



Chapter 2

Introduction to C++ Programming

李慧颖

huiyingli@seu.edu.cn



OBJECTIVES



- ☐ To write simple computer programs in C++.
- ☐ To write simple **input** and **output** statements.
- ☐ To use **fundamental types**.
- ☐ Basic computer memory concepts.
- ☐ To use **arithmetic operators** (算术操作符)
- ☐ The **precedence** (优先级) of arithmetic operators.
- ☐ To write simple **decision-making** statements



Topics



- ☐ **2.1 Introduction**
- ☐ **2.2 First Program in C++: Printing a Line of Text**
- ☐ **2.3 Standard library(标准库)**
- ☐ **2.4 Modifying Our First C++ Program**
- ☐ **2.5 Streams(流)**
- ☐ **2.6 Stream Input(输入流)/ Stream Output(输出流)**
- ☐ **2.7 Another C++ Program: Adding Integers**
- ☐ **2.8 Memory Concepts**
- ☐ **2.9 Arithmetic**
- ☐ **2.10 Decision Making: Equality and Relational Operators**



2.1 Introduction



- 五个C++程序实例（数据输入/输出/关系判断）
- 1-3个例子
 - ❖ 如何在屏幕上显示信息？
- 第4个例子
 - ❖ 如何进行数学运算？
- 第5个例子
 - ❖ 如何使用判断语句？



Topics



- ☐ 2.1 Introduction
- ☐ **2.2 First Program in C++: Printing a Line of Text**
- ☐ 2.3 Standard library(标准库)
- ☐ 2.4 Modifying Our First C++ Program
- ☐ 2.5 Streams(流)
- ☐ 2.6 Stream Input(输入流)/ Stream Output(输出流)
- ☐ 2.7 Another C++ Program: Adding Integers
- ☐ 2.8 Memory Concepts
- ☐ 2.9 Arithmetic
- ☐ **2.10 Decision Making: Equality and Relational Operators**



2.2 First Program in C++: Printing a Line of Text



□ Simple program

❖ Prints a line of text

❖ Illustrates several important features of C++

```
1 // Fig. 2.1: fig02_01.cpp
2 // Text-printing program.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "Welcome to C++!\n"; // display message
9
10    return 0; // indicate that program ended successfully
11
12 } // end function main
```

Welcome to C++!



2.2 First Program in C++: Printing a Line of Text



```
1 // Fig. 2.1: fig02_01.cpp
2 // Text-printing program.
3 #include <iostream> // allows program to output data to the screen
4
5 // f
6 int
7 {
8     s
9
10    r
11
12 } //
```

Line1-2: Comments (注释)

- // single-line comment (单行注释)
- /* */ (多行注释)
- Used to describe program, 目的、作者、日期和时间等信息
- 注释文字不影响编译和运行



2.2 First Program in C++: Printing a Line of Text



去掉，what happens?

```
1 // Fig. 2.1: fig02_01.cpp
2 // Text-printing program.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "Hello, world!" << endl;
9     return 0;
10 }
11
12 // end
```

- #: 预处理命令(preprocessor directive), 编译之前处理
- include 文件包含
 - <文件名> (标准库) 或者 “文件名”
 - <iostream> (Ch2.5)
 - 功能: 数据输入/输出



2.2 First Program in C++: Printing a Line of Text



```
1 // Fig. 2.1: fig02_01.cpp
2 // Text-printing program.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
```

Line 4: 空行 (blank lines)

- 被编译器忽略
- 作用: 增加程序的可读性
 - TAB字符、空格字符起同样作用



2.2 First Program in C++: Printing a Line of Text



```
1 // Fig. 2.1: fig02_01.cpp
2 // Text-printing program.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "Welcome to C++!\n"; // display message
9
10
11
12 }
```

Line 5: // function main begins program execution

● 单行注释



2.2 First Program in C++: Printing a Line of Text



```
1 // Fig. 2.1: fig02_01.cpp
2 // Text-printing program.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "Welcome to C++!\n"; // display message
9
10    return 0; // indicate that program ended successfully
```

Line 6: `int`

Line 7: {

- `()`表示main函数
- 关键字`int`

关键字(Keyword): 关键字是预留的标识符, 每个关键字都有特殊的含义. 我们不能在C++程序中使用与关键字同名的标识符, 例如: `int`, `main`等等(Fig. 4.3)

- C++程序可包含多个函数, 但只能有一个`main`函数;
- 一般, `main`函数是程序执行的入口, 可执行程序必须有;
- 左花括号{ 应放在每个函数体(Body)开头, 对应的右花括号} 应在函数体结尾出现



2.2 First Program in C++: Printing a Line of Text



```
1 // Fig. 2.1: fig02_01.cpp
2 // Text-printing program.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "Welcome to C++!\n"; // display message
9
10    return 0; // indicate that program ended successfully
```

Line 8: **std::cout << "Welcome to C++!\n";** // display message

- C++程序的语句 (**Statement**);
- **std::cout**: std名空间, 输出流对象 (Output stream object);
- **<<**: 流插入操作符 (stream insertion operator);
- **"Welcome to C++!\n"**: 输出到**std::cout** (标准输出流) 的**字符串**;
- ****: Escape character (转义字符), 不打印在屏幕上, 意味着一个特殊的字符将被打印;
- **\n**: Escape sequence (转义序列) (见下页);
- **;**: semicolon (分号), C++语句结束.



2.2 First Program in C++: Printing a Line of Text



Escape sequence (转义序列)	Description
<code>\n</code>	Newline(换行符) Position the screen cursor to the beginning of the next line.
<code>\t</code>	Horizontal tab(水平制表符) Move the screen cursor to the next tab stop.
<code>\r</code>	Carriage return(回车符) Position the screen cursor to the beginning of the current line; do not advance to the next line.
<code>\a</code>	Alert(警告). Sound the system bell.
<code>\\</code>	Backslash(反斜杠) Used to print a backslash character.
<code>\'</code>	Single quote(单引号) Use to print a single quote character.
<code>\"</code>	Double quote(双引号) Used to print a double quote character.



2.2 First Program in C++: Printing a Line of Text



```
1 // Fig. 2.1: fig02_01.cpp
2 // Text-printing program.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "Welcome to C++!\n"; // display message
9
10    return 0; // indicate that program ended successfully
11
12 }
```

Line 10: `return 0; // indicate that program ended successfully`

Line 12: `}`

- 关键字 `return` 是退出函数的几种方式之一
- 数值 `0` 表示程序执行顺利结束
- 目前只要记住在每个程序的 `main` 函数中都要包括这个语句, 后面将介绍包括这个语句的原因.



Topics



- ☐ 2.1 Introduction
- ☐ 2.2 First Program in C++: Printing a Line of Text
- ☐ **2.3 Standard library(标准库)**
- ☐ 2.4 Modifying Our First C++ Program
- ☐ 2.5 Streams(流)
- ☐ 2.6 Stream Input(输入流)/ Stream Output(输出流)
- ☐ 2.7 Another C++ Program: Adding Integers
- ☐ 2.8 Memory Concepts
- ☐ 2.9 Arithmetic
- ☐ **2.10 Decision Making: Equality and Relational Operators**



2.3 Standard library(标准库)



□ 由编译器厂商提供, 与系统平台、厂商和编译器版本无关

□ 1. 标准函数库

❖ 从C语言中继承下来

❖ C格式的输入输出函数、字符与字符串处理函数、数学函数、时间日期函数、动态分配函数以及一些实用函数

□ 2. 标准类库

❖ 标准C++的I/O流类、字符串类、数字类、异常处理和杂项类以及STL (Standard Template Library, 标准模板库)容器类



2.3 Standard library(标准库)



- 包含(include)相应的头文件

- C++的头文件来源:

- ❖ 标准C语言库函数的头文件, 带有.h后缀;

- #include <string.h>

- ❖ 标准C++语言类库的头文件, 不带.h后缀;

- #include <iostream>

- ❖ 由标准C语言库函数头文件包装成的标准C++的头文件, 把原有标准C语言库函数头文件去掉.h后缀而加上c前缀。

- #include <cstring>



Topics



- ☐ 2.1 Introduction
- ☐ 2.2 First Program in C++: Printing a Line of Text
- ☐ 2.3 Standard library(标准库)
- ☐ **2.4 Modifying Our First C++ Program**
- ☐ 2.5 Streams(流)
- ☐ 2.6 Stream Input(输入流)/ Stream Output(输出流)
- ☐ 2.7 Another C++ Program: Adding Integers
- ☐ 2.8 Memory Concepts
- ☐ 2.9 Arithmetic
- ☐ **2.10 Decision Making: Equality and Relational Operators**



2.4 Modifying Our First C++ Program



```
1 // Fig. 2.3: fig02_03.cpp
2 // Printing a line of text with multiple statements.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "welcome ";
9     std::cout << "to C++!\n";
10
11     return 0; // indicate that program ended successfully
12
13 } // end function main
```

多行语句打印一行文本

welcome to C++!



2.4 Modifying Our First C++ Program



```
1 // Fig. 2.4: fig02_04.cpp
2 // Printing multiple lines of text with a single statement.
3 #include <iostream> // allows program to output data to the screen
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "welcome\nto\n\nC++!\n";
9
10    return 0; // indicate that program ended successfully
11
12 }
```

一行语句打印多行文本

```
welcome
to

C++!
```



Topics



- ☐ 2.1 Introduction
- ☐ 2.2 First Program in C++: Printing a Line of Text
- ☐ 2.3 Standard library(标准库)
- ☐ 2.4 Modifying Our First C++ Program
- ☐ **2.5 Streams(流)**
- ☐ 2.6 Stream Input(输入流)/ Stream Output(输出流)
- ☐ 2.7 Another C++ Program: Adding Integers
- ☐ 2.8 Memory Concepts
- ☐ 2.9 Arithmetic
- ☐ **2.10 Decision Making: Equality and Relational Operators**



2.5 Streams(流)



- C++的输入/输出(I/O)是以**字节流**的形式实现的,流实际上就是一个**字节序列**.
 - ❖ 在**输入操作**中,字节从**输入设备**(如键盘、磁盘、网络连接等)流向**内存**;
 - ❖ 在**输出操作**中,字节从**内存**流向**输出设备**(如显示器、打印机、磁盘、网络连接等).



Topics



- ☐ 2.1 Introduction
- ☐ 2.2 First Program in C++: Printing a Line of Text
- ☐ 2.3 Standard library(标准库)
- ☐ 2.4 Modifying Our First C++ Program
- ☐ 2.5 Streams(流)
- ☐ **2.6 Stream Input(输入流)/ Stream Output(输出流)**
- ☐ 2.7 Another C++ Program: Adding Integers
- ☐ 2.8 Memory Concepts
- ☐ 2.9 Arithmetic
- ☐ **2.10 Decision Making: Equality and Relational Operators**



2.6 Stream Input(输入流)/ Stream Output(输出流)



- C++中提供了一套输入输出流类的对象, 它们是**cin**、**cout**和**cerr**等, 分别指向**终端输入**、**终端输出**和**标准出错输出**(也从终端输出)
- **cin**与**>>**一起完成输入操作, **cout**、**cerr**与**<<**一起完成输出与标准错误输出。

```
int i;// 变量, Ch2.7  
cin>>i;  
cout<<"Welcom to C++"<<endl;  
cout<<"Welcom to C++\n";
```

注意箭头的方向。在输出中使用**endl** (end of line), 表示换行, 注意最后一个字符是**'l'**, 而不是数字**1**, **endl**相当于C语言的**'\n'**, 表示输出一个换行。



2.6 Stream Input(输入流)/ Stream Output(输出流)



□ 标准输入流对象 **cin**

在C++程序中, 数据的输入(从键盘输入上读取数据)通常采用**cin**流对象来完成, 其格式如下:

❖ **cin**>>变量1;

❖ **cin**>>变量1>>变量2>>.....>>变量n;

❖ “>>”是流提取运算符.

□ 一般在该语句之前用**cout**输出一个需要输入数据的提示信息, 以正确引导和提示用户输入正确的数据:

❖ **cout**<< "请输入一个整数: ";



Topics



- ☐ 2.1 Introduction
- ☐ 2.2 First Program in C++: Printing a Line of Text
- ☐ 2.3 Standard library(标准库)
- ☐ 2.4 Modifying Our First C++ Program
- ☐ 2.5 Streams(流)
- ☐ 2.6 Stream Input(输入流)/ Stream Output(输出流)
- ☐ **2.7 Another C++ Program: Adding Integers**
- ☐ 2.8 Memory Concepts
- ☐ 2.9 Arithmetic
- ☐ **2.10 Decision Making: Equality and Relational Operators**

```
1 // Fig. 2.5: fig02_05.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 // function main begins program execution
6 int main()
7 {
8     // variable declarations
9     int number1; // first integer to add
10    int number2; // second integer to add
11    int sum; // sum of number1 and number2
12
13    std::cout << "Enter first integer: "; // prompt user for data
14    std::cin >> number1; // read first integer from user into number1
15
16    std::cout << "Enter second integer: "; // prompt user for data
17    std::cin >> number2; // read second integer from user into number2
18
19    sum = number1 + number2; // add the numbers; store result in sum
20
21    std::cout << "Sum is " << sum << std::endl; // display sum; end line
22
23    return 0; // indicate that program ended successfully
24
25 } // end function main
```

声明(Declarations)

三个变量(Variables).

int number1, number2, sum;

```
1 // Fig. 2.5: fig02_05.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 // function main begins program execution
6 int main()
7 {
8     // variable declarations
9     int number1; // first integer to add
10    int number2; // second integer to add
11    int sum; // sum of number1 and number2
12
13    std::cout << "Enter first integer: "; // prompt user for data
14    std::cin >> number1; // read first integer from user into number1
15
16    std::cout << "Enter second integer: "; // prompt user for data
17    std::cin >> number2; // read second integer from user into number2
18
19    sum = number1 + number2; // add the numbers; store result in sum
20
21    std::cout << "Sum is " << sum << std::endl; // display sum; end line
22
23    return 0; // indicate that program ended successfully
24
25 } // end function main
```

基本数据类型，如：double, char



2.7 Another C++ Program: Adding Integers



Integral Types	Floating-Point Types
<u>bool</u> 布尔型(真、假)	<u>float</u> 浮点型
<u>char</u> 字符型	<u>double</u> 双精度浮点型
signed char	long double
unsigned char	
short int	
unsigned short int	
<u>int</u> 整型	
unsigned int	
long int	
unsigned long int	
wchar_t	

```

1 // Fig. 2.5: fig02_05.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 // function main begins program execution
6 int main()
7 {
8     // variable declarations
9     int number1; // first integer to add
10    int number2; // second integer to add
11    int sum; // sum of number1 and number2
12
13    std::cout << "Enter first integer: ";
14    int number1; // first integer to add
15    int number2; // second integer to add
16    int sum; // sum of number1 and number2
17
18    // Get first integer
19    int number1; // first integer to add
20
21    // Get second integer
22    int number2; // second integer to add
23
24    // Calculate sum
25    int sum; // sum of number1 and number2
26
27    // Display sum
28    std::cout << "Sum is " << sum << std::endl; // display sum; end line
29
30    return 0; // indicate that program ended successfully
31 } // end function main

```

变量名：程序中用于保存值value的内存区域

1. 字母、数字、下划线(_)的组合, 大小写敏感;

2. 不能以数字打头, 不建议_打头;

3. 不能和关键字重复;

4. 变量的使用必须在声明之后;

5. 应使用有意义的变量名, 例如采用匈牙利命名法.

er1

ber2

um



2.7 Another C++ Program Adding Integers



□ 匈牙利命名法

- ❖ 微软的总设计师, 查尔斯·西蒙尼发明.
- ❖ 在匈牙利命名法中, 一个变量名由一个或多个**小写字母**开始, 这些字母有助于记忆变量的类型; 紧跟着的就是**首字母大写**的由程序员选择的有助于描述变量作用的名词.

□ 例子

- ❖ **bBusy** : 布尔型
- ❖ **nSize** : 整型
- ❖ **fPrice** : 浮点数
- ❖ **pFoo** : 指针

```
1 // Fig. 2.5: fig02_05.cpp
2 // Addition program that displays the sum of two numbers.
3 #include <iostream> // allows program to perform input and output
4
5 // function main begins program execution
6 int main()
7 {
8     // variable declarations
9     int number1; // first integer to add
10    int number2; // second integer to add
11    int sum; // sum of number1 and number2
12
13    std::cout << "Enter first integer: "; // prompt user for data
14    std::cin >> number1; // read first integer from user into number1
15
16    // 从键盘输入, 用回车键表示输入结束
17    std::cin >> number2; // read second integer from user into number2
18
19    sum = number1 + number2; // add the numbers; store result in sum
20
21    std::cout << "Sum is " << sum << std::endl; // display sum; end line
22
23    return 0; // indicate that program ended successfully
24
25 }
```



```
Enter first integer: 45
Enter second integer: 72
Sum is 117
```

it

```
6  int main()
7  {
8      // variable declarations
9      int number1; // first integer to add
10     int number2; // second integer to add
11     int sum; // sum of number1 and number2
12
13     std::cout << "Enter first integer: "; // prompt user for data
14     std::cin << number1;
15
16     std::cout << "Enter second integer: "; // prompt user for data
17     std::cin << number2;
18
19     sum = number1 + number2; // add the numbers; store result in sum
20
21     std::cout << "Sum is " << sum << std::endl; // display sum; end line
22
23     return 0;
24
25 } // end func
```

很多std::, 如何简化?

=: 赋值运算符 (assignment operator) ;

+: 算术运算符 (Arithmetic operator) , 例如+, -, *, /, %

等价于:

```
std::cout << "Sum is " << number1 + number2 <<
std::endl;
```



Topics



- ☐ 2.1 Introduction
- ☐ 2.2 First Program in C++: Printing a Line of Text
- ☐ 2.3 Standard library(标准库)
- ☐ 2.4 Modifying Our First C++ Program
- ☐ 2.5 Streams(流)
- ☐ 2.6 Stream Input(输入流)/ Stream Output(输出流)
- ☐ 2.7 Another C++ Program: Adding Integers
- ☐ **2.8 Memory Concepts**
- ☐ 2.9 Arithmetic
- ☐ 2.10 Decision Making: Equality and Relational Operators



2.8 Memory Concepts



□ Variables (变量)

- ❖ Variable names correspond to **locations in the computer's memory**
- ❖ Every variable has a **name**, a **type**, a **size** and a **value**
- ❖ Whenever a new value is placed into a variable (through **cin**, for example), it replaces (and destroys) the previous value
- ❖ Reading variables from memory does not change them (读出变量的值不会改变存储器的值).

□ **number1**和它对应的值

number1

45



2.8 Memory Concepts



❑ `std::cin >> number1;` // read first integer from user into number1

执行之后:

number1	45
---------	----

❑ `std::cin >> number2;` // read second integer from user into number2

执行之后:

number1	45
number2	72

❑ `sum = number1 + number2;` // add the numbers; store result in sum

执行之后:

number1	45
number2	72
sum	117

}
nondestructive !



2.8 Memory Concepts



□ 思考:

```
int iSum = 10;
```

```
iSum = iSum + 1;
```



Q & A



1. 指出下面语句段存在的问题:

```
int num, a, b;  
std::cout >> num;  
std::cin >> a, b;
```

2. 什么是基本数据类型? 说出至少4种:

```
int, char, bool, float, double
```

3. 简述变量概念及其属性. 哪些是正确的变量名?

```
3number, $number, return, Hello, _number, hello
```



Q & A



1. 设变量 $x=2$, $y=3$, 下列语句是否有输出, 如有是什么?

- `cout<<x;`
- `cout<<x+x;`
- `cout<<x+y<<"="<<y+x;`
- `cout<<"x+y="<<y+x;`



Topics



- ☐ 2.1 Introduction
- ☐ 2.2 First Program in C++: Printing a Line of Text
- ☐ 2.3 Standard library(标准库)
- ☐ 2.4 Modifying Our First C++ Program
- ☐ 2.5 Streams(流)
- ☐ 2.6 Stream Input(输入流)/ Stream Output(输出流)
- ☐ 2.7 Another C++ Program: Adding Integers
- ☐ 2.8 Memory Concepts
- ☐ **2.9 Arithmetic**
- ☐ **2.10 Decision Making: Equality and Relational Operators**



2.9 Arithmetic (算术运算)



□ Arithmetic calculations(算术运算)

❖ * multiplication(乘)

❖ / division (除)

- $\text{int1}/\text{int2}$: 整除, truncates(截去) remainder(余数)

- $7 / 5$ evaluates to 1

❖ % Modulus operator(取模) returns the remainder

- $7 \% 5$ evaluates to 2

□ Operator precedence(运算符优先级)

❖ Example: Find the average of three variables a, b and c

- $a + b + c / 3$ $(a + b + c) / 3$



2.9 Arithmetic (算术运算)



C++ operation	C++ arithmetic operator	Algebraic expression	C++ expression
Addition	+	$f + 7$	<code>f + 7</code>
Subtraction	-	$p - c$	<code>p - c</code>
Multiplication	*	bm or $b \cdot m$	<code>b * m</code>
Division	/	x / y or $\frac{x}{y}$ or $x \div y$	<code>x / y</code>
Modulus	%	$r \bmod s$	<code>r % s</code>

Operator(s)	Operation(s)	Order of evaluation (precedence)
()	Parentheses	优先级最高，最内层的括号最先运算，同级的括号，从左到右。
*	Multiplication	to right. Evaluated second. If there are several, they are
/	Division	优先级次之，从左到右。
%	Modulus	
+	Addition	优先级最低，从左到右。
-	Subtraction	



2.9 Arithmetic (算术运算)



Algebra:
$$m = \frac{a + b + c + d + e}{5}$$

C++:
$$m = (a + b + c + d + e) / 5;$$

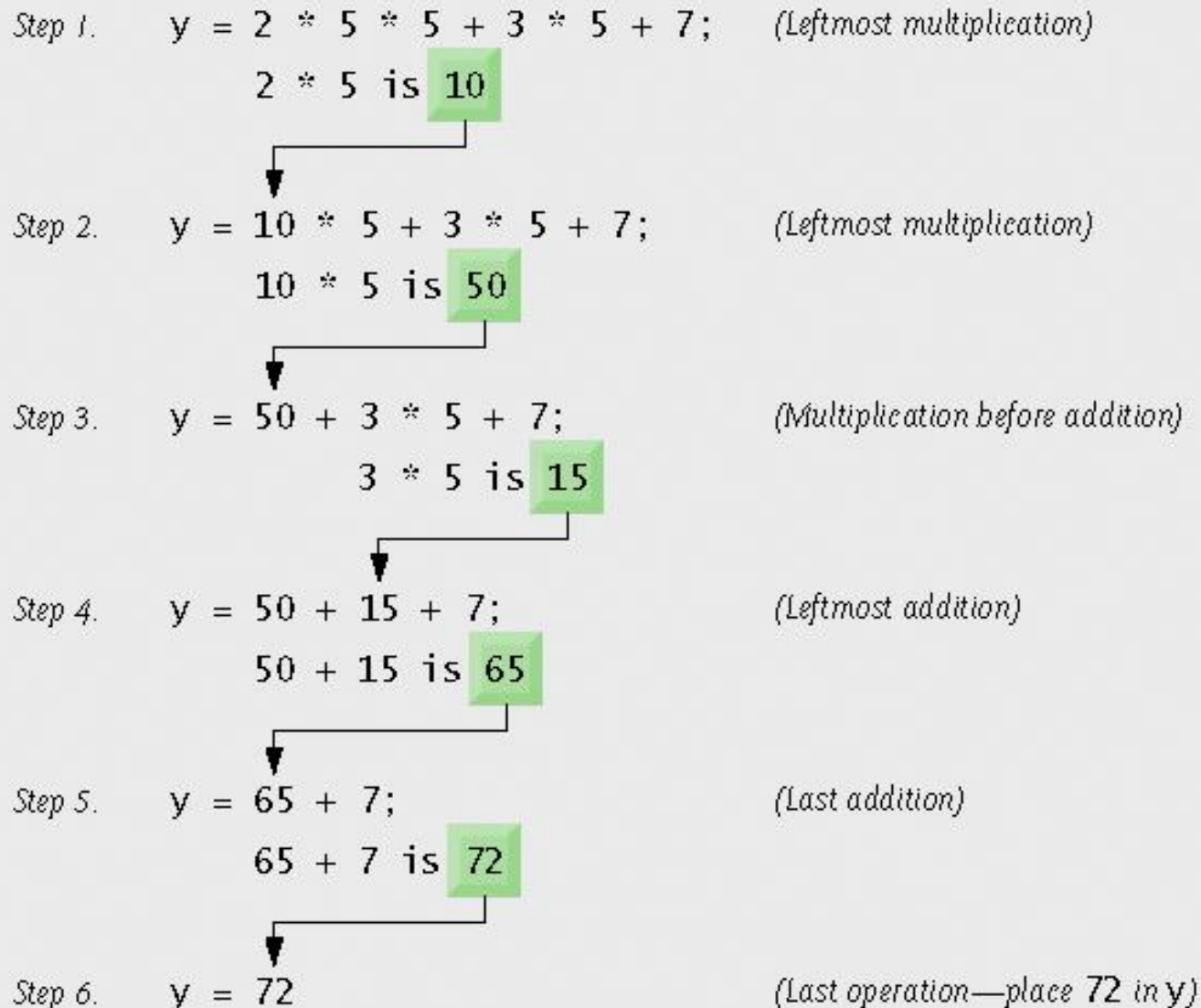
Algebra:
$$y = mx + b$$

C++:
$$y = m * x + b;$$



2.9 Arithmetic (算术运算)

-优先级实例





Topics



- ☐ 2.1 Introduction
- ☐ 2.2 First Program in C++: Printing a Line of Text
- ☐ 2.3 Standard library(标准库)
- ☐ 2.4 Modifying Our First C++ Program
- ☐ 2.5 Streams(流)
- ☐ 2.6 Stream Input(输入流)/ Stream Output(输出流)
- ☐ 2.7 Another C++ Program: Adding Integers
- ☐ 2.8 Memory Concepts
- ☐ 2.9 Arithmetic
- ☐ **2.10 Decision Making: Equality and Relational Operators**



2.10 Decision Making: Equality and Relational Operators

(判断：相等与关系运算符)



Standard algebraic equality or relational operator	C++ equality or relational operator	Sample C++ condition	Meaning of C++ condition
<i>Relational operators</i>			
>	>	<code>x > y</code>	x is greater than y
<	<	<code>x < y</code>	x is less than y
≥	>=	<code>x >= y</code>	x is greater than or equal to y
≤	<=	<code>x <= y</code>	x is less than or equal to y
<i>Equality operators</i>			
=	==	<code>x == y</code>	x is equal to y
≠	!=	<code>x != y</code>	x is not equal to y



2.10 Decision Making: Equality and Relational Operators

(判断：相等与关系运算符)



if (**condition**)

statements of if-body

if control structure

□ If a **condition** is true, then the body of the if statement executed

❖ 0 is false, non-zero is true



2.10 Decision Making: Equality and Relational Operators

(判断：相等与关系运算符)



- 例： Figure 2.13. Equality and relational operators
判断两个整数的大小关系。


```

1 // Fig. 2.13: fig02_13.cpp
2 // Comparing integers using if statements, relational
3 // and equality operators.
4 #include <iostream> // allows program to perform I/O
5
6 using std::cout; // program uses cout
7 using std::cin; // program uses cin
8 using std::endl; // program uses endl
9
10 // function main begins program execution
11 int main()
12 {
13     int number1; // first integer to compare
14     int number2; // second integer to compare
15
16     cout << "Enter two integers to compare: ";
17     cin >> number1 >> number2; // read two integers
18
19     if ( number1 == number2 )
20         cout << number1 << " == " << number2 << endl;
21
22     if ( number1 != number2 )
23         cout << number1 << " != " << number2 << endl;
24
25     if ( number1 < number2 )
26         cout << number1 << " < " << number2 << endl;
27
28     if ( number1 > number2 )
29         cout << number1 << " > " << number2 << endl;
30

```

Line 6-8: **using** 声明

- then we can use

cout instead of **std::cout**

cin instead of **std::cin**, and

endl instead of **std::endl**

两个输入间以空白字符分隔, 如**空格**、**TAB键**或者**回车键**, **3<SP>7<RET>**

条件判断语句
体内只有一条语句, 缩进

```
31  if ( number1 <= number2 )
32      cout << number1 << " <=" << number2 << endl;
33
34  if ( number1 >= number2 )
35      cout << number1 << " >=" << number2 << endl;
36
37  return 0; // indicate that program ended successfully
38
39 } // end function main
```

```
Enter two integers to compare: 3 7
3 != 7
3 < 7
3 <= 7
```

```
Enter two integers to compare: 22 12
22 != 12
22 > 12
22 >= 12
```

```
Enter two integers to compare: 7 7
7 == 7
7 <= 7
7 >= 7
```



2.10 Decision Making: Equality and Relational Operators

(判断：相等与关系运算符)



BOOK. P48

Operators				Associativity	Type	
()				left to right	parentheses	
*	/	%		left to right	multiplicative	算术
+	-			left to right	additive	
<<	>>			left to right	stream insertion/extraction	流
<	<=	>	>=	left to right	relational	关系
==	!=			left to right	equality	等值
=				right to left	assignment	赋值

同一优先级的运算符具备相同的结合性!



2.10 Decision Making: Equality and Relational Operators

(判断：相等与关系运算符)



- ❑ **Associativity (结合性) and Precedence(优先级)**
- ❑ 表达式求值时, 首先按运算符的**优先级**从高到低执行, 其次同一优先级运算符根据**结合性**处理:
 - ❖ $a+b-c*d+e/f$ // 从左到右
 - ❖ $a=b=c+d$ // 从右到左
- ❑ 如果不确定优先顺序, 则加括号以确保正确!



Q & A



□ 1. 结合Precedence和Associativity概念, 简述表达式的求值过程.

❖ $a = b = c + d \% e;$

□ 2. 解释程序段的输出:

```
int num = 0;
```

```
if (num == 1)
```

```
    cout << "num is 1" << endl;
```

```
cout << "num is " << num << endl;
```



Q & A



- 3. 编写程序，读入3个整数，输出他们的最大值。
- 4. 编写程序，读入一个3位整数，将其逆序输出。如输入234，输出432。



4. 编写程序，读入一个**3**位整数，将其逆序输出。 。如输入**234**，输出**432**。



```
int main()
{
    int n1,n2,n3;
    int number;

    cout<<"please input a three digits number: ";
    cin>>number;

    n3=number%10;
    n2=(number%100)/10;
    n1=number/100;

    cout<<"the inverse number is: "<<n3<<n2<<n1<<endl;

    return 0;
}
```



总结



- ❑ 理解简单的C++程序
- ❑ 简单的输入、输出语句
- ❑ 使用基本的数据类型
- ❑ 理解变量和程序对内存的使用
- ❑ 使用算术运算符
- ❑ 算术和关系运算符的优先级关系
- ❑ 简单的判断语句



Homework



□ 实验必选题目:

2.19, 2.29

□ 实验任选题目:

2.31

□ 作业题目(Homework):

2.9, 2.17, 2.24

QQ群 673534850