
CT 2101

Procedural Programming

โดย ผศ.ดร.อุไร ทองหัวไผ่

เนื้อหารายวิชา

นำความรู้เกี่ยวกับการสร้างอัลกอริทึมและการเขียนโปรแกรมมาใช้งานโดยเน้นการออกแบบและการสร้างโปรแกรมขนาดใหญ่ การออกแบบโปรแกรมแบบบนลงล่าง แบบล่างขึ้นบน ศึกษาโครงสร้างข้อมูลเบื้องต้น เช่น รายการโยง สแตก คิว ทรี และแฟ้ม การทำข้อมูลเป็นนามธรรม เพื่อสร้างส่วนประกอบโปรแกรมในรูปส่วนจำเพาะ การเขียนโปรแกรมเรียกตัวเอง การเรียงลำดับและการค้นหาข้อมูลแบบทั่วไป ปัญหาทางทฤษฎีของวิทยาการคอมพิวเตอร์ที่เกี่ยวข้องกับการเขียนโปรแกรม ความเข้าใจเบื้องต้นเกี่ยวกับวิศวกรรมซอฟต์แวร์ เช่น คุณสมบัติของโปรแกรมที่ดี วิธีการทดสอบความถูกต้องของโปรแกรม วิธีแก้ไขความผิดพลาดในโปรแกรม วิธีการทำงานแบบเป็นทีมและวิธีการบริหารทีมงานเขียนโปรแกรมเบื้องต้น

PR.

- เนื้อหารายวิชานี้เชื่อมโยงมาจากรายวิชา COS1101 การออกแบบโปรแกรม โดยผู้เรียนต้องมีพื้นฐานความรู้ในการออกแบบโปรแกรมเป็นอย่างดี สามารถออกแบบแนวคิดในการแก้ปัญหาโปรแกรมเบื้องต้นได้

วัตถุประสงค์

- มุ่งเน้นให้นักศึกษามีความรู้ ความเข้าใจ และมีความสามารถในการจัดการด้านข้อมูลเบื้องต้นได้
- มีความสามารถในการวิเคราะห์ระบบงานและเขียนโปรแกรมได้อย่างถูกต้องตามหลักไวยากรณ์ภาษา
- ผู้เรียนสามารถทดสอบและแก้ไขความผิดพลาดของโปรแกรมก่อนนำไปใช้งานได้
- นอกจากนี้ผู้เรียนสามารถทำงานเป็นทีม และมีวิธีการบริหารทีมงานเขียนโปรแกรมเบื้องต้นได้

- Structure Programming
 - Sequence
 - Condition
 - Repetition
- Array
- Record
- Recursive
- File

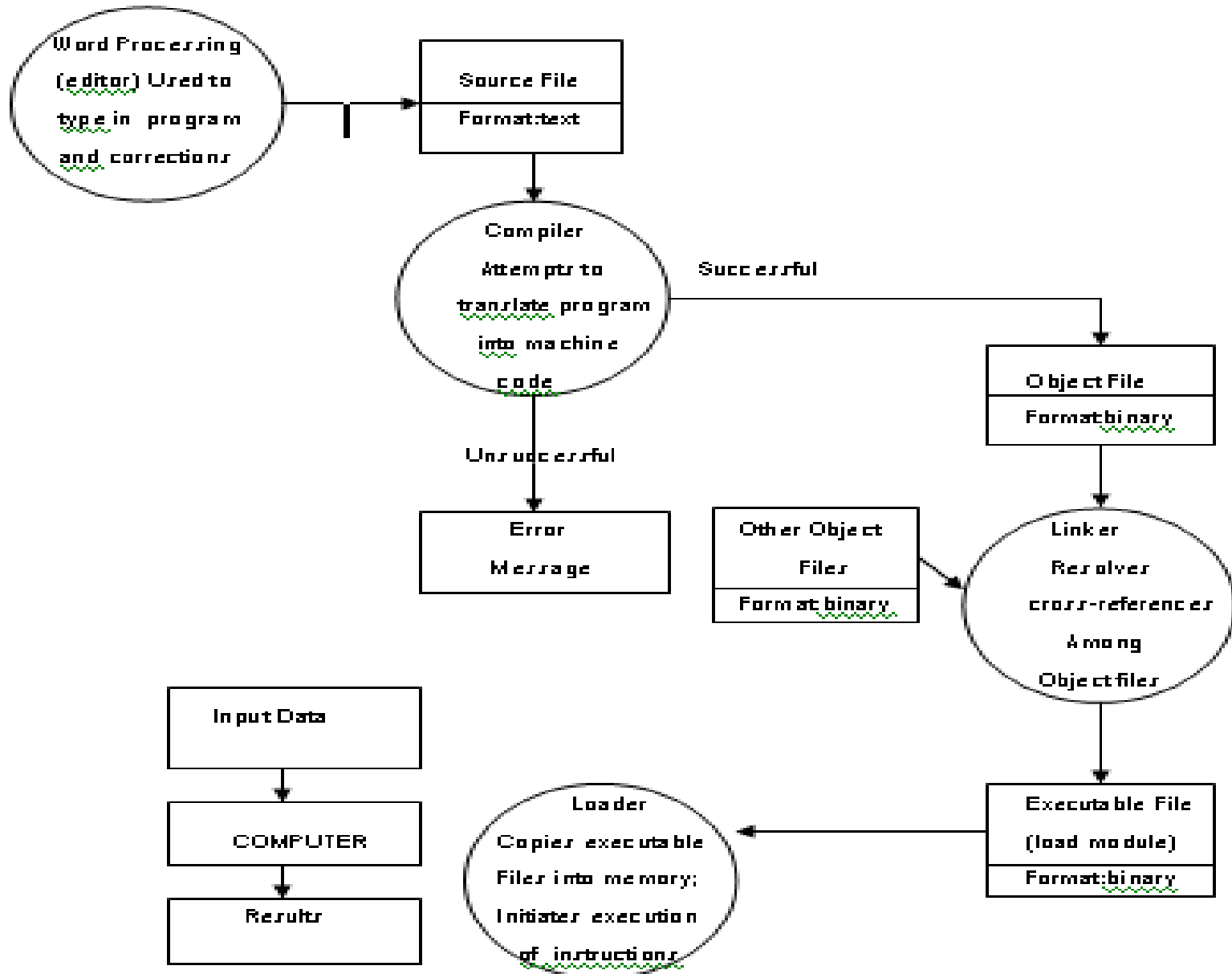
เนื้อหา(ต่อ)

- String
- Pointer
- Data structure
 - Stack
 - Queue
 - Linked-list
 - Tree

คอมพิวเตอร์คืออะไร?

- อุปกรณ์ที่ใช้สำหรับประมวลผล
- องค์ประกอบหลักที่สำคัญของคอมพิวเตอร์ประกอบด้วย 2 ส่วนคือ ฮาร์ดแวร์ และซอฟต์แวร์
- ฮาร์ดแวร์คืออุปกรณ์อิเล็กทรอนิกส์ต่างๆที่ใช้ในการปฏิบัติการ
- ซอฟต์แวร์เป็นโปรแกรมหรือชุดคำสั่งที่ใช้ในการติดต่อกับฮาร์ดแวร์นั่นเอง
- โดยการติดต่อกันใช้ภาษาโปรแกรมที่เครื่องคอมพิวเตอร์เข้าใจ เรียกว่าภาษาเครื่อง (machine language) ซึ่งประกอบด้วยเลขฐานสอง

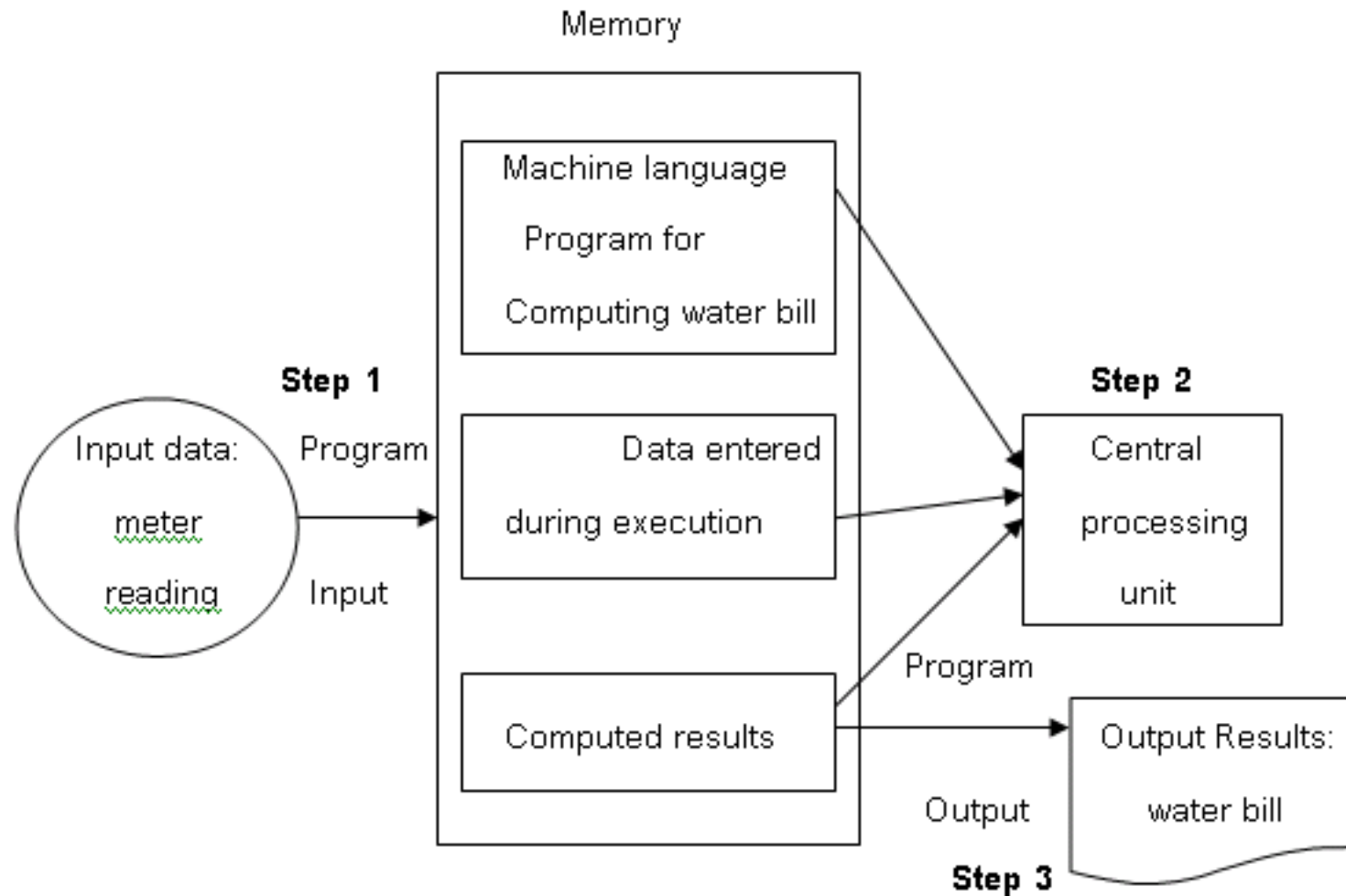
การประมวลผลโปรแกรมภาษาระดับสูง



การประมวลผลโปรแกรมคิดค่าน้ำ

- ขั้นตอนที่**1** โปรแกรมภาษาเครื่องในการคิดค่าน้ำ และข้อมูลที่ใช้ในการประมวลผลคือจำนวนของยูนิตน้ำที่ใช้ จะถูกเก็บในหน่วยความจำ
- ขั้นตอนที่**2** โปรแกรมจะทำการประมวลผลโดยนำจำนวนของยูนิตของน้ำมาคำนวณหาค่าน้ำที่ต้องจ่าย และเก็บผลลัพธ์ไว้ในหน่วยความจำ
- ขั้นตอนที่**3** นำผลลัพธ์ที่คำนวณเก็บไว้ในหน่วยความจำ แสดงผลออกมาเป็นบิลค่าน้ำ

การประมวลผลโปรแกรมคิดค่าน้ำ



software development method

- ระบุถึงความต้องการ (problem)
- วิเคราะห์ปัญหา (analysis)
- ออกแบบอัลกอริทึมเพื่อแก้ปัญหา (design)
- สร้างโปรแกรม (implementation)
- ทดสอบและตรวจสอบความสมบูรณ์ของโปรแกรม (testing)
- บำรุงรักษาและแก้ไขโปรแกรม (maintenance)

ระบุถึงความต้องการ (problem)

- ผู้พัฒนาสามารถสอบถามได้จากบุคคลต่างๆที่เกี่ยวข้อง
- ความต้องการที่ดีต้องชัดเจน ไม่คลุมเครือ มีความจำเป็น สามารถทวนสอบหรือตรวจสอบได้ ติดตามได้

วิเคราะห์ปัญหา (analysis)

- ต่อจากนั้นจะนำความต้องการต่างๆเหล่านี้มาวิเคราะห์ถึงหนทางในการแก้ปัญหา
- โดยนำปัญหาต่างๆเหล่านี้มากำหนดเป็นนิยาม(**abstraction**)ของ
 - ข้อมูลเข้า
 - ข้อมูลนำออก
 - การคำนวณ
 - เงื่อนไขในการปฏิบัติการต่างๆ

ออกแบบอัลกอริทึมเพื่อแก้ปัญหา (design)

- โดยมีการแบ่งปัญหออกเป็นปัญหาย่อยๆเพื่อให้ง่ายต่อการพัฒนาซึ่งเรียกว่า **top-down design** หรือ **divide and conquer**
- ในแต่ละปัญหาย่อยจะกำหนดเป็นลำดับขั้นตอนในการแก้ปัญหาโดยละเอียดเรียกว่า **algorithm refinement**
- **Tools**
 - Flowchart
 - Structure Chart
 - Stepwise refinements (Pseudo code)

สร้างโปรแกรม (implementation)

- เขียนเป็นภาษาโปรแกรมให้ถูกต้องตามไวยากรณ์ของภาษานั้นๆ
- ภาษา C++
 - Syntax
 - Semantic
 - Algorithm
 - Goal
 - Context

ทดสอบและตรวจสอบความสมบูรณ์ของโปรแกรม (testing)

- ทดสอบโดยกำหนดข้อมูลทดสอบหลายๆชุด เพื่อให้แน่ใจว่าการทำงานถูกต้อง และสามารถนำไปปฏิบัติการในการแก้ปัญหาได้
- Test Case
 - Path
 - Statements
 - Valid data
 - Invalid data

บำรุงรักษาและแก้ไขโปรแกรม (maintenance)

- ป้องกันและแก้ไขข้อผิดพลาดที่อาจเกิดขึ้นในขณะทำงาน
 - Corrective maintenance
 - Adaptive maintenance
 - Perfective maintenance
 - Preventive maintenance

กรณีศึกษา : การแปลงไมล์เป็นกิโลเมตร

- ปัญหา ต้องการแปลงระยะทางที่เป็นไมล์ให้เป็นกิโลเมตร
- วิเคราะห์ ต้องการข้อมูลว่าอะไรคือข้อมูลเข้า อะไรคือผลลัพธ์ที่ต้องการ และต้องมีการประมวลผลอย่างไร ในที่นี้ข้อมูลเข้าคือระยะทางที่เป็นไมล์ ผลลัพธ์ที่ต้องการคือระยะทางที่เป็นกิโลเมตร สำหรับการประมวลผลเป็นการคำนวณที่ต้องทราบว่า **1** ไมล์มีค่าเท่ากับ **1.609** กิโลเมตร ต่อจากนั้น นิยามข้อมูลต่างๆเหล่านี้เพื่อให้สามารถเก็บในหน่วยความจำของเครื่องคอมพิวเตอร์ได้

วิเคราะห์

DATA REQUIREMENTS

Problem Input

miles ระยะทางเป็นไมล์

Problem Output

kms ระยะทางเป็นกิโลเมตร

Formular

1 ไมล์ = 1.609 กิโลเมตร

ALGORITHM WITH REFINEMENTS

1. รับระยะทางเป็นไมล์
2. แปลงระยะทางจากไมล์เป็นกิโลเมตร
 - 2.1 ระยะทาง 1 ไมล์มีค่าเท่ากับ 1.609 กิโลเมตร
3. แสดงระยะทางเป็นกิโลเมตร

เขียนโปรแกรม

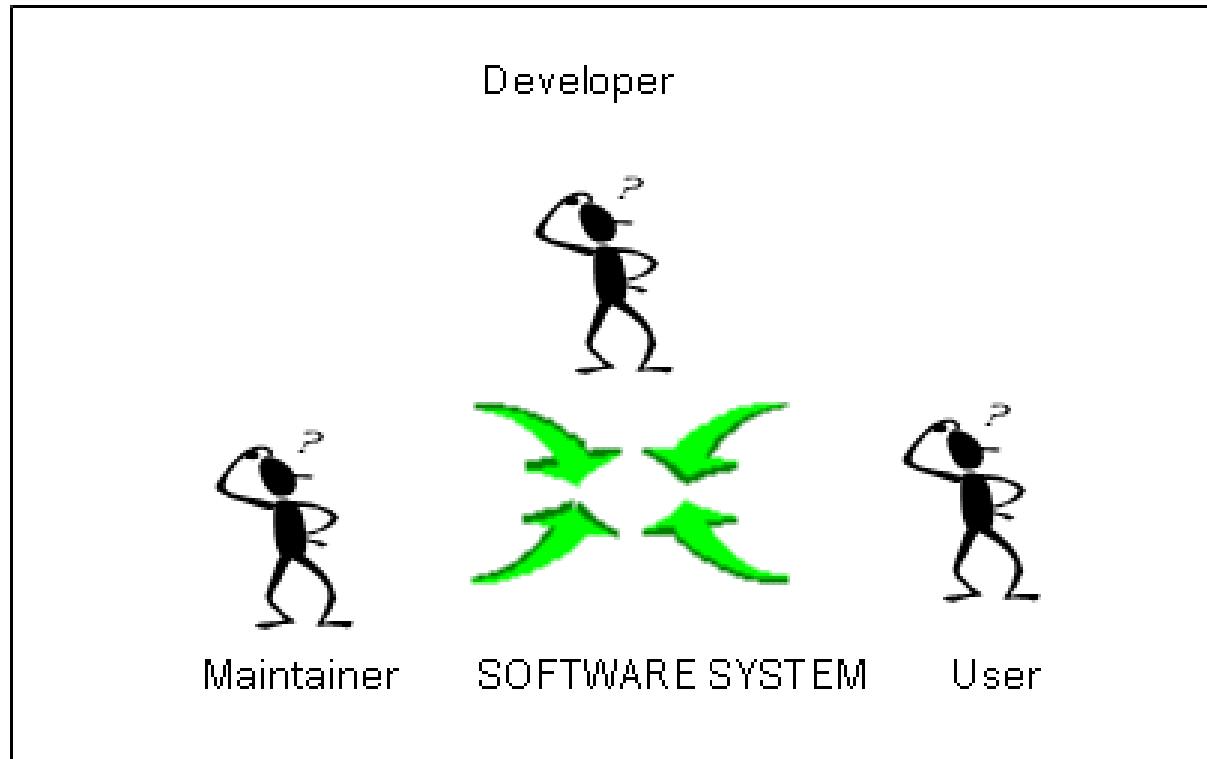
```
//miles.cpp
//Coverts distance in miles to kilometers.
#include <iostream>
using namespace std;
int main()                                //start of main function
{
    const float KM_PER_MILE=1.609         //1.609 km in a mile
    float miles,                          //input : distance in miles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

ทดสอบ

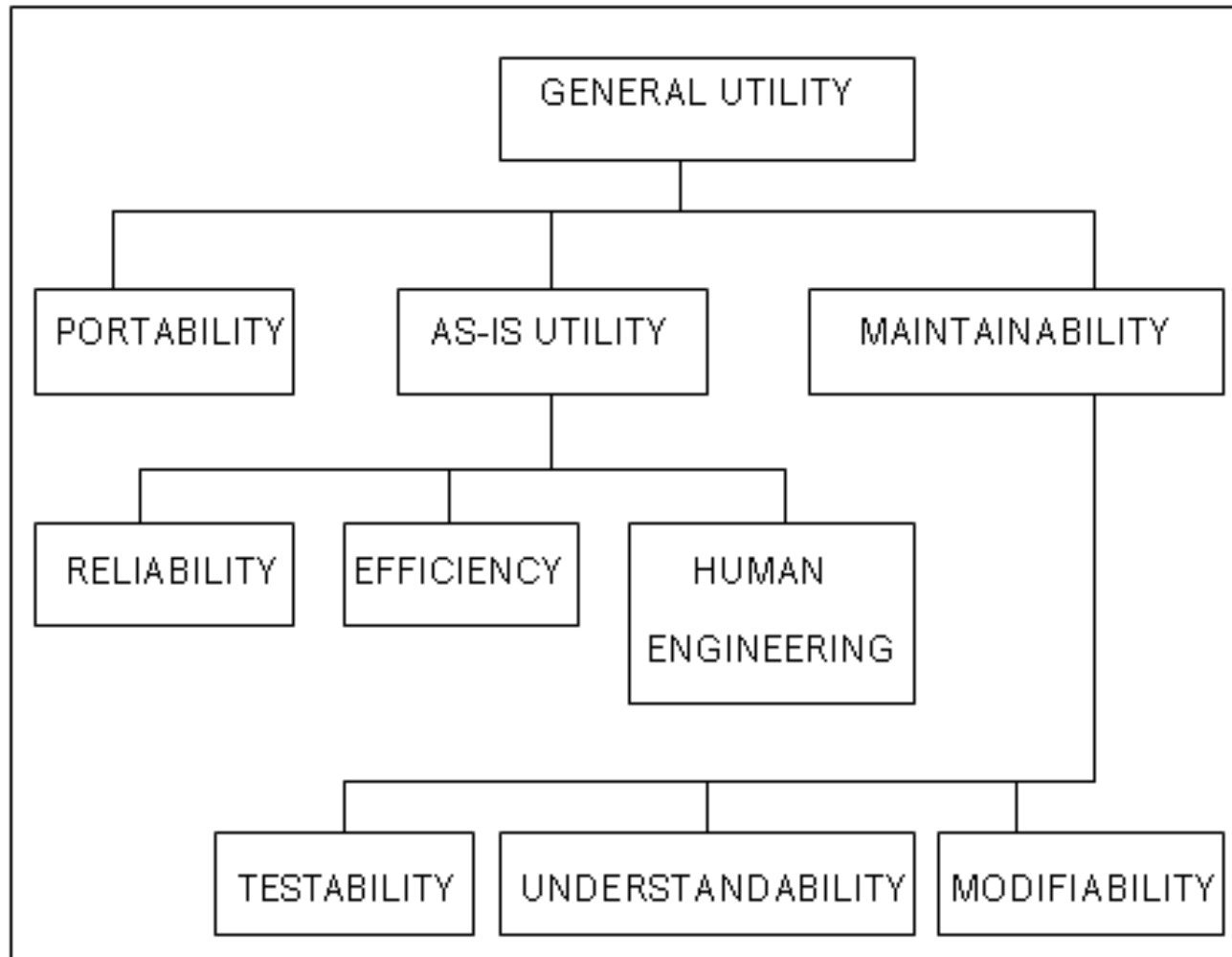
Enter the distance in miles: 10.0

The distance in kilometer is 16.09

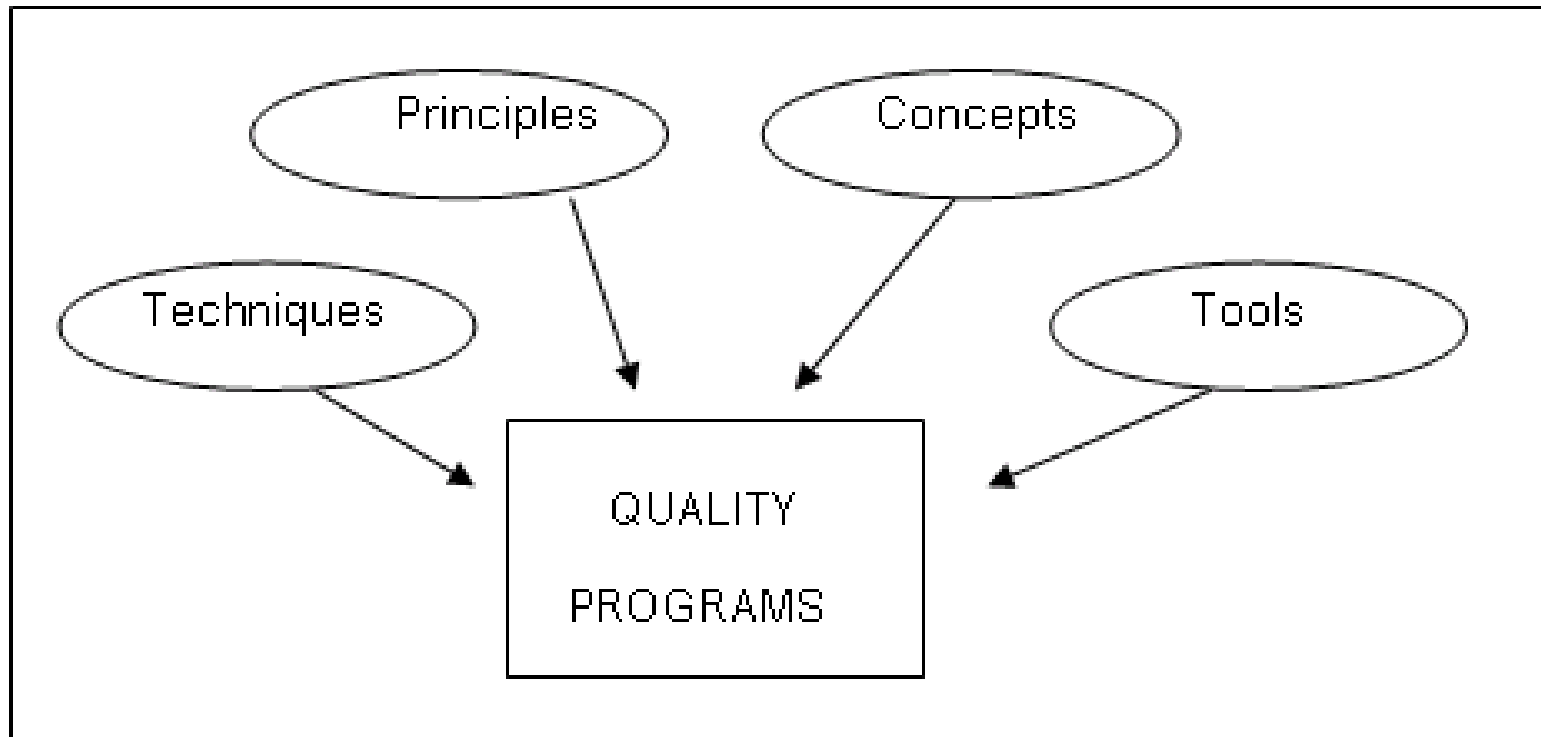
คุณภาพของซอฟต์แวร์



คุณภาพของซอฟต์แวร์



How to Build Quality Programs



รูปแบบทั่วไปของโปรแกรม C++

compiler directives

using namespace std;

void main()

{

 declaration statements;

 executable statements;

}

```
//miles.cpp
```

```
//Coverts 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;
```

```
}  
-----
```

Comment

รูปแบบ: // comment
 /* comment */

ตัวอย่าง: // miles.cpp
 /* Coverts distance in miles to kilometers.*/

ความหมาย : โดย // เป็นจุดเริ่มต้นของคำอธิบายในกรณีที่มีคำอธิบายเพียงบรรทัดเดียว โดยอาจใช้ /* เป็นจุดเริ่มต้น และ */ เป็นจุดสิ้นสุดก็ได้ ในกรณีที่มีคำอธิบายหลายบรรทัด

Compiler Directive #include

รูปแบบ: #include <filename>

ตัวอย่าง: #include <iostream>

ความหมาย : ก่อนการแปล บรรทัดนี้จะถูกแทนที่ด้วย standard library filename โดยทั่วไปคำสั่งนี้จะอยู่ต้นโปรแกรม และต้องกำหนดก่อนการเรียกใช้ฟังก์ชันภายในคลาสนั้น

หมายเหตุ : คำสั่ง compiler directive จะไม่จบด้วยเครื่องหมาย ;

using namespace

รูปแบบ: using namespace *region* ;

ตัวอย่าง: using namespace std;

ความหมาย : เป็นคำสั่งที่แสดงว่าโปรแกรมมีการใช้ออปเจ็กซึ่งกำหนดใน namespace ที่ระบุโดย *region* โดยทั่วไปบรรทัดนี้จะตามจากบรรทัดที่

#include เติมข

main Function definition

รูปแบบ: int main()
{
 function body
}

ตัวอย่าง: int main()
{
 cout << "Enjoy C++ " << endl;
 return 0;
}

ความหมาย : เป็นจุดเริ่มต้นของการปฏิบัติงาน โดยกระทำงานตามคำสั่ง
ในเครื่องหมาย { } และต้องมีคำสั่ง return 0 เพื่อส่งค่ากลับไปให้ระบบปฏิบัติการ
เมื่อการปฏิบัติการสิ้นสุดลง

คำเฉพาะ (Reserved Words)

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

ชื่อ(Identifiers)

- ขึ้นต้นด้วยตัวอักษรภาษาอังกฤษ หรือเครื่องหมาย underscore (จีดล่าง) ภาษา C++ ถือว่าอักษรภาษาอังกฤษตัวเล็กกับตัวใหญ่เป็นคนละตัวกัน เช่น Rate , rate , RATE เป็นต้น
- ตัวต่อไปอาจเป็นตัวเลข ตัวอักษร หรือ จีดล่างก็ได้ เช่น Letter1 , letter1 , Letter_1
- ห้ามใช้คำเฉพาะ เป็นชื่อที่แทนความหมายอื่นใด

ชนิดของข้อมูล(Data Types)

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

String Class

- เป็นคลาสที่ภาษา C++ เตรียมไว้ให้กับโปรแกรมเมอร์สำหรับการใช้ข้อมูลที่เป็นข้อความ โดย ข้อความคือการนำอักขระมาเรียงต่อกันเป็นกลุ่มคำ อาทิ เช่น “A” “1234” “Enter the distance in miles “ เป็นต้น
- ผู้ใช้สามารถอ่าน บันทึก เปรียบเทียบ รวม หรือค้นหา ข้อความต่างๆในการปฏิบัติงานได้
- โดยต้องมีคำสั่ง `#include <string>` ในโปรแกรมก่อนใช้งาน

Variable and Object Declarations

รูปแบบ: type identifier-list ;

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

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

Constant Declaration

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

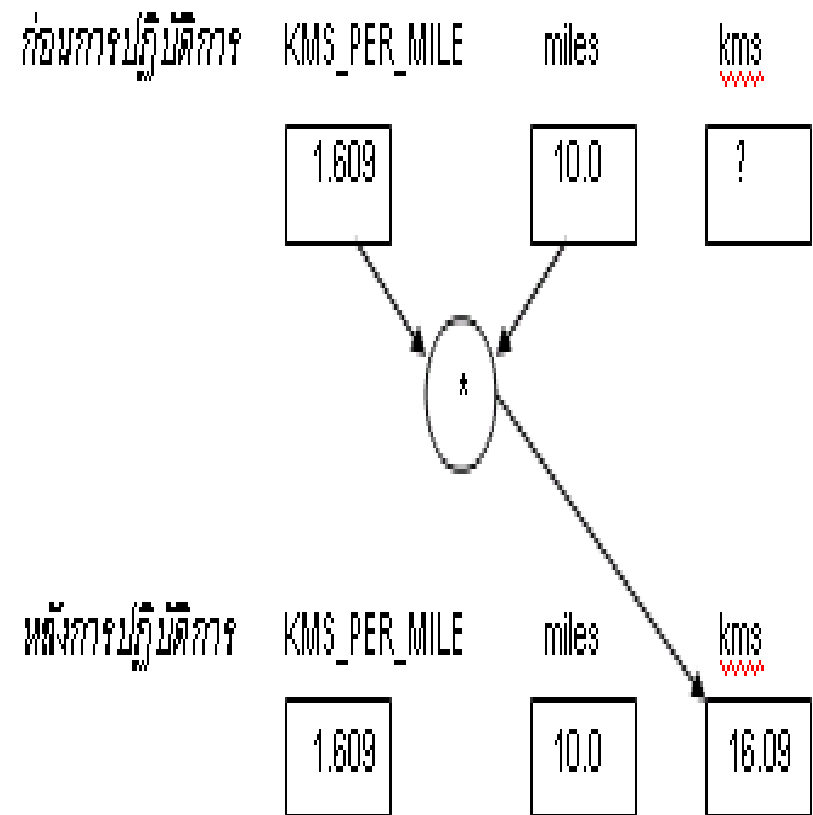
ตัวอย่าง: const float PI = 3.14159;

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

คำสั่งกำหนดค่า (Assignment Statement)

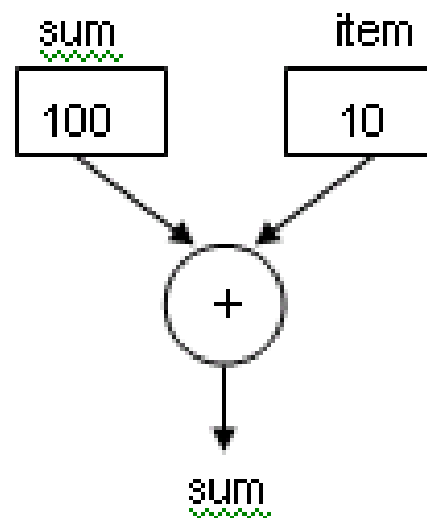
ใช้สัญลักษณ์เท่ากับ = เป็นการให้แก่ตัวแปร

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



ค่าของตัวแปรต่างๆนั้นจะแปรเปลี่ยนไปตามคำสั่งที่ประมวลผล โดยเก็บค่าได้เพียงค่าเดียวและ
เป็นค่าที่ได้รับการปรับเปลี่ยนครั้งสุดท้ายเท่านั้น เช่นคำสั่ง sum = sum + item ;

ก่อนการปฏิบัติการ



The Input (Extraction) Operator >>

รูปแบบ: >> *data-store*;

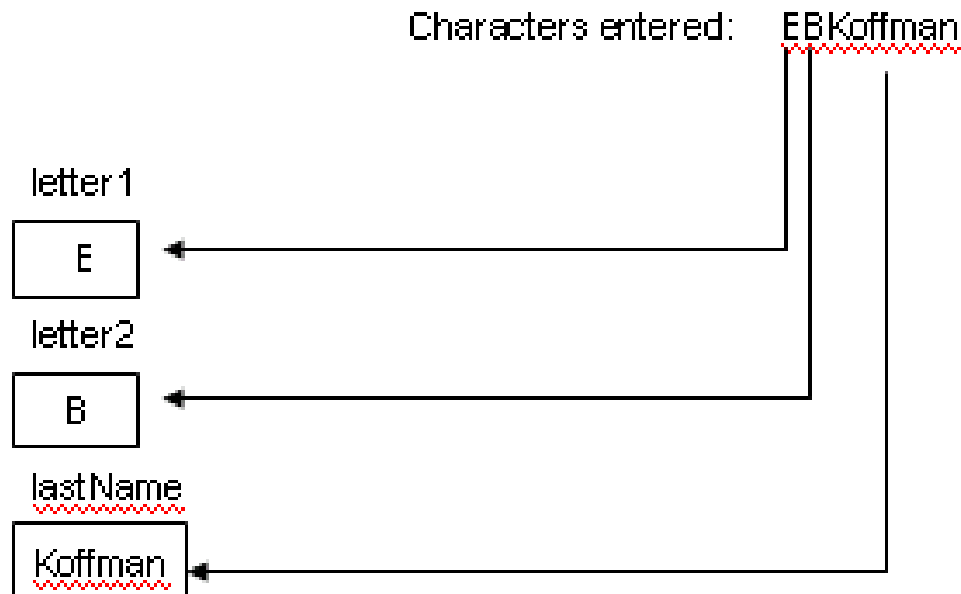
ตัวอย่าง: cin >> age > .firstInitial;

ความหมาย : รับข้อมูลทางแป้นพิมพ์นำมาเก็บใน *data-store* จากตัวอย่าง
ผู้ใช้ป้อนข้อมูล 2 ค่าโดยเว้นช่องว่างเพื่อแยกข้อมูล เมื่อป้อนข้อมูลครบกดแป้น
Enter ข้อมูลที่ป้อนจะเก็บใน *data-store* ที่ระบุไว้ตามลำดับ

การทำงานของคำสั่ง `cin >> miles`



คำสั่ง `cin >> letter1 >> letter2 >> lastName ;`
กำหนดให้ ตัวแปร `letter1` และ `letter2` เป็นข้อมูลชนิด `char`
และ `lastName` เป็นข้อมูลชนิด `string`



The Output (Insertion) Operator <<

รูปแบบ: << data-element;

ตัวอย่าง: cout << " My height in inches is " << height << endl ;

ความหมาย : data-element เป็นได้ทั้งตัวแปร, ออปเจ็ก, หรือข้อความ ซึ่งจะแสดงผลออกมาทางจอภาพ ในที่นี้จะแสดงข้อความ My height in inches is ตามด้วยค่าที่เก็บในตัวแปร height และขึ้นบรรทัดใหม่จากคำสั่ง endl

Return Statement

รูปแบบ: `return expression ;`

ตัวอย่าง: `return 0 ;`

ความหมาย : ส่งค่า expression ไปให้กับระบบปฏิบัติการ

ตัวอย่างการพิมพ์ข้อความทักทาย

```
//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++.
```

นิพจน์คณิตศาสตร์ (Arithmetic Expression)

ตารางที่ 2.1 ตัวกระทำทางคณิตศาสตร์

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

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

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

ตารางที่ 2.3 แสดงการหาเศษของการหารเลขจำนวนเต็ม

$$3 \% 15 = 3$$

$$5 \% 3 = 2$$

$$15 \% 3 = 0$$

$$15 / \underline{6} = 3$$

$$8 \% 5 = 1$$

$$5 \% 5 = 0$$

กฎการคำนวณของนิพจน์คณิตศาสตร์

- เมื่อพบวงเล็บ $()$ จะกระทำวงเล็บในสุดก่อน และกระทำเป็นลำดับจนถึงวงเล็บนอกสุด

- กรณีที่มีตัวกระทำหลายตัวมีลำดับของการกระทำดังนี้

unary $+$, $-$

กระทำก่อน

$*$, $/$, $\%$

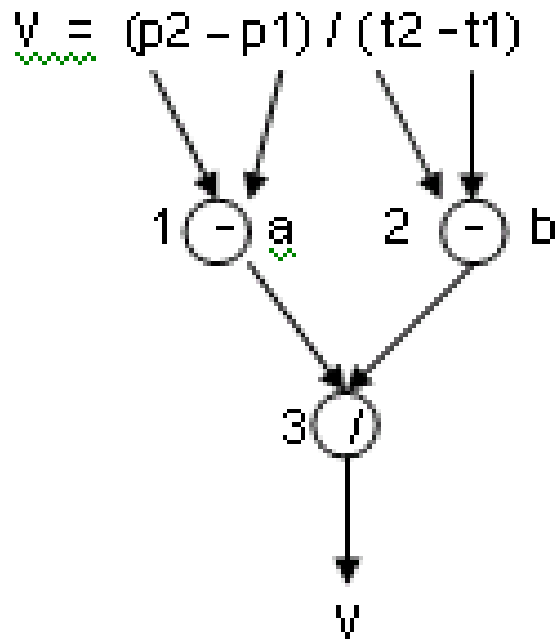
กระทำต่อมา

$+$, $-$

สุดท้าย

- กรณีที่มีลำดับของการกระทำอยู่ในระดับเดียวกันเช่น $+$, $-$ จะกระทำตัวที่พบก่อน หรือกระทำจากซ้ายไปขวา

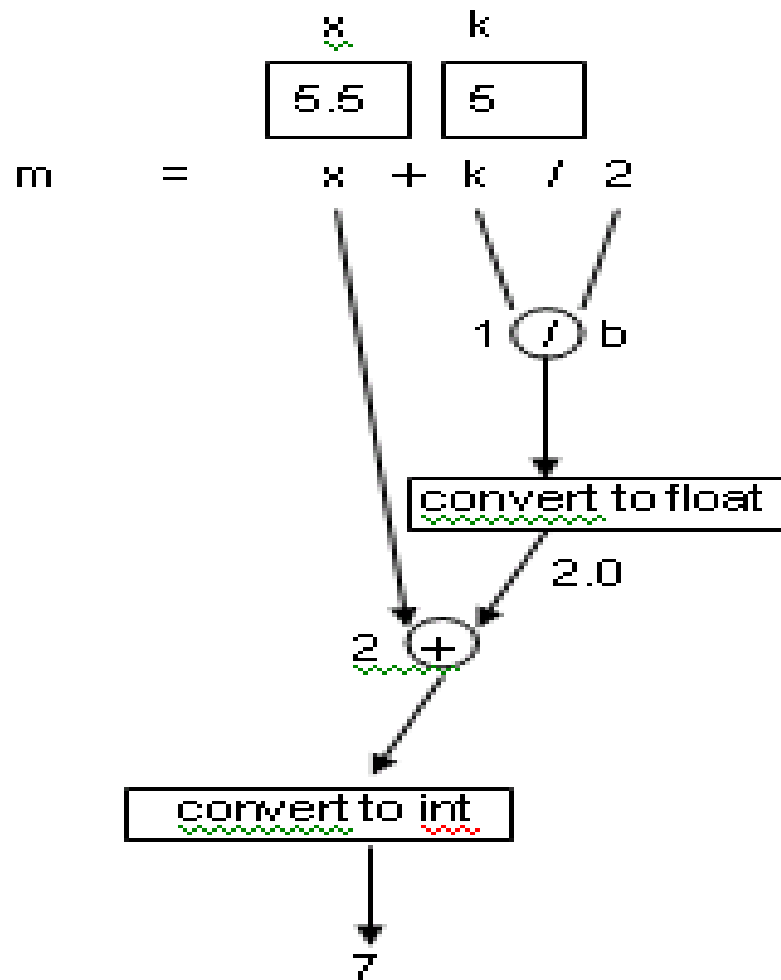
ต้องการการหาค่า $V = (p_2 - p_1) / (t_2 - t_1) ;$



p1	p2	t1	t2
4.5	9.0	0.0	60.0

$$\begin{aligned} V &= (p_2 - p_1) / (t_2 - t_1) \\ &= \frac{9.0 - 4.5}{60.0 - 0.0} \\ &= \frac{4.5}{60.0} \\ &= 0.075 \end{aligned}$$

ต้องการการหาค่าของ $m = x + k / 2$



ข้อผิดพลาดในการเขียนโปรแกรม

- **Syntax Errors**
- **Run-Time-Errors**
- **Logic Errors**

Syntax Errors

- เป็นความผิดพลาดที่สามารถตรวจพบในขณะที่แปลโปรแกรม เป็นความผิดพลาดไวยากรณ์ภาษา ซึ่งการเขียนคำสั่งโปรแกรมไม่สมบูรณ์ มีการพิมพ์ตกหล่นหรือผิดพลาด ซึ่งตัวแปลภาษาตรวจพบและแสดงข่าวสารให้ผู้เขียนโปรแกรมทราบ

[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.

Run-Time-Errors

- เป็นความผิดพลาดที่เกิดขึ้นในขณะที่ปฏิบัติการ (run) โดยไม่สามารถปฏิบัติงานตามคำสั่งโปรแกรมได้ อาทิเช่น มีการหารค่าด้วย 0 เป็นต้น เมื่อมีความผิดพลาดลักษณะนี้เกิดขึ้น โปรแกรมจะหยุดการประมวลผลทันที พร้อมแสดงบรรทัดที่ผิดพลาดรวมทั้งข้อความที่อธิบายถึงความผิดพลาดที่เกิดขึ้นให้กับโปรแกรมเมอร์ได้ทราบ เพื่อให้โปรแกรมเมอร์แก้ไขต่อไป
- ความผิดพลาดที่เกิดขึ้นโดยปรากฏดังนี้คือ
“Floating point division by zero”

Logic Errors

- เป็นความผิดพลาดของผลลัพธ์ ซึ่งมีผลมาจากอัลกอริทึมในการแก้ปัญหาโปรแกรม ซึ่งยากที่จะค้นพบ ต้องใช้วิจรรย์ญาณ ความเชี่ยวชาญของโปรแกรมเมอร์ในการค้นหาและแก้ไข
- ความผิดพลาดนี้อาจเกิดจาก การเขียนนิพจน์ทางคณิตศาสตร์ไม่ถูกต้อง
- มีการใช้เงื่อนไขในการทำงานผิดพลาด
- หรือการลืมหำหนดค่าเริ่มต้น สำหรับปฏิบัติงานเป็นต้น

แบบฝึกหัด 1

จงเขียนโปรแกรมหาพื้นที่ของสามเหลี่ยมหน้าจั่ว

สูตร

$$\text{พื้นที่} = \frac{1}{2} * \text{ความยาวฐาน} * \text{ส่วนสูง}$$

แบบฝึกหัด 2

จงเขียนโปรแกรมหาค่าเฉลี่ยของเลขจำนวนเต็มใดๆ 5 จำนวน

แบบฝึกหัด 3

จงเขียนโปรแกรมในการแปลง จำนวนหลา จำนวนฟุต และจำนวนนิ้ว ใดๆ
ให้เป็นนิ้วทั้งหมด

เช่น ป้อน 2 หลา 2 ฟุต 4 นิ้ว

ผลคือ 100 นิ้ว

แบบฝึกหัด 4

จงแปลง นิ้ว ให้เป็น จำนวนหลา จำนวนฟุต และจำนวนนิ้ว

การบ้าน

1. จงแปลง ชั่วโมง นาที วินาที ให้เป็นวินาทีทั้งหมด
2. ป้อนจำนวนเงินที่เป็นเศษเหรียญทางแป้นพิมพ์ โปรแกรมสามารถนำไปหาจำนวนแบงค์ที่น้อยที่สุดสามารถแลกได้จากธนาคาร โดยแลกจากแบงค์ใหญ่จนกระทั่งถึงแบงค์ที่น้อยที่สุด
3. จงหาค่ายกกำลัง 2 , ยกกำลัง 3 และค่ายกกำลัง 4 ของเลขใดๆ