในการกำหนดค่าให้แก่ตัวแปร ภาษา C++ ใช้สัญลักษณ์เท่ากับ = เป็นการให้แก่ตัวแปร โดยการให้ค่านี้เป็นผลจากการประมวลผลข้อมูลของนิพจน์ทางขวาของเครื่องหมาย = นำให้แก่ตัวแปรทางซ้ายของเครื่องหมาย ดังเช่นคำสั่ง

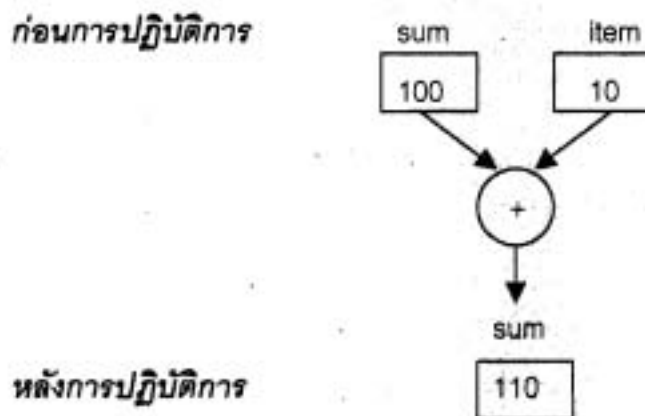


$$\text{kms} = \text{KM\_PER\_MILE} * \text{miles} ;$$

การประมวลผลจะนำค่าคงที่ KM\_PER\_MILE (1.609) คูณกับค่าที่เก็บในหน่วยความจำของตัวแปร miles (10.0) นำผลลัพธ์ให้แก่ตัวแปร kms



ค่าของตัวแปรต่างๆนั้นจะแปรเปลี่ยนไปตามคำสั่งที่ประมวลผล โดยเก็บค่าได้เพียงค่าเดียวและเป็นค่าที่ได้รับการปรับเปลี่ยนหลังสุดเท่านั้น เช่นคำสั่ง  $\text{sum} = \text{sum} + \text{item} ;$



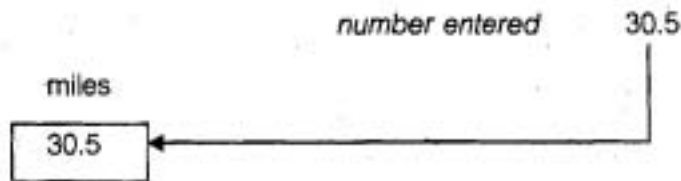
ก่อนการปฏิบัติการค่าของ sum มีค่าเท่ากับ 100 และ item มีค่าเท่ากับ 10 การประมวลผลเริ่มด้วยการนำค่าของตัวแปรทั้งสองบวกกัน ได้ผลลัพธ์เท่ากับ 110 นำค่าไปเก็บในตัวแปร sum ซึ่งแทนที่ข้อมูลตัวเดิม มีผลให้ค่าของ sum เปลี่ยนไปมีค่าเท่ากับ 110

## การรับและแสดงผล (Input/Output Operations)

ข้อมูลสามารถเก็บในหน่วยความจำได้หลายวิธี อาจจากการกำหนดค่าให้กับตัวแปร หรือจากการอ่านข้อมูลจากแป้นพิมพ์ ซึ่งจะได้ผลลัพธ์ที่ต่างกันตามข้อมูลที่ป้อนจากผู้ใช้งาน คำสั่งที่กระทำการอ่านและพิมพ์ข้อมูล ภาษา C++ ได้เตรียมไว้ให้ไว้ในคลาส `iostream` โดยกำหนดให้ออปเจกต์ `cin` และ `>>` เป็นตัวกระทำที่ใช้ในการรับข้อมูลทางแป้นพิมพ์ ส่วนออปเจกต์ `cout` และ `<<` เป็นตัวกระทำที่ใช้ในการพิมพ์ผลลัพธ์ออกทางจอภาพ โดยผู้ใช้ต้องแทรก compiler directive `#include <iostream>` ก่อนฟังก์ชัน `main`

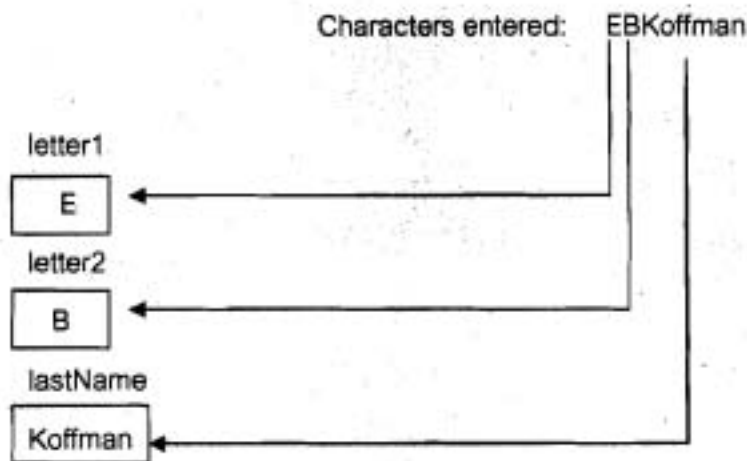
### คำสั่งในการรับข้อมูล(Input Statement)

คำสั่ง `cin >> miles ;` เป็นการรับข้อมูลจำนวนจริงจากผู้ใช้นำมาเก็บในตัวแปร `miles` โดยเครื่องหมาย `>>` เรียกว่า extraction operator ซึ่งจะนำข้อมูลจากการป้อนของผู้ใช้จากรูปที่ 2.2 ผู้ใช้ป้อนค่า 30.5 นำไปเก็บในตัวแปร `miles`



รูปที่ 2.2 แสดงการปฏิบัติงานของคำสั่ง `cin >> miles ;`

การรับข้อมูลจากผู้ใช้งานสามารถรับชนิดข้อมูลอื่นๆได้ อาทิเช่น `int` , `float` , `double` , `char` , `string` เป็นต้น จากรูปที่ 2.3 แสดงการปฏิบัติงานของคำสั่ง `cin >> letter1 >> letter2 >> lastName ;` กำหนดให้ ตัวแปร `letter1` และ `letter2` เป็นข้อมูลชนิด `char` และ `lastName` เป็นข้อมูลชนิด `string`



รูปที่ 2.3 แสดงการปฏิบัติงานของคำสั่ง `cin >> letter1 >>letter2 >>lastName ;`

The Input (Extraction) Operator >>	
<b>รูปแบบ:</b>	<code>&gt;&gt; data-store ;</code>
<b>ตัวอย่าง:</b>	<code>cin &gt;&gt; age &gt;. firstinitial;</code>
<b>ความหมาย :</b>	รับข้อมูลทางแป้นพิมพ์นำมาเก็บใน data-store จากตัวอย่าง ผู้ใช้ป้อนข้อมูล 2 ค่าโดยเว้นช่องว่างเพื่อแยกข้อมูล เมื่อป้อนข้อมูลครบกดแป้น Enter ข้อมูลที่ป้อนจะเก็บใน data-store ที่ระบุไว้ตามลำดับ

### คำสั่งแสดงผล(Program Output)

การปฏิบัติการของโปรแกรมสามารถแสดงผลลัพธ์ออกมาพิมพ์ทางจอภาพ โดยใช้คำสั่ง `cout` ดังนี้ `cout << " The distance in kilometers is " << kms << endl ;` ซึ่งผลลัพธ์ของการทำงานจะพิมพ์ข้อความในเครื่องหมาย " " ออกมาทางจอภาพ ตามด้วย ค่าของข้อมูล kms

ออกมาทางจอภาพ ในที่นี้พิมพ์ The distance in kilometers is 16.09 โดย endl เป็นการขึ้นบรรทัดใหม่ ซึ่งคำสั่งนี้มีรูปแบบดังนี้

The Output (Insertion) Operator <<	
<b>รูปแบบ:</b>	<code>&lt;&lt; data-element ;</code>
<b>ตัวอย่าง:</b>	<code>cout &lt;&lt; " My height in inches is " &lt;&lt; height &lt;&lt; endl ;</code>
<b>ความหมาย :</b>	data-element เป็นได้ทั้งตัวแปร, ออปเจ็ก, หรือข้อความ ซึ่งจะแสดงผลออกมาทางจอภาพ ในที่นี้จะแสดงข้อความ My height in inches is ตามด้วยค่าที่เก็บในตัวแปร height และขึ้นบรรทัดใหม่จากคำสั่ง endl

### คำสั่ง return

คำสั่งสุดท้ายของฟังก์ชัน main คือ return 0 ; โดยค่า 0 จะส่งให้กับระบบปฏิบัติการเพื่อแสดงว่าโปรแกรมสามารถประมวลผลได้โดยปราศจากข้อผิดพลาด

Return Statement	
<b>รูปแบบ:</b>	<code>return expression ;</code>
<b>ตัวอย่าง:</b>	<code>return 0 ;</code>
<b>ความหมาย :</b>	ส่งค่า expression ไปให้กับระบบปฏิบัติการ

## 2.5 รูปแบบทั่วไปของโปรแกรม C++

---

จากตัวอย่างที่ผ่านมาทำให้เราทราบถึงคำสั่งพื้นฐานในการสั่งให้เครื่องปฏิบัติการซึ่งเป็นการประมวลผลคำนวณค่าของข้อมูลที่ผู้ใช้ป้อนผ่านทางแป้นพิมพ์ซึ่งสามารถสรุปถึงรูปแบบของโปรแกรมภาษา C++ ได้ดังนี้

### รูปแบบทั่วไปของโปรแกรม C++

---

```
compiler directives
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    declaration statements
```

```
    executable statements
```

```
}
```

---

การเขียนคำสั่งภาษา C++ นั้นผู้เขียนสามารถเขียนได้หลายลักษณะ ตัวแปลภาษาสามารถตรวจสอบไวยากรณ์ภาษาได้โดยแยกคำสั่งแต่ละคำสั่งด้วยเครื่องหมาย ; ซึ่งเป็นอักขระที่บ่งบอกถึงจุดจบของคำสั่ง เราสามารถเขียน 2 คำสั่งในบรรทัดเดียวกันได้ เช่น

```
cout << "Enter the distance in miles : " ; cin >> miles ;
```

นอกจากนี้ต้องระวังการพิมพ์ค่าต่างๆที่อยู่ในคำสั่งเพราะตัวแปลภาษาจะแยกแต่ละคำด้วยช่องว่าง ดังนั้นควรมีการเว้นช่องว่างระหว่างค่าต่างๆรวมทั้งตัวกระทำต่างๆเหล่านี้ เช่น \* , / , + , - , = ซึ่งต้องมีการเว้นช่องว่างก่อนและหลังเพื่อแยกคำออกจากกัน การเขียนคำสั่งต่างๆในโปรแกรมควรมีการเอียงรวมทั้งการตั้งชื่อตัวแปรต่างๆควรให้มีความหมายและสอดคล้องกับข้อมูลที่ใช้ เพื่อให้ง่ายต่อการทำความเข้าใจ แต่สำหรับอักขระพิเศษต่างๆบางตัวไม่อนุญาตให้มีช่องว่างระหว่างอักขระ ได้แก่ // , / , \* / , >> , << ผู้เขียนโปรแกรมต้องระมัดระวังด้วย จากตัวอย่างที่ผ่านมาจะเห็นว่าภายในโปรแกรมจะมีคำอธิบายในโปรแกรม ซึ่งคำอธิบายนี้จะมีเป็น

ระยะๆ เป็นเครื่องมือที่ช่วยให้โปรแกรมเมอร์เข้าใจโปรแกรมได้ง่ายขึ้น ซึ่งบรรยายวัตถุประสงค์ของโปรแกรม ความหมายของตัวแปรต่างๆ และอธิบายถึงการปฏิบัติงานต่างๆ ในโปรแกรม โดยทั่วไปแต่ละโปรแกรมควรมีส่วนหัวโปรแกรมที่ระบุถึงรายละเอียดต่างๆ อันประกอบด้วย

- ชื่อโปรแกรมเมอร์
- วันที่หรือรุ่น(version) ในการพัฒนา
- วัตถุประสงค์ของโปรแกรม

### ตัวอย่างที่ 2.3 แสดงคำอธิบายโปรแกรม

---

```
/*  
* Programmer : William bell Date completed: May 9, 2003  
* Instructor: Janet Smith Class : CIS62  
*  
* Calculates and displays the area and circumference of a  
* circle  
*/
```

---

จากตัวอย่างที่ 2.3 เป็นการเขียนคำอธิบายโปรแกรมในส่วนของหัวโปรแกรม โดยอยู่ในเครื่องหมาย /\* \*/ ซึ่งทำให้ทราบถึงข้อมูลของโปรแกรมเบื้องต้นของผู้เขียนโปรแกรมและวัตถุประสงค์ของโปรแกรม นอกจากนี้ควรมีคำอธิบายโปรแกรมในแต่ละขั้นตอนของการปฏิบัติงาน ในกรณีที่โปรแกรมยวขึ้นโปรแกรมเมอร์สามารถเข้าใจการทำงานของโปรแกรมได้ง่ายขึ้น ซึ่งการเขียนคำอธิบายภายในโปรแกรมสามารถเขียนได้ตามตัวอย่างดังนี้

```
// Convert the distance to kilometers.
```

```
kms = KM_PER_MILE * miles ;
```

หรืออธิบายโดยละเอียดมากขึ้นได้ดังนี้

```
// Multiply KM_PER_MILE by miles and store result in kms.
```

```
kms = KM_PER_MILE * miles ;
```

## 2.6 นิพจน์คณิตศาสตร์(Arithmetic Expression)

การแก้ปัญหาทางคณิตศาสตร์ส่วนมาก โปรแกรมเมอร์ต้องเขียนนิพจน์คณิตศาสตร์เพื่อใช้ในการแก้ปัญหาเพื่อให้เครื่องคอมพิวเตอร์ปฏิบัติงานอย่างถูกต้อง ภาษาโปรแกรมต่างๆมีตัวกระทำทางคณิตศาสตร์ที่แตกต่างกันไป สำหรับภาษา C++ มีตัวกระทำทางคณิตศาสตร์ตามตารางที่ 2.1 ดังนี้

ตารางที่ 2.1 ตัวกระทำทางคณิตศาสตร์

ตัวกระทำ	ความหมาย	ตัวอย่าง	ผลของการทำงาน
+	addition	$5 + 2$	7
		$5.0 + 2.0$	7.0
-	subtraction	$5 - 3$	3
		$5.0 - 3.0$	3.0
*	multiplication	$5 * 2$	10
		$5.0 * 2.0$	10.0
/	division	$5.0 / 2.0$	2.5
		$5 / 2$	2
%	remainder	$5 \% 2$	1

จากตารางที่ 2.1 แสดงตัวกระทำต่างๆ กรณีที่ข้อมูลที่กระทำเป็นเลขจำนวนเต็ม การ + , - , \* , / ผลลัพธ์มีชนิดข้อมูลเป็นเลขจำนวนเต็ม แต่ถ้าข้อมูลที่กระทำเป็นเลขทศนิยม ผลลัพธ์มีชนิดข้อมูลเป็นเลขทศนิยมเช่นเดียวกัน ยกเว้นตัวกระทำ % ที่จะกระทำกับข้อมูลที่เป็น int เท่านั้น และผลลัพธ์เป็น int ได้โดยตรงเนื่องจาก ตัวกระทำ % เป็นการหาเศษของเลขจำนวนเต็มเศษที่ได้จากการหาร ดังตารางที่ 2.2 และ ตารางที่ 2.3

ตารางที่ 2.2 แสดงการหารของเลขจำนวนเต็ม

$3 / 15 = 0$	$18 / 3 = 6$
$15 / 3 = 5$	$0 / 4 = 0$

$$\begin{array}{r}
 7 / 2 \\
 \downarrow \\
 3 \\
 2 \overline{) 7} \\
 \underline{6} \\
 1 \rightarrow 7 \% 2
 \end{array}
 \qquad
 \begin{array}{r}
 299 / 100 \\
 \downarrow \\
 2 \\
 100 \overline{) 299} \\
 \underline{200} \\
 99 \rightarrow 299 \% 100
 \end{array}$$

### ตารางที่ 2.3 แสดงการหาเศษของการหารเลขจำนวนเต็ม

$3 \% 15 = 3$	$5 \% 3 = 2$
$15 \% 3 = 0$	$15 / 6 = 3$
$8 \% 5 = 1$	$5 \% 5 = 0$

เศษที่ได้จากการหารนั้นต้องน้อยกว่าตัวหารเสมอ อย่างเช่น  $m \% 100$  เศษของการหารต้องมีค่าอยู่ในช่วง 0 ถึง 99 ซึ่งสามารถเขียนเป็นสูตรได้ดังนี้

$$m = (m / n) * n + (m \% n)$$

กำหนดให้  $m = 7$  และ  $n = 2$  ดังนั้น

$$\begin{aligned}
 7 &= (7 / 2) * 2 + (7 \% 2) \\
 &= 3 * 2 + 1 \\
 &= 7
 \end{aligned}$$

กำหนดให้  $m = 299$  และ  $n = 100$  ดังนั้น

$$\begin{aligned}
 299 &= (299 / 100) * 100 + (299 \% 100) \\
 &= 2 * 100 + 99 \\
 &= 299
 \end{aligned}$$

ในกรณีที่ค่าของข้อมูลผสมกันระหว่างเลขจำนวนเต็มและเลขทศนิยม ภาษา C++ จะมีการปรับเปลี่ยนข้อมูลที่เป็นเลขจำนวนเต็มเป็นเลขทศนิยม ทำให้ผลลัพธ์ที่ได้เป็นเลขทศนิยม เช่น

$$5 / 2.5 = 2.0$$



ในกรณีที่นิพจน์คณิตศาสตร์มีตัวกระทำหลายๆตัว เช่น  $X + Y / Z$  ปัญหาก็คือตัวกระทำใดทำก่อนหรือหลัง เครื่องหมาย+ กระทำก่อน เครื่องหมาย/ หรือเครื่องหมาย / กระทำก่อน เครื่องหมาย + กันแน่ ภาษา C++ มีกฎเกณฑ์ในการประมวลผลซึ่งโปรแกรมเมอร์ต้องเข้าใจถึงลำดับของการปฏิบัติการเพื่อใช้ในการสร้างนิพจน์คณิตศาสตร์ที่ถูกต้องดังนี้

### กฎการคำนวณของนิพจน์คณิตศาสตร์

1. เมื่อพบวงเล็บ () จะกระทำวงเล็บในสุดก่อน และกระทำเป็นลำดับจนถึงวงเล็บนอกสุด
2. กรณีที่มีตัวกระทำหลายๆตัวมีลำดับของการกระทำดังนี้
 

unary +, -	กระทำก่อน
*, /, %	กระทำต่อมา
+, -	สุดท้าย
3. กรณีที่มีลำดับของการกระทำอยู่ในระดับเดียวกันเช่น +, -, จะกระทำตัวที่พบก่อน หรือกระทำจากซ้ายไปขวา

### ตัวอย่างที่ 2.4 สูตรของการหาพื้นที่ของวงกลม

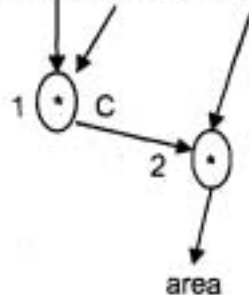
สูตรการหาพื้นที่ของวงกลมคือ  $a = \pi r^2$

เมื่อเปลี่ยนเป็นนิพจน์คณิตศาสตร์ของภาษา C++ ได้ดังนี้

```
area = PI * radius * radius ;
```

ลำดับของการปฏิบัติงานเป็นดังนี้

```
area = PI * radius * radius ;
```

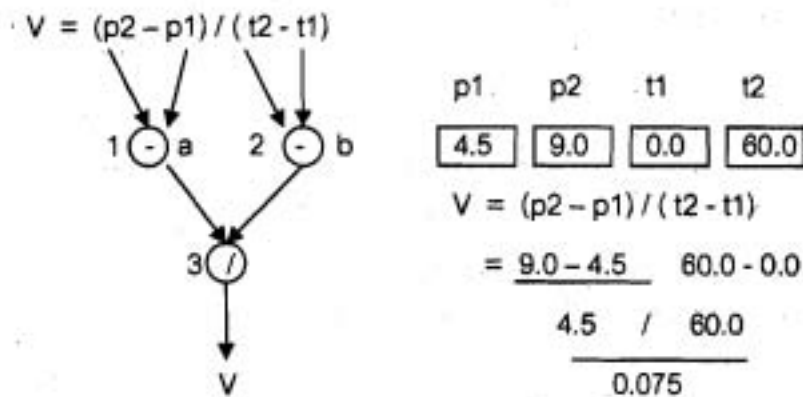


$$\begin{array}{r}
 \text{area} = \text{PI} * \text{radius} * \text{radius}; \\
 \underline{3.14159 * 2.0 * 2.0} \\
 6.28318 * 2.0 \\
 \underline{\hspace{1.5cm}} \\
 12.56636
 \end{array}$$

การทำงานของนิพจน์จะกระทำเครื่องหมาย \* ตัวแรกก่อน เนื่องจากมีเครื่องหมาย \* สองตัวจะกระทำเครื่องหมายที่พบก่อน ต่อจากนั้นนำผลลัพธ์มากระทำเครื่องหมาย \* ตัวถัดไป จนกระทั่งได้ผลลัพธ์เก็บในตัวแปร area

### ตัวอย่างที่ 2.5

ต้องการหาค่าของ  $V = (p2 - p1) / (t2 - t1)$  ;

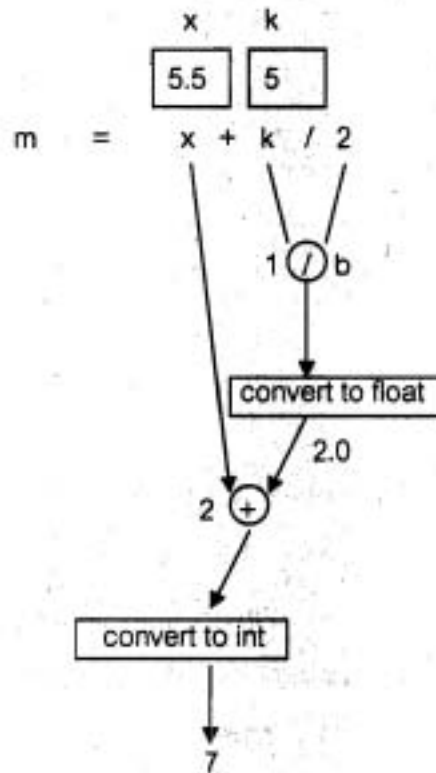


การทำงานจะกระทำนิพจน์ที่อยู่ในวงเล็บ โดยกระทำวงเล็บที่พบก่อนเป็นอันดับแรก กระทำวงเล็บที่เหลือเป็นลำดับถัดไป นำผลลัพธ์มากระทำการหารเป็นลำดับที่สาม ผลลัพธ์ที่ได้เก็บในตัวแปร V ดังภาพ

## ตัวอย่างที่ 2.6

### ต้องการหาค่าของ $m = x + k / 2$

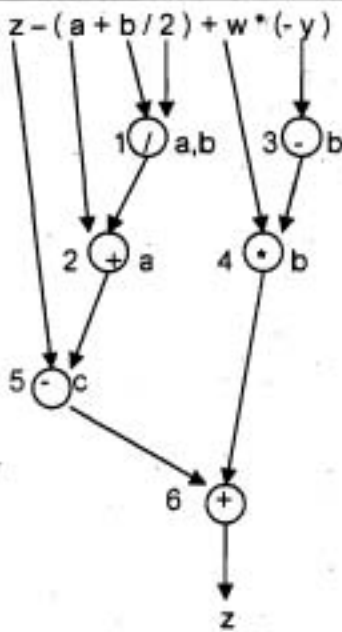
การทำงานของคำสั่งนี้กำหนดให้ตัวแปร  $m$  และ  $k$  เก็บข้อมูลเป็นเลขจำนวนเต็ม ตัวแปร  $x$  เก็บข้อมูลเป็นเลขจำนวนจริง การปฏิบัติการเป็นค่านวณค่าของข้อมูลต่างชนิดกัน ดังนั้นต้องมีการแปลงค่าเพื่อให้สามารถคำนวณค่าต่างๆได้ ดังรูป



จากรูปข้างต้น การปฏิบัติการเริ่มจากการกระทำหารข้อมูลก่อนโดยนำค่า  $5 / 2$  มีค่าเท่ากับ  $2.5$  เป็นเลขจำนวนเต็ม การคำนวณต่อไปคือการบวก แต่ค่าของ  $x$  มีค่าเป็น  $5.5$  ดังนั้นจึงมีการแปลงค่าเลขจำนวนเต็ม ให้เป็นเลขทศนิยม ดังนั้น ค่า  $2$  จึงปรับเปลี่ยนเป็นค่า  $2.0$  แทน กระทำ การบวกได้ผลลัพธ์เป็น  $7.5$  เป็นเลขทศนิยม จากนั้นจึงดำเนินการกำหนดให้ผลลัพธ์ของการปฏิบัติการเก็บในตัวแปรที่เป็นเลขจำนวนเต็ม ดังนั้นจึงต้องมีการแปลงค่าเลขทศนิยมให้เป็นเลขจำนวนเต็มโดยตัดเศษทิ้ง ผลลัพธ์สุดท้ายค่าของตัวแปร  $m$  มีค่าเท่ากับ  $7$

## ตัวอย่างที่ 2.7

ต้องการหาค่าของ  $z - (a + b/2) + w * (-y)$



$z$	$a$	$b$	$w$	$y$
8	3	9	2	-5
$z - (a + b/2) + w * (-y)$				
		$9/2$		$-5$
	$3 + 4$		$2 * 5$	
$8 -$	$7$		$10$	
$1$			$+ 10$	
			$11$	

การทำงานของนิพจน์คณิตศาสตร์นี้เริ่มจากการคำนวณค่าในวงเล็บที่พบก่อนเป็นอันดับแรก โดยคำนวณตัวกระทำ / ก่อน และตามด้วยตัวกระทำ + ต่อจากนั้นกระทำในวงเล็บต่อไปซึ่งเป็น unary - ซึ่งได้ผลลัพธ์จากการกระทำมีค่าเท่ากับ +5 ตัวกระทำที่เหลือคือเครื่องหมาย \* และ + ซึ่งต้องกระทำเครื่องหมาย \* ก่อนเครื่องหมาย + ซึ่งเมื่อกำหนดให้ค่า  $z = 8, a = 3, b = 9, w = 2$  และ  $y = -5$  ผลลัพธ์จากการทำงานมีค่าเท่ากับ 11 ดังภาพ

## กรณีศึกษา : การแปลงสกุลเงิน

กรณีศึกษานี้เป็นการศึกษาถึงการปฏิบัติการกับเลขจำนวนเต็มที่มีการใช้ตัวกระทำ / และ %

**ปัญหา** ต้องการแลกเงิน nickels และ pennies เป็น dollar และ cents

**วิเคราะห์** การแก้ปัญหาต้องวิเคราะห์ถึงข้อมูลเข้าว่ามีอะไรบ้าง ผลลัพธ์ที่ต้องการมีอะไร และอัตราของการแลกเปลี่ยนเงินสกุลที่ต้องการเป็นอย่างไร ผลจากการวิเคราะห์สามารถสรุปได้ดังนี้

## DATA REQUIREMENTS

---

### Problem Input

String name // ชื่อของผู้ที่ต้องการแลกเปลี่ยนเงิน  
int nickels // จำนวนเงิน nickels  
int pennies // จำนวนเงิน pennies

### Problem Output

int dollars // จำนวนเงิน dollars  
int change // จำนวนเงินที่เป็นเศษไม่ถึง dollar

### Additional Program Variable

int totalCents // จำนวนเงินที่แลกเปลี่ยนเป็นเงิน cents ทั้งหมด

## FORMULAS

---

1 dollar = 100 pennies

1 nickel = 5 pennies

**ออกแบบ** จากการวิเคราะห์ถึงปัญหาต่างๆ สามารถออกแบบอัลกอริทึมในการแปลงค่าสกุลเงินจาก nickels และ pennies เป็น dollar และ cents ได้ดังนี้

## ALGORITHM

---

1. อ่านชื่อของผู้ใช้ทางแป้นพิมพ์
2. อ่านจำนวนเงินของ nickels และ จำนวนของ pennies ทางแป้นพิมพ์
3. นำค่าที่รับมาจากข้อ 2 คำนวณ โดยแปลงสกุลเงินให้เป็น cents ทั้งหมด
4. นำค่าที่คำนวณได้จากข้อ 3 หาจำนวนของ dollars และ เศษที่เหลือ
5. แสดงผลที่ได้จากข้อที่ 4 ออกทางจอภาพ

สำหรับขั้นตอนที่ 3 และ ขั้นตอนที่ 4 สามารถแสดงขั้นตอนที่ละเอียดได้อีกดังนี้

### ขั้นตอนที่ 3 Refinement

3.1 ผลรวมของเงิน cents ทั้งหมด เกิดจากการนำค่า nickels \* 5 + pennies

totalCents = 5 \* nickels + pennies ;

### ขั้นตอนที่ 4 Refinement

4.1 เงิน dollars เกิดจากการนำผลรวมของเงิน cents ทั้งหมด หารด้วย 100

dollars = totalcents / 100 ;

change = totalCents % 100;

4.2 เศษที่เหลือ เกิดจากการเศษที่เหลือจากการนำผลรวมของเงิน cents ทั้งหมด หารด้วย 100

## เขียนโปรแกรม

```
// File : coins.cpp
```

```
// Determines the value of a coin collection
```

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
int main ( )
```

```
{
```

```
    string name;                //input: name
```

```
    int pennies;                //input: count of pennies
```

```
    int nickels;                //input: count of nickels
```

```
    int dollars;                //output: value of coins in dollars
```

```
    int change;                 //output: value of coins in cents
```

```
    int totalCents;             //total cents represented
```

```
    // Read in your name,
```

```
    cout << " Enter your first name: ";
```

```
    cin >> name;
```

```

// Read in the count of nickels and pennies.
cout << " Enter the number of nickels: ";
cin >> nickels;
cout << " Enter the number of pennies: ";
cin >> pennies;

// Compute the total value in cents.
totalCents = 5 * nickels + pennies;

// Find the value in dollars and change.
dollars = totalCents / 100;
change = totalCents % 100 ;

// Display the value in dollars and change.
cout << " Good work " << name << "!" << endl;
    << " Your collection is worth " << dollars << " dollars and "
    << change << " cents. " << endl;

return 0;
}

```

### ทดสอบ

```

Enter your first name : Sally
Enter the number of nickels : 30
Enter the number of pennies : 77
Good work Sally!
Your collection is worth 2 dollars and 27 cents.

```

## 2.7 ข้อผิดพลาดในการเขียนโปรแกรม

---

ข้อผิดพลาดที่เกิดขึ้นในการเขียนโปรแกรมนั้นเราเรียกว่า bugs ซึ่งอาจเป็นเหตุการณ์ที่โปรแกรมทำงานได้ผลลัพธ์ที่ผิดพลาดไปจากความเป็นจริง หรือโปรแกรมทำงานไม่ได้เพราะเขียนไวยากรณ์ผิดไม่ผ่านตั้งแต่การแปลภาษา จะด้วยเหตุผลใดก็ตาม โปรแกรมเมอร์ต้องสืบค้นให้ได้ว่าผิดพลาดที่จุดไหน อย่างไร เพื่อแก้ไขให้โปรแกรมที่พัฒนาขึ้นมีความถูกต้อง มีประสิทธิภาพ และมีความน่าเชื่อถือ โดยทั่วไปความผิดพลาดสามารถแบ่งได้เป็น 3 ชนิดใหญ่ๆ ได้แก่ ความผิดไวยากรณ์ภาษา(syntax) , ความผิดในขณะปฏิบัติการ(run) และความผิดทางตรรก(logic)

### Syntax Errors

---

เป็นความผิดพลาดที่สามารถตรวจพบในขณะแปลโปรแกรม เป็นความผิดทางไวยากรณ์ภาษา ซึ่งการเขียนคำสั่งโปรแกรมไม่สมบูรณ์ มีการพิมพ์ตกหล่น หรือผิดพลาด ซึ่งตัวแปลภาษาตรวจพบและแสดงข่าวสารให้ผู้เขียนโปรแกรมทราบ จากตัวอย่างที่ 2.8 แสดงถึงการเขียนโปรแกรมที่ผิดพลาด

### ตัวอย่างที่ 2.8

#### การแปลงระยะทางไมล์ให้เป็นกิโลเมตร ผิดไวยากรณ์ภาษา

---

```
//milesError.cpp
//Converts distance from miles to kilometers.
#include <iostream>
using namespace std;
int main()                                //start of main function
{
    const float KM_PER_MILE=1.609; //1.609 km in a mile
    float miles                      //Input : distance in miles
        kms ;                        //output: distance in kilometers
```



```

//Get the distance in miles.
cout << " Enter the distance in miles : ";
cin >> miles;
// Convert the distance to kilometers.
kms = KM_PER_MILE * miles;
// Display the distance to kilometers.
cout << " The distance in kilometers is " << kms << endl;
return 0;
}

```

เมื่อนำโปรแกรมนี้ไปแปลจะเกิดความผิดพลาดทางไวยากรณ์ภาษาปรากฏดังนี้

```

[C++ Error] miles.cpp (12) : E2141 Declaration syntax error.
[C++ Error] miles.cpp (15) : E2380 Unterminated string or character constant.
[C++ Error] miles.cpp (16) : E2379 Statement missing ; .
[C++ Error] miles.cpp (19) : E2451 Undefined symbol 'kms'.
[C++ Warning] miles.cpp (25) : W8080 'miles' declared but never used.
[C++ Warning] miles.cpp (12) : W8004 'KM_PER_MILE' is assigned a value
that is never used.

```

การแก้ไขเพียงแต่โปรแกรมเมอร์นำเมาส์คลิกที่บรรทัดที่ข่าวสารความผิดพลาด โปรแกรมภาษา จะไปยังบรรทัดนั้นๆในโปรแกรม ความผิดพลาดในบรรทัดแรกของโปรแกรมนี้คือบรรทัดที่ 12 ในโปรแกรม

```

float miles //input : distance in miles
เมื่อพิจารณาแล้วเห็นว่าส้อมใส่เครื่องหมาย , เป็นการแยกตัวแปรออกจากกัน แก้ไขดังนี้
float miles , //input : distance in miles
บรรทัดต่อไปคือบรรทัดที่ 15
cout << " Enter the distance in miles : ";
พิจารณาแล้วความผิดพลาดคือส้อมใส่เครื่องหมาย " เมื่อจบข้อความ แก้ไขเป็นดังนี้
cout << " Enter the distance in miles : " ;

```

สำหรับความผิดพลาดอื่นๆที่เกิดขึ้น รวมทั้งข้อความเตือนนั้นก็มีผลมาจากความผิดพลาดในส่วนแรกๆนั่นเอง ซึ่งส่งผลให้ตัวแปลภาษาไม่รู้จักตัวแปร เมื่อแก้ไขในสองบรรทัดดังกล่าวข้างต้นจะส่งผลให้โปรแกรมมีไวยากรณ์ภาษาที่ถูกต้อง

### Run-Time-Errors

เป็นความผิดพลาดที่เกิดขึ้นในขณะที่ปฏิบัติการ (run) โดยไม่สามารถปฏิบัติงานตามคำสั่งโปรแกรมได้ อาทิเช่น มีการหารค่าด้วย 0 เป็นต้น เมื่อมีความผิดพลาดลักษณะนี้เกิดขึ้น โปรแกรมจะหยุดการประมวลผลทันที พร้อมแสดงบรรทัดที่ผิดพลาดรวมทั้งข้อความที่อธิบายถึงความผิดพลาดที่เกิดขึ้นให้กับโปรแกรมเมอร์ได้ทราบ เพื่อให้โปรแกรมเมอร์แก้ไขต่อไป

### ตัวอย่างที่ 2.9 แสดงความผิดพลาดที่เกิดขึ้นขณะโปรแกรมทำงาน

```
// File: runtimeError.cpp
// Provides a possible " division by zero " run-time error.
#include <iostream>
using namespace std;
int main ( )
{
    int first , second;
    float temp ,ans;
    cout << " Enter 2 integers : " ;
    cin >> first >> second;
    temp = second / first;
    ans = first / temp;
    cout << " The result is " << ans << endl;
    return 0 ;
}
```

การทำงานของโปรแกรมนี้ เกิดความผิดพลาดในกรณีที่ใช้ป้อนข้อมูลเลขจำนวนเต็ม 2 จำนวน โดยให้เลขจำนวนเต็มตัวแรกมีค่ามากกว่าเลขจำนวนเต็มตัวที่สอง เพราะมีผลให้ค่าของตัวแปร

temp มีค่าเท่ากับ 0 จากการนำ second / first และเมื่อนำค่า 0 ไปหารค่าใดๆก็ตาม คอมพิวเตอร์ไม่สามารถหาค่าให้ได้ ทำให้โปรแกรมหยุดการทำงานและแจ้งข่าวสารให้ผู้ใช้ทราบถึงความผิดพลาดที่เกิดขึ้นโดยปรากฏดังนี้คือ "Floating point division by zero"

### Logic Errors

เป็นความผิดพลาดของผลลัพธ์ ซึ่งมีผลมาจากข้อบกพร่องที่มในการแก้ปัญหาโปรแกรม ซึ่งยากที่จะค้นพบ ต้องใช้วิจารณ์ญาณ ความเชี่ยวชาญของโปรแกรมเมอร์ในการค้นหาและแก้ไข ความผิดพลาดนี้อาจเกิดจาก การเขียนนิพจน์ทางคณิตศาสตร์ไม่ถูกต้อง มีการใช้เงื่อนไขในการทำงานผิดพลาด หรือการลิมกำหนดค่าเริ่มต้น สำหรับปฏิบัติงานเป็นต้น เพื่อป้องกันความผิดพลาดที่เกิดขึ้นโปรแกรมเมอร์ต้องตรวจสอบข้อบกพร่องให้ถูกต้องก่อนการเขียนโปรแกรมจะเป็นการดีที่สุด

## สรุป

1. ภาษา C++ ประกอบด้วยคำสั่ง ในแต่ละคำสั่งประกอบด้วยคำต่างๆที่เรียงต่อกัน โดยทุกคำสั่งสิ้นสุดที่เครื่องหมายเซมิโคลอน
2. คำในภาษา C++ แบ่งออกเป็น 2 ชนิดคือคำเฉพาะ และคำที่ผู้ใช้กำหนดขึ้นใหม่
3. ภาษา C++ ทุกโปรแกรมต้องมีฟังก์ชัน main ซึ่งเป็นจุดเริ่มต้นในการปฏิบัติงานของโปรแกรม คำสั่งในการปฏิบัติงานของฟังก์ชันจะอยู่ภายในเครื่องหมาย { }
4. Declaration statement เป็นคำสั่งที่ประกาศตัวแปรทั้งหมดที่ใช้ในฟังก์ชันให้คอมไพเลอร์ได้รู้จัก เพื่อสามารถปฏิบัติงานกับข้อมูลต่างๆได้
5. Reserved words หรือ keyword เป็นคำสงวนที่มีความหมายใช้งานเฉพาะในภาษา C++ ซึ่งโปรแกรมเมอร์ไม่สามารถนำไปใช้ในวัตถุประสงค์อื่นๆได้
6. Identifier คือคำใหม่ที่ผู้ใช้กำหนดขึ้น อาจเป็นชื่อของตัวแปร หรือชื่อฟังก์ชัน การกำหนดขึ้นมาต้องถูกต้องตามกฎเกณฑ์ของภาษา โดยขึ้นต้นด้วยตัวอักษรภาษาอังกฤษหรือเครื่องหมายขีดล่าง ตัวต่อไปเป็นตัวเลขหรือขีดล่างหรือตัวอักษรภาษาอังกฤษก็ได้ แต่ห้ามเป็นอักขระพิเศษอื่นๆ และตัวอักษรตัวเล็กตัวใหญ่ถือว่าเป็นคนละตัวกัน
7. ความแตกต่างระหว่างการประกาศตัวแปร และการกำหนดค่าคงที่ คือ การประกาศตัวแปรนั้นข้อมูลต่างๆที่เก็บสามารถเปลี่ยนแปลงค่าได้ตามลำดับของการปฏิบัติงานของโปรแกรม แต่การกำหนดค่าคงที่นั้น ค่าที่เก็บในตัวแปรจะไม่สามารถเปลี่ยนแปลงค่าได้
8. Assignment Statement เป็นคำสั่งในการกำหนดค่าให้แก่ตัวแปรใดๆ ใช้สัญลักษณ์ = เป็นการนำค่าใดๆทางด้านขวาของเครื่องหมายไปให้แก่ตัวแปรทางซ้าย  
$$Kms = 10.0 * 5 ;$$
9. การรับและแสดงผลนั้นภาษา C++ ได้เตรียมไว้ให้ในคลาส iostream โดยกำหนดให้ออปเจกต์ cin และ >> เป็นตัวกระทำในการรับข้อมูลทางแป้นพิมพ์ และออปเจกต์ cout และ << เป็นตัวกระทำที่ใช้ในการพิมพ์ผลลัพธ์ออกทางจอภาพ
10. รูปแบบทั่วไปของโปรแกรม C++  
compiler directive  
using namespace std;

```

int main( )
{
    declaration statements
    executable statements
}

```

11. คำอธิบายโปรแกรมจะเขียนอยู่ในเครื่องหมาย /\* \*/ หรือ ในแต่ละบรรทัดขึ้นต้นด้วย // ก็ได้
12. กฎการคำนวณของนิพจน์คณิตศาสตร์ เมื่อพบวงเล็บเล็กจะกระทำในสุดก่อน ถ้าพบเครื่องหมาย unary + หรือ - จะกระทำเป็นลำดับต่อมา ถ้าพบ \* , / , % จะกระทำเป็นลำดับต่อมา กรณีเป็นเครื่องหมายบวกและลบ จะกระทำเป็นลำดับสุดท้าย ถ้าเครื่องหมายอยู่ในระดับเดียวกัน จะกระทำเครื่องหมายที่พบก่อนเสมอ
13. ความผิดพลาดในการเขียนโปรแกรมที่พบโดยทั่วไปแบ่งเป็น 3 ชนิดคือ ความผิดไวยากรณ์ภาษา , ความผิดในขณะปฏิบัติการ และความผิดพลาดทางตรรก

## แบบฝึกหัด

---

1. พิจารณาชื่อ(identifier)ต่อไปนี้ ถ้าตั้งถูกต้องตามกฎหมายก็ให้กาเครื่องหมายถูก ถ้าผิดให้กาเครื่องหมายผิด

income.....	two fold.....	hour*rate.....	myproram .....
ex1.cpp.....	ex1 .....	_abc .....	MAIN .....
pi .....	variable .....	return .....	tom's .....
main .....	income#1.....	const .....	1time

2. จงหาผลลัพธ์ของนิพจน์ต่อไปนี้

2.1  $X = 25.0 * 3.0 / 2.5 ;$   
2.2  $X = 100 - 35 / 4 ;$   
2.3  $X = 100.0 - 35 / 4.0 ;$   
2.4  $X = 299 / 23 * 80 + 633 \% 100 ;$   
2.5  $X = 299 / ( (23 * 80) + 633 ) \% 100 ;$   
2.6  $X = (299 / 23) * (80 + 633) \% (100 - 23) ;$   
2.7  $X = 15.0 + 522 / 2 ;$

3. จงเขียนโปรแกรมในการแปลงอุณหภูมิจากองศาฟาเรนไฮน์ให้เป็น องศาเซลเซียส

### DATA REQUIREMENTS

---

#### Problem Input

int fahrenheit // temperature in degree Fahrenheit

#### Problem Output

float celsius // temperature in degree Celsius

#### Formula

$celsius = (5/9) * (fahrenheit - 32)$

4. จงเขียนโปรแกรมอ่านจำนวนตัวเลข 2 จำนวนทางแป้นพิมพ์ จงหาผลบวก , ผลต่าง , ผลคูณ และผลหาร ของเลขสองจำนวนนี้

#### DATA REQUIREMENTS

---

##### Problem Input

int x,y // two item

##### Problem Output

int sum // sum of x and y

int difference // difference of x and y

int product // product of x and y

int quotient // quotient of a divided by y

5. จงเขียนโปรแกรมอ่านน้ำหนักเป็นปอนด์ คำนวณให้เป็นกิโลกรัม และกรัม  
( 1 ปอนด์ มีค่าเท่ากับ 0.453592 กิโลกรัม หรือ 453.59237 กรัม )
6. จงเขียนโปรแกรมพิมพ์รูปสี่เหลี่ยมใดๆ ขนาด 6 x 6
7. จงเขียนโปรแกรมอ่านความยาวและความกว้างของสี่เหลี่ยมซึ่งเป็นขนาดเนื้อที่หนึ่ง ถ้าเจ้าของที่ต้องการตัดหญ้าโดยมีอัตรา 1 วินาที ต่อ 2 ตารางเมตร เนื้อที่นี้จะใช้เวลาเท่าใดจึงจะตัดหญ้าเสร็จ
8. จงเขียนโปรแกรมรับค่าเศษ และ ส่วน ของค่าเศษส่วน 2 จำนวน นำมาหาผลคูณของค่าเศษส่วนทั้งสอง และหาด้วยว่าคิดเป็นกี่เปอร์เซ็นต์
9. เขียนโปรแกรมอ่านจำนวนของปีที่ไดโนเสาร์ยังมีชีวิตอยู่ เช่น triceratops มีชีวิตอยู่เมื่อ 145 ล้านปีมาแล้ว หรือ brontosaurus มีชีวิตอยู่เมื่อ 182 ล้านปีมาแล้ว โดยนำมาแปลงให้เป็นจำนวนวัน ( 1 ปีมีค่าเท่ากับ 365.25 วัน)
10. จงเขียนโปรแกรมรับจำนวนวินาทีระหว่างค่า 0 ถึง 18000 ทางแป้นพิมพ์ จงแปลงจำนวนวินาทีนี้ให้เป็นจำนวนชั่วโมง จำนวนนาที และจำนวนวินาที
11. จงเขียนโปรแกรมในการรับเลข 3 จำนวนใดๆทางแป้นพิมพ์ จงหาค่าเฉลี่ยของตัวเลขชุดนี้
12. แก้ไขโปรแกรมในข้อที่ 8 โดยหาผลบวกของเศษส่วนทั้งสอง พิมพ์ออกมาทางจอภาพ