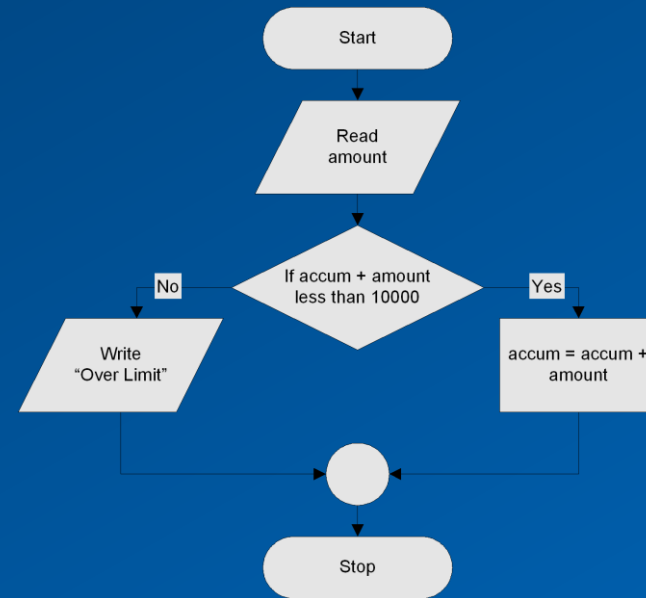


Day/Time	8:00-9:00	9:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00
จันทร์	1101031 (3) 3, B1209 B		1101032 (3) 3, B1209 B			1101031 (3) 3, Lab Com11:cadIII F6		1101032 (3) 3, Lab Com11:cadIII F6	
อังคาร	213102 (3) 3, ONLINE ONLINE	1101033 (2) 1, B1209 B				1101033 (2) 1, B1209 B			
พุธ		1101023 (2) 1, B2102 B				1101023 (2) 1, B2102 B			
พฤหัสบดี			213102 (3) 3, B2102 B						
ศุกร์			1101021 (3) 3, B2102 B		1101022 (3) 3, B2102 B				
	1101021 (3) 3, B2102 B		1101022 (3) 3, B2102 B			1101021 (3) 3, Lab_Com7 B2		1101022 (3) 3, Lab_Com7 B2	



1101021 Software Dev. 1

01 FLOWCHART & BASIC C++ PROGRAMMING

Details for 1101021

■ Details for 1101021

- Lecturer 1: Assistant Professor Dr. Jitimon Angskun
 - DIGITECH
 - Email: jitimon@sut.ac.th, Tel.: 0-4422-4309
- Lecturer 2: Dr. Phichayasini Kitwatthanathawon
 - DIGITECH
 - Email: pichak@sut.ac.th, Tel.: 0-4422-4352
- TA: Sasit Vongrukun (P' CD)
 - DIGITECH
 - Email: sasit53@sut.ac.th, Tel.: 0-4422-3000 Ext. 6203

Details for 1101021

Details for 1101021 (cont.)

Laboratory

```
C:\Cherpongom\TEST.CPP - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
[Icons] [globals] [TDM-GCC 4.9.2 64-bit Release]
Project Classes Debug [*] TEST.CPP
1  /* LANG: C */
2
3  #include <iostream>
4
5  using namespace std;
6
7  int main()
8  {
9      int a;
10
11      cin >> a;
12      cout << "Hello world!" << endl;
13
14      return 0;
15 }
```

Submit



http://it2.sut.ac.th/grader/posn_web/login.php

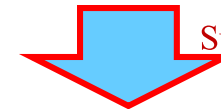
Welcome: Administrator

[\[upload student info\]](#) [\[list user passwords\]](#) [\[random user passwords\]](#) [\[list printers\]](#)

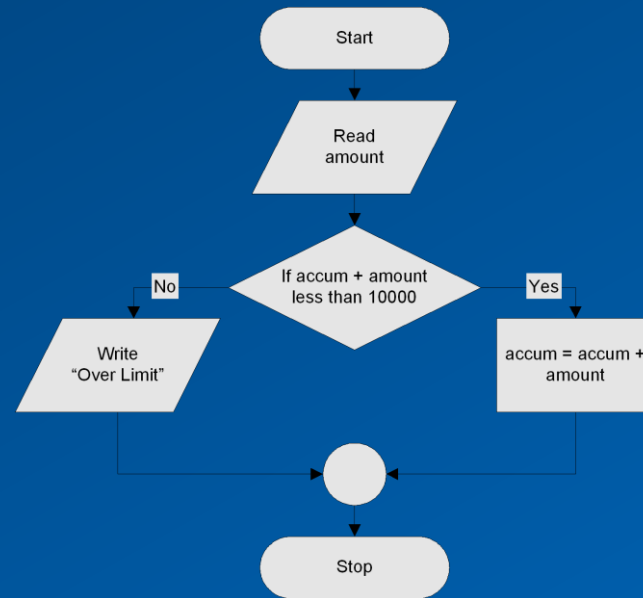
- add [add]
1 submission(s), last on 2017-11-24 09:59:43 of size 127 bytes
ผลการตรวจ: **accepted** [PPPPP]
[\[compiler message\]](#) [\[source\]](#)
- Pr2 [Pr2]
ยังไม่ได้ส่ง
- Pr3 [Pr3]
ยังไม่ได้ส่ง

Task: add ▾ Source: Choose File No file chosen ส่งไปตรวจ

Summary



Back	
User Name	Score
toi	100
โอจักรพรรดิ สุขชัยสิทธิ์ (เจแปน)	100
Administrator	100



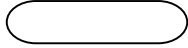
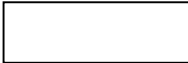



Flowchart

01 FLOWCHART & BASIC C++ PROGRAMMING

Flowchart & Intro. to C++

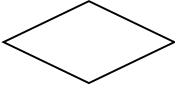
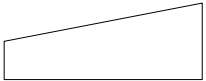
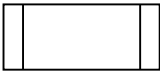
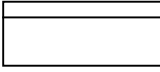
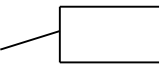



■ Flowchart

- Graphical representation
- Tool for programmers to develop program logic
- Interchange logic between programmers

Symbol	Meaning
	Terminator Symbol ใช้แสดงตำแหน่งของจุดเริ่มต้น จุดสิ้นสุดโปรแกรม
	Process Symbol ใช้ในการคำนวณ
	Input/Output Symbol ใช้ในการรับข้อมูล การแสดงผลข้อมูล
	Display /Monitor Symbol ใช้เมื่อต้องการระบุให้แสดงข้อมูลบนจอภาพ
	Document Symbol ใช้เมื่อต้องการระบุให้แสดงข้อมูลบนเครื่องพิมพ์

Flowchart & Intro. to C++

■ Flowchart (cont.)

Symbol	Meaning
	Decision Symbol ใช้ในการเปรียบเทียบเงื่อนไขหรือตัดสินใจ
	Keyboard Symbol ใช้ในการรับข้อมูลจากแป้นพิมพ์
	External Subroutine Symbol ใช้เรียกฟังก์ชันต่าง ๆ ที่ C Compiler สร้างไว้แล้ว
	Internal Subroutine Symbol ใช้เรียกฟังก์ชันต่าง ๆ ที่ผู้เขียนโปรแกรมสร้างขึ้นเอง
	Comment Symbol ใช้แสดงความเห็น คำอธิบาย หรือข้อสังเกตต่าง ๆ
	Flowline Symbol ใช้เป็นจุดเชื่อมต่อระหว่างสัญลักษณ์
	On-page Connector Symbol ใช้ในการเชื่อมโยงในหน้าเดียวกัน
	Off-page Connector Symbol ใช้ในการเชื่อมโยงไปหน้าอื่น

Flowchart & Intro. to C++

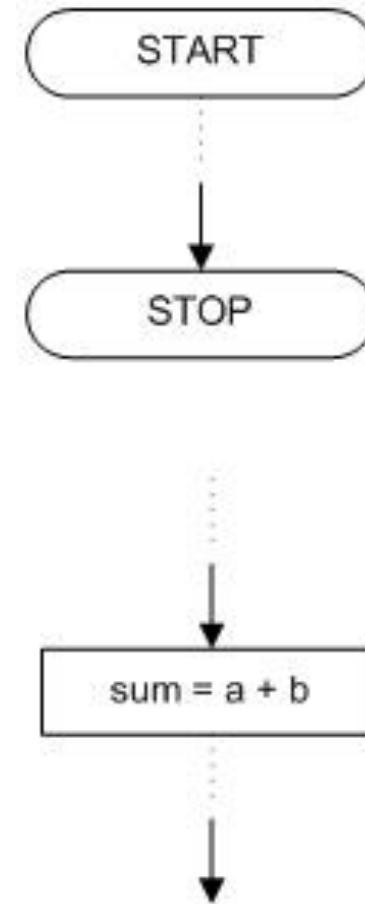
■ Flowchart (cont.)

■ Examples:

■ Terminator Symbol



■ Process Symbol



Flowchart & Intro. to C++

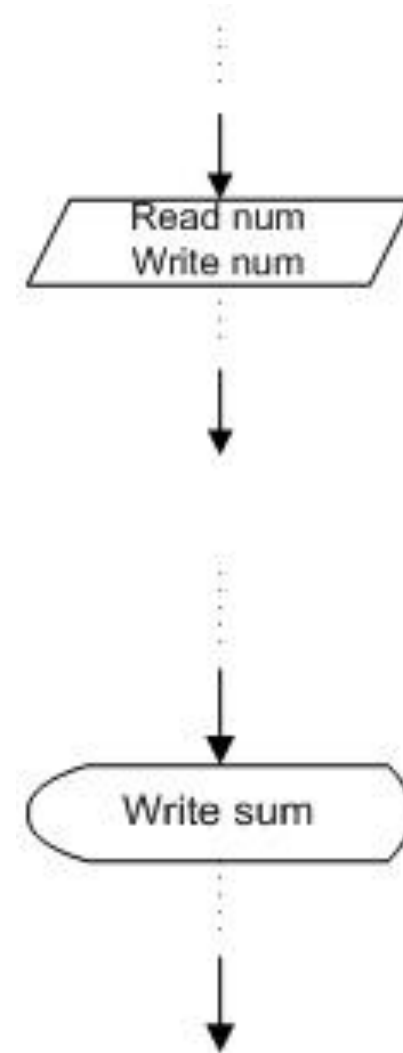
■ Flowchart (cont.)

■ Examples:

■ Input/Output Symbol



■ Display/Monitor Symbol

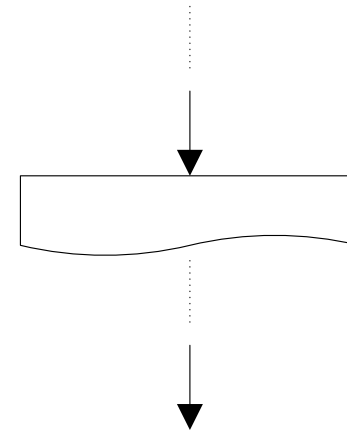


Flowchart & Intro. to C++

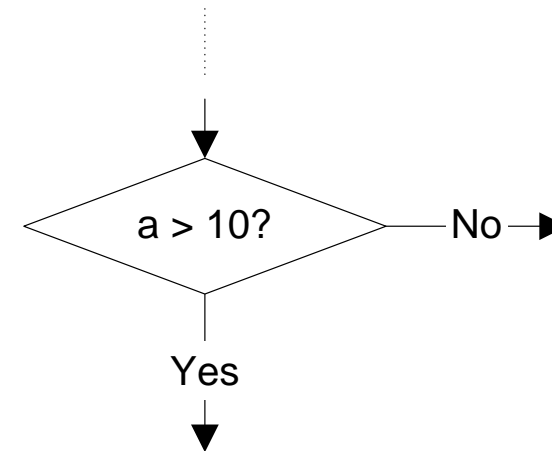
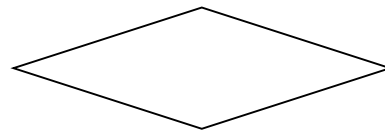
■ Flowchart (cont.)

■ Examples:

■ Document Symbol



■ Decision Symbol

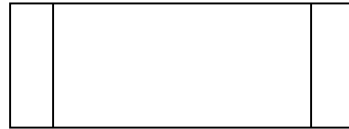


Flowchart & Intro. to C++

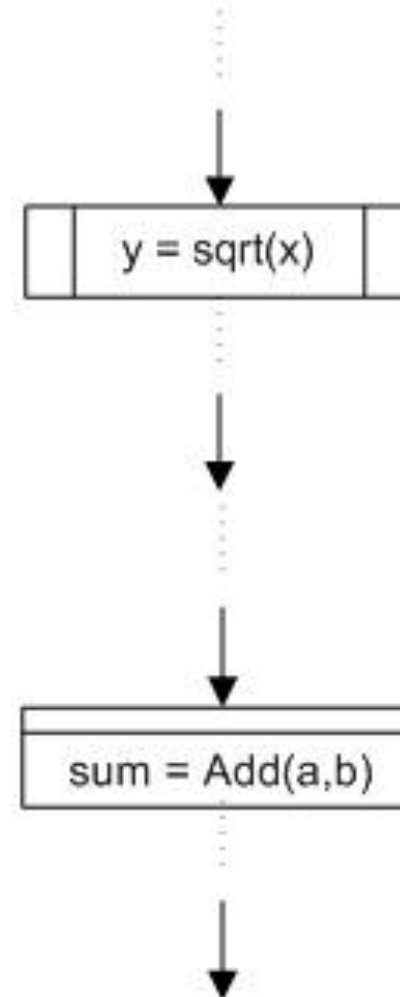
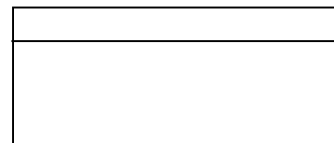
■ Flowchart (cont.)

■ Examples:

■ External Subroutine Symbol



■ Internal Subroutine Symbol

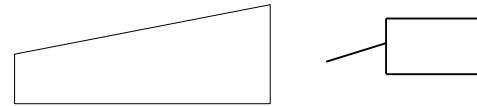


Flowchart & Intro. to C++

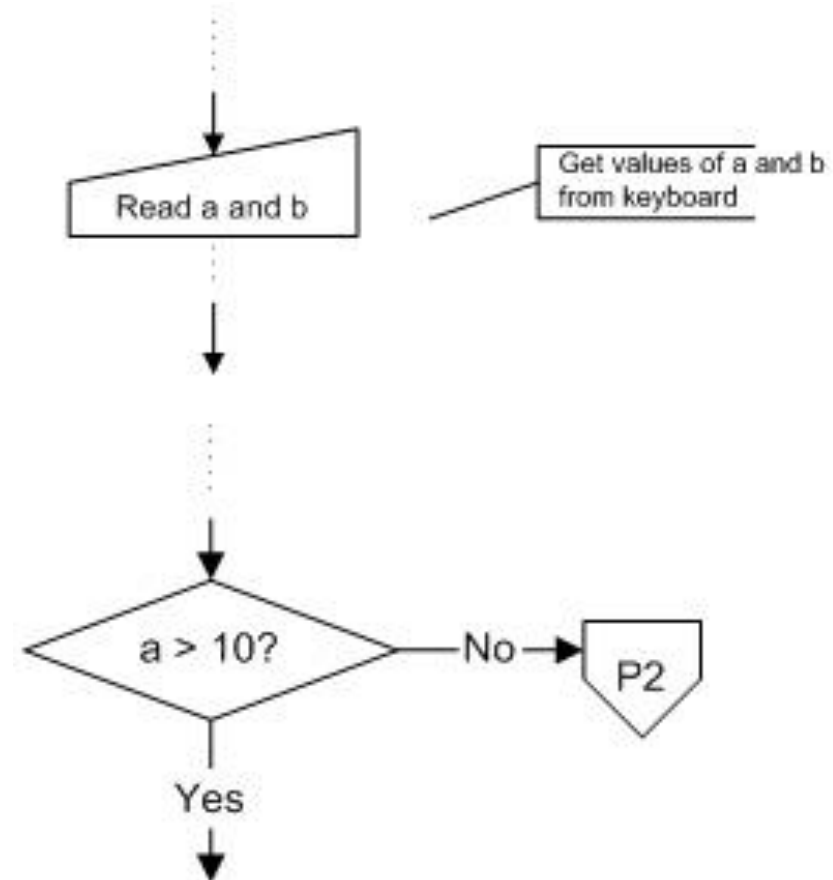
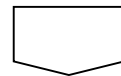
Flowchart (cont.)

Examples:

Keyboard and Comment Symbol



Offpage Connector Symbol



Flowchart & Intro. to C++

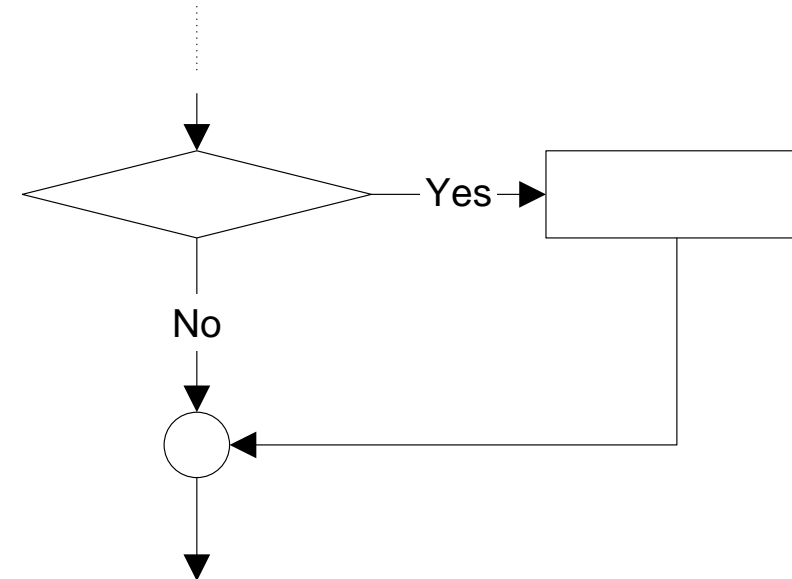
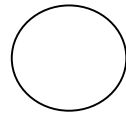
■ Flowchart (cont.)

■ Examples:

■ Flowline Symbol



■ Connector Symbol



Flowchart & Intro. to C++

■ Flowchart (cont.)

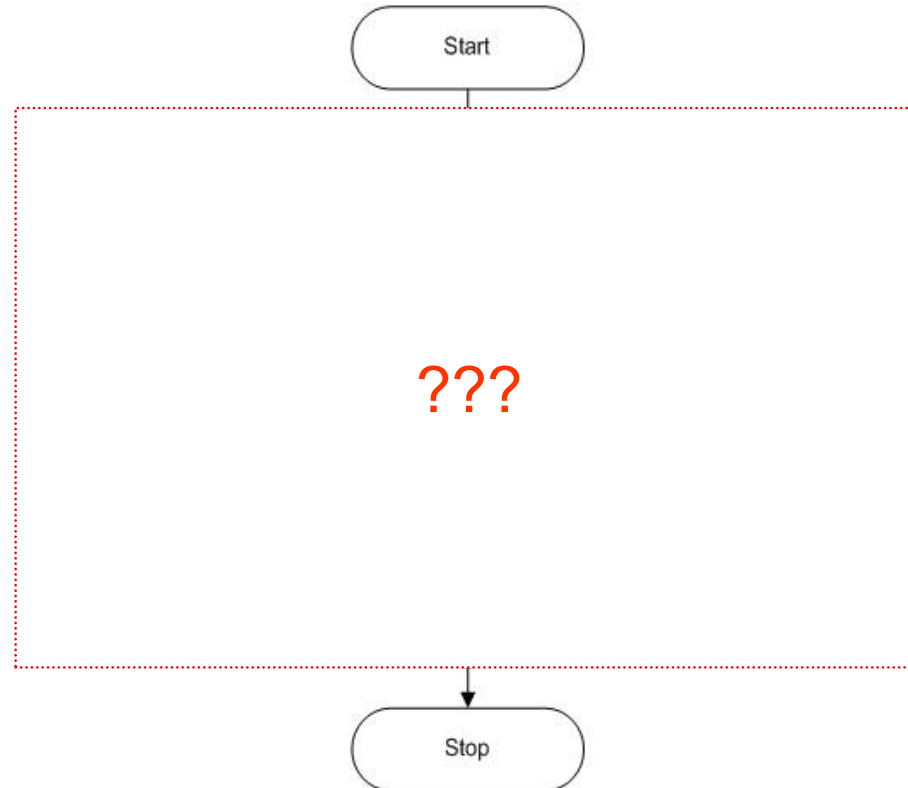
- ตัวอย่างที่ 1 การใช้สัญลักษณ์ flowchart แทนการเขียนโปรแกรมในการรับค่า 3 ค่าจากคีย์บอร์ด แล้วพิมพ์ผลบวกของเลขทั้ง 3 ออกที่จอภาพ



Flowchart & Intro. to C++

Flowchart (cont.)

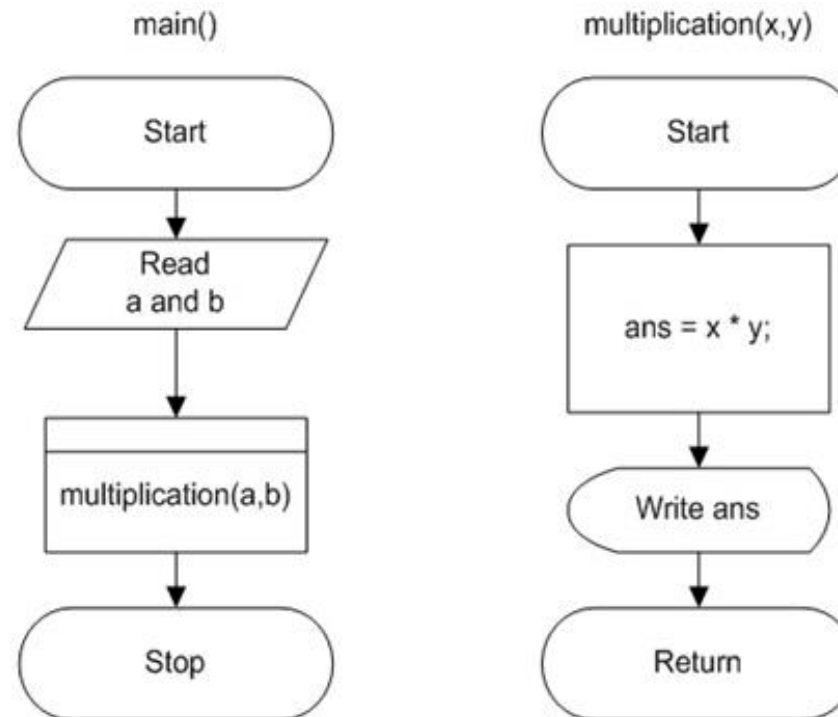
- ตัวอย่างที่ 2 การใช้สัญลักษณ์ flowchart แทนการเขียนโปรแกรมควบคุมยอดลูกหนี้ ซึ่งเมื่อลูกค้าต้องการซื้อสินค้าด้วยยอดเงินจำนวนหนึ่ง (amount) โปรแกรมตรวจสอบยอดหนี้คงค้าง (accum) ถ้ารวมกับหนี้ที่จะเพิ่มใหม่แล้วมีค่าน้อยกว่าวงเงินที่กำหนด (ในที่นี้มีค่าเท่ากับ 10,000 บาท) จึงยอมให้เพิ่มหนี้ได้

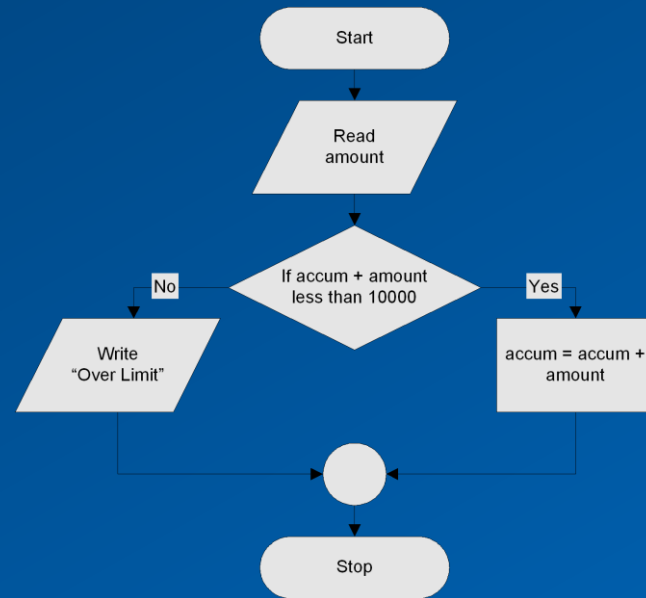


Flowchart & Intro. to C++

Flowchart (cont.)

- ตัวอย่างที่ 3 การใช้สัญลักษณ์ flowchart แทนการเขียนโปรแกรมเพื่อทำการหาผลคูณของเลข 2 จำนวน โดยโปรแกรมจะรับข้อมูลของเลขจำนวนเต็มทั้ง 2 ในฟังก์ชัน `main()` แล้วทำการส่งข้อมูลนั้นไปที่ฟังก์ชัน `multiplication()` เพื่อทำการคำนวณหาผลลัพธ์และแสดงออกที่จอภาพ





History of C++

01

FLOWCHART & BASIC C++ PROGRAMMING

Flowchart & Intro. to C++

History of C and C++

- **Originally develop in 1978**

- by Dennis Ritchie and Brian Kernighan at Bell Labs
- Outgrowth of two earlier languages
- BCPL and B

- **Design to write an operating system**

- UNIX

- **Popular since 1980's**

- most institutions use C as an introductory programming language

- **C++ develop in 1983**

- by Bjarne Stroustrup at Bell Labs
- OOP (Object Oriented Programming)



Flowchart & Intro. to C++

■ Some Difficulties in Learning How to Program

■ Lazy!

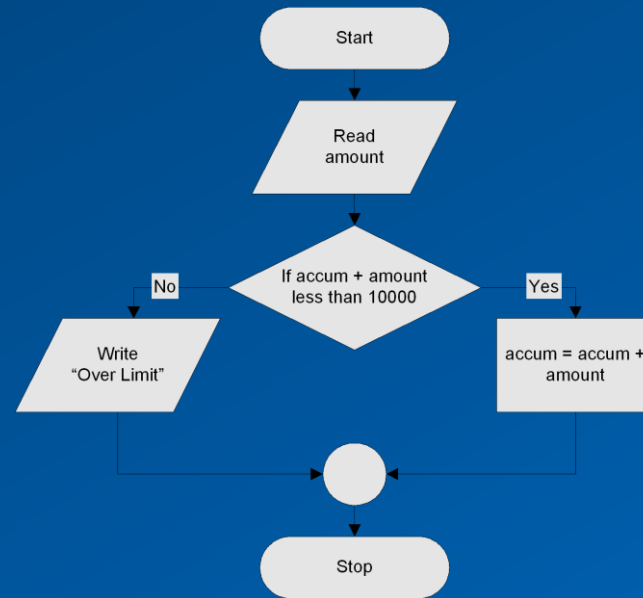
- spend insufficient time in programming

■ Difficult and complicated

- too many things happen at once
- more difficult than other courses
- invisible process
- involving syntactic knowledge (specific details and rules), semantic knowledge (eg., `total = total + 1;`), and strategic knowledge (techniques to plan and combine syntactic and semantic knowledge)

■ Abstract concepts

- difficult to visualize programming concepts

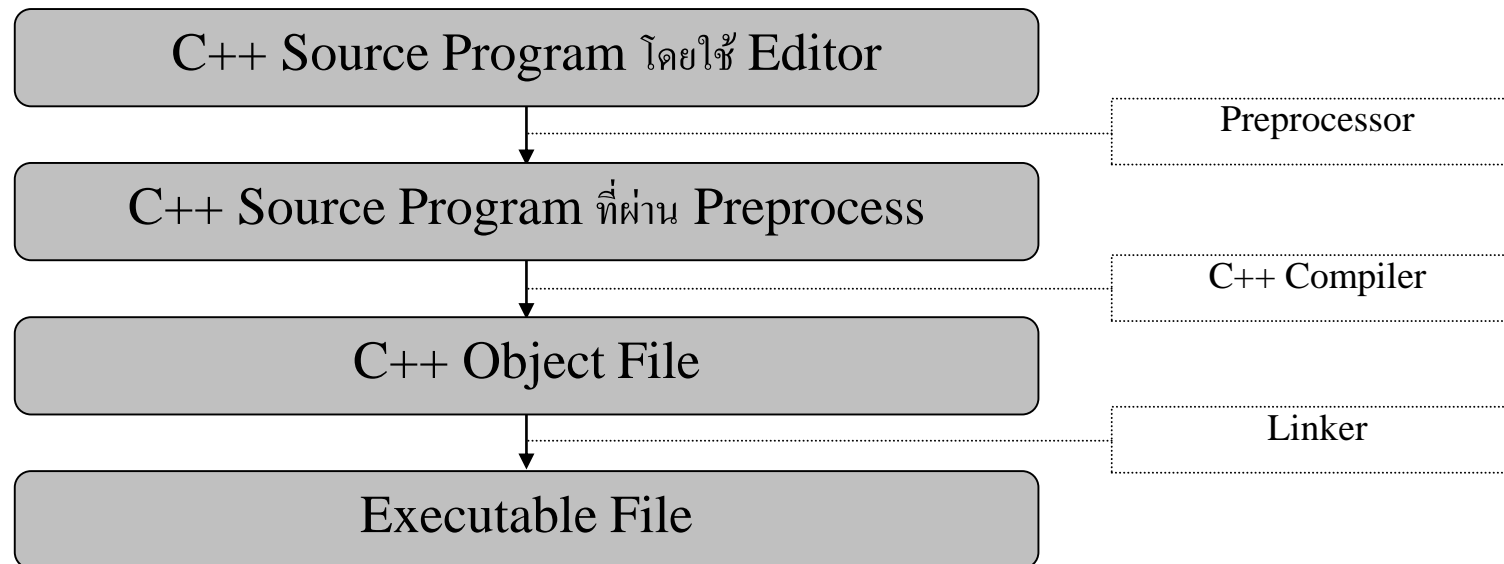


Structure of C++

01 FLOWCHART & BASIC C++ PROGRAMMING

Flowchart & Intro. to C++

■ Structure of C++



Flowchart & Intro. to C++

■ Structure of C++

■ Preprocessor commands

- `#include<stdio.h>` หรือ `#include "stdio.h"`
- `getch()` → `#include<conio.h>`

■ Define data type

- `int`, `float`, `char`, etc.

■ Declare function prototypes

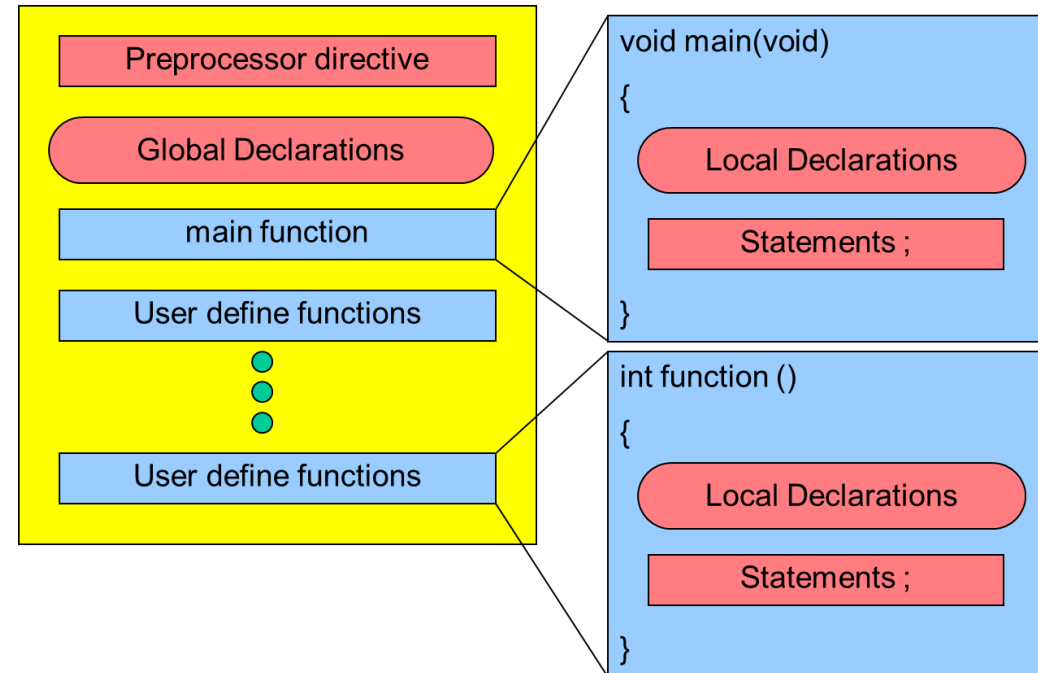
- `int add_two_number(int, int)`

■ Declare variables

- local or global variables

■ Define functions

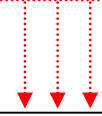
- `void main()`
- User define function



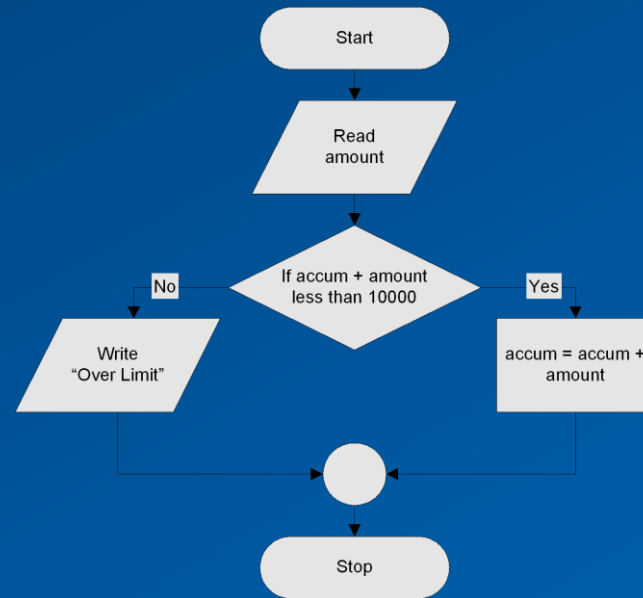
Flowchart & Intro. to C++

■ Example of a simple C++ program

```
1. /* LANG: C++ */
2. /* This is my first program */           /* comment */
3. #include <iostream>                       /* preprocessor directive */
4. using namespace std;
5. void main(){                             /* function main - start program */
6.     int a, b, sum;                       /* variables declaration */
7.     cout << "Enter two numbers: ";       /* output */
8.     cin >> a;    // input
9.     cin >> b;    // input
10.    sum = a + b;                          /* computation */
11.    cout << "The sum is: " << sum << endl; /* output */
12. }                                       /* end program */
```



Enter two numbers: 5 7
The sum is: 12



Keywords in C++

01

FLOWCHART & BASIC C++ PROGRAMMING

Basic C++ Programming



Identifier

- Keyword
 - void, if, else, int, char, float, case, auto, return
- Standard Identifier
 - abort, abs, arc, ftime, getch, open, rename
- User-defined identifier

Basic C++ Programming

Keywords in C++

<code>alignas</code> (since C++11) <code>alignof</code> (since C++11) <code>and</code> <code>and_eq</code> <code>asm</code> <code>atomic_cancel</code> (TMTS) <code>atomic_commit</code> (TMTS) <code>atomic_noexcept</code> (TMTS) <code>auto</code> (1) <code>bitand</code> <code>bitor</code> <code>bool</code> <code>break</code> <code>case</code> <code>catch</code> <code>char</code> <code>char16_t</code> (since C++11) <code>char32_t</code> (since C++11) <code>class</code> (1) <code>compl</code> <code>concept</code> (since C++20) <code>const</code> <code>constexpr</code> (since C++11) <code>const_cast</code> <code>continue</code> <code>co_await</code> (coroutines TS) <code>co_return</code> (coroutines TS) <code>co_yield</code> (coroutines TS) <code>decltype</code> (since C++11) <code>default</code> (1) <code>delete</code> (1)	<code>do</code> <code>double</code> <code>dynamic_cast</code> <code>else</code> <code>enum</code> <code>explicit</code> <code>export</code> (1) <code>extern</code> (1) <code>false</code> <code>float</code> <code>for</code> <code>friend</code> <code>goto</code> <code>if</code> <code>import</code> (modules TS) <code>inline</code> (1) <code>int</code> <code>long</code> <code>module</code> (modules TS) <code>mutable</code> (1) <code>namespace</code> <code>new</code> <code>noexcept</code> (since C++11) <code>not</code> <code>not_eq</code> <code>nullptr</code> (since C++11) <code>operator</code> <code>or</code> <code>or_eq</code> <code>private</code> <code>protected</code> <code>public</code>	<code>register</code> (2) <code>reinterpret_cast</code> <code>requires</code> (since C++20) <code>return</code> <code>short</code> <code>signed</code> <code>sizeof</code> (1) <code>static</code> <code>static_assert</code> (since C++11) <code>static_cast</code> <code>struct</code> (1) <code>switch</code> <code>synchronized</code> (TMTS) <code>template</code> <code>this</code> <code>thread_local</code> (since C++11) <code>throw</code> <code>true</code> <code>try</code> <code>typedef</code> <code>typeid</code> <code>typename</code> <code>union</code> <code>unsigned</code> <code>using</code> (1) <code>virtual</code> <code>void</code> <code>volatile</code> <code>wchar_t</code> <code>while</code> <code>xor</code> <code>xor_eq</code>
---	--	---

<code>override</code> (C++11) <code>final</code> (C++11) <code>transaction_safe</code> (TMTS) <code>transaction_safe_dynamic</code> (TMTS)

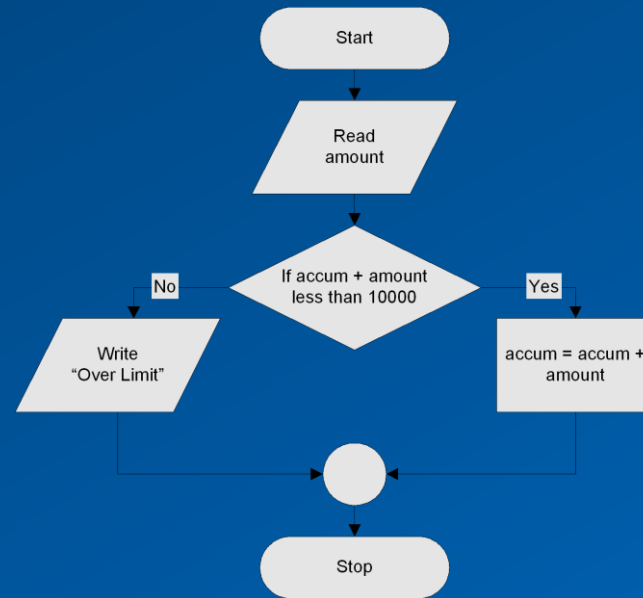
<code>if</code> <code>elif</code> <code>else</code> <code>endif</code> <code>defined</code>	<code>ifdef</code> <code>ifndef</code> <code>define</code> <code>undef</code>	<code>include</code> <code>line</code> <code>error</code> <code>pragma</code>
---	--	--

Basic C++ Programming



Identifier

- Keyword
 - void, if, else, int, char, float, case, auto, return
- Standard Identifier
 - abort, abs, arc, ftime, getch, open, rename
- User-defined identifier



Data Type

01

FLOWCHART & BASIC C++ PROGRAMMING

Basic C++ Programming



What and Why Data Type?

- What kind of data are we dealing with?

- character, number etc.

- What is a data type?

- a keyword used to determine the values that the variable can contain and the operations that can be performed on it

Basic C++ Programming

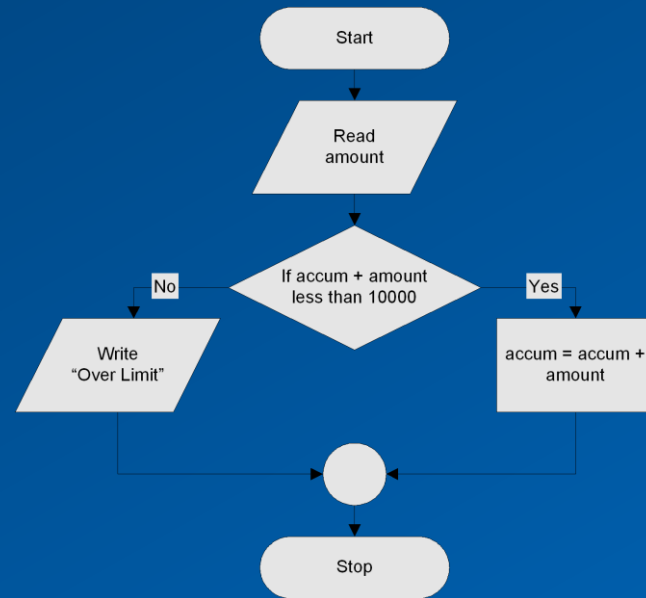
■ Simple/Basic Data Types

- int
- double
- string

Basic C++ Programming

■ Data Types Details

ชนิดข้อมูล	ค่าต่ำสุด	ค่าสูงสุด	ใช้พื้นที่หน่วยความจำ
char	-128	127	1 byte
unsigned char	0	255	1 byte
int	-32,768	32,767	2 byte
unsigned int	0	65,535	2 byte
short int	-32,768	32,767	2 byte
long	-2,147,483,648	2,147,483,647	4 byte
unsigned long	0	4,294,967,295	4 byte
float	3.4×10^{-38}	$3.4 \times 10^{+38}$	4 byte
double	1.7×10^{-308}	$1.7 \times 10^{+308}$	8 byte
long double	3.4×10^{-4932}	$3.4 \times 10^{+4932}$	10 byte



Variable

01 FLOWCHART & BASIC C++ PROGRAMMING

Basic C++ Programming

■ What variable?

- an identifier used to represent some specific information
- must be declared before it can be used
 - `data_type variable_name;`
 - `data_type variable_name1, variable_name2, variable_name3, ... ;`

Example1

```
float sales, purchase;  
char grade;
```

Example2

```
float total=sale*price;
```

Basic C++ Programming

■ Variable Names

- first character must be a letter (a-z or A-Z), underscore (_) or @
- other characters may be letter, underscore or digit (0-9) but not space in between
- case-sensitive (small letter is not the same as capital letter)
- normally, first underscore character is used for system variables
- name must not be the same as C reserved keywords

Basic C++ Programming

■ Declarations

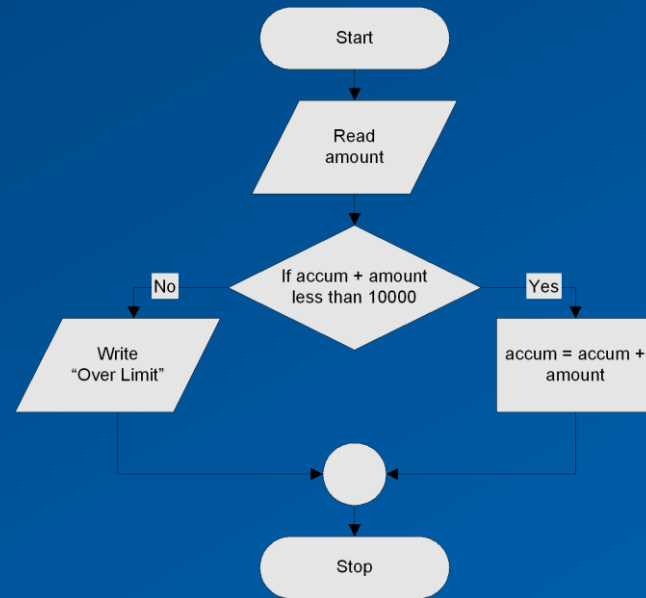
- associates a group of variables with a specific data type

```
. . .
int    a, b, sum = 0;
double gpa;
string grade, text = "Suranaree";
. . .
. . .
a = 3;
b = 5;
sum = a + b;
gpa = 3.25;
grade = "A";
. . .
```

```
void main() {
    {
        int i=0;
    }
    cout << i;           // invalid
}
```

Note: all variables must be declared before they can be used.

and they can be used only in the scope ({}) of the declaration.



Constant

01 FLOWCHART & BASIC C++ PROGRAMMING

Basic C++ Programming

■ What constant?

- a value assigned to a variable
 - `const data_type constant_name = constant_value;`

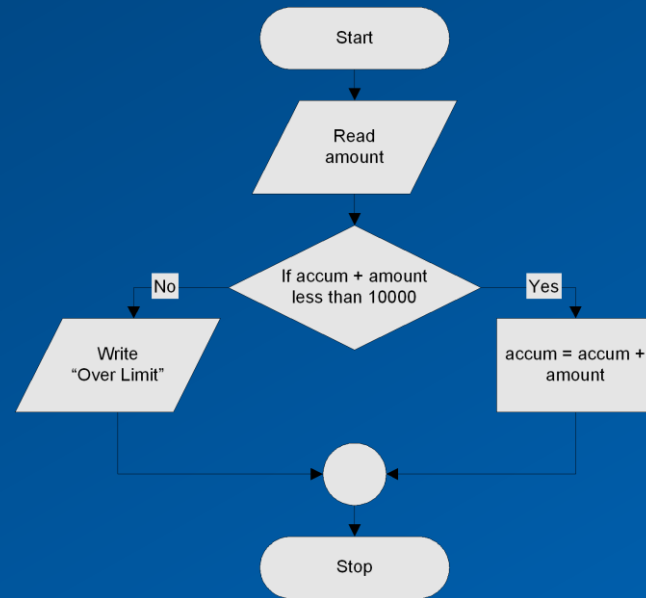
Example

```
const int Day = 7;
```

```
const int month = 12;
```

```
const float PI = 3.1418928;
```

```
const char name = 'A'; //ASCII of A is 65
```



What is Input?

01

FLOWCHART & BASIC C++ PROGRAMMING

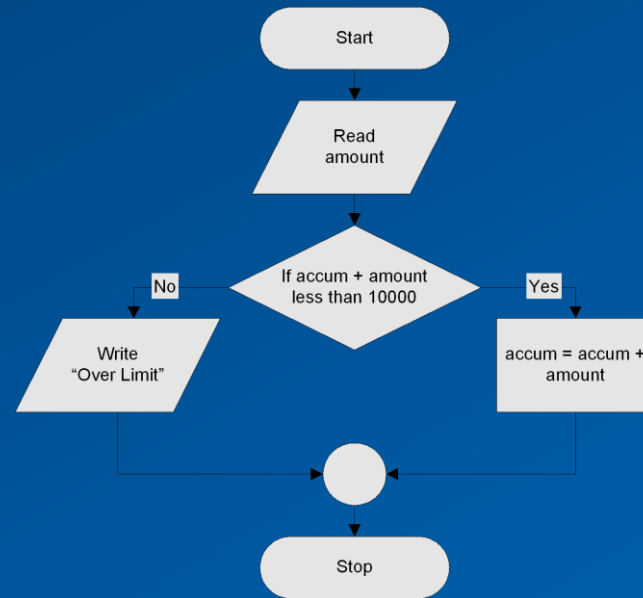
Basic C++ Programming

■ What is Input?

- data obtained from keyboard, disk, etc.

■ cin

- `cin >> ชื่อตัวแปร;`
- `cin >> ชื่อตัวแปร >> ชื่อตัวแปร >> ชื่อตัวแปร >> ...;`
- `cin >> number1 >> number2 >> number3;`
 - 50 30 40 <Enter>
or
 - 50 <Enter>
30 <Enter>
40 <Enter>



What is Output?

01

FLOWCHART & BASIC C++ PROGRAMMING

Basic C++ Programming

■ What is Output?

- Data sent from computer

■ cout

- `cout << "ข้อความ" << "ข้อความ" << ตัวแปร;`
- `cout << ตัวแปร << "ข้อความ" << ตัวแปร << ...;`

■ endl

- new line

■ setw(n)

- `#include <iomanip.h>`
- `cout << setw(10) << "World";` หรือ `cout << setw(10) << left << "World";`

■ setprecision(n)

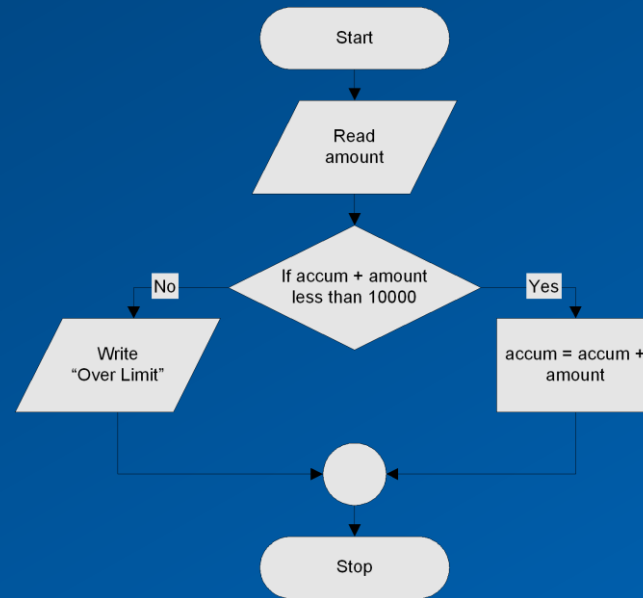
- `#include <iomanip.h>`
- `cout << "A+B = " << fixed << setprecision(2) << A+B << endl;`

Basic C++ Programming

■ What is Output? (cont.)

■ Example

```
1. /* LANG: C++ */
2. #include <iostream>
3. using namespace std;
4. int main()
5. {
6.     int a, b, sum;                // variable declaration
7.     cout << "Enter the first number: "; // output
8.     cin >> a;                      // input
9.     cout << "Enter the second number: "; // output
10.    cin >> b;                      // input
11.    sum = a + b;                   // computation
12.    cout << "The sum is " << sum << "."; // output
13. return 0;
14. }
```



Example of a C++ program

01

FLOWCHART & BASIC C++ PROGRAMMING

Basic C++ Programming

■ Example

```
1. /* LANG: C++ */
2. #include <iostream>
3. using namespace std;
4. int main()
5. {
6.     cout << "Display change base number..." << endl << endl;
7.     cout << "10 decimal to hexadecimal = " << hex << 10 << endl;
8.     cout << "5865 decimal to hexadecimal = " << hex << 5865 << endl;
9.     cout << "1250 decimal to octal = " << oct << 1250 << endl;
10.    cout << "02342 octal to decimal = " << dec << 02342 << endl;
11.    cout << "0xabc125 hexadecimal to decimal = " << dec << 0xabc125 << endl;
12.    cout << "0xf hexadecimal to decimal = " << dec << 0xf << endl;
13.    return 0;
14. }
```