



Chap01 绪论

程勇

北京化工大学

信息科学与技术学院计算机系

Sept. 2020

课程提纲

- ✦ Chap01 绪论
- ✦ Chap02 面向对象程序设计方法
- ✦ Chap03 重载与类型转换
- ✦ Chap04 继承与派生
- ✦ Chap05 多态性
- ✦ Chap06 输入输出流
- ✦ Chap07 容错与异常处理
- ✦ Chap08 模版

本章提纲

✚ 编程语言简介

- ❑ C++的产生
- ❑ C++的特点
- ❑ C++应用
- ❑ C/C++比较

✚ 一个简单的C++程序

✚ 程序开发环境

- ❑ 开发工具
- ❑ 编译过程

✚ 基本概念

- ❑ 字符集
- ❑ 关键字
- ❑ 标识符
- ❑ 变量/类型/函数/命名空间/语句
- ❑ 注释
- ❑ 标准库

✚ 编程风格

编程语言简史

✚ 计算机发展早期

- ❑ 机器语言

- ❑ 汇编语言

✚ 二十世纪50年代中期

- ❑ FORTRAN

- 首先出现的高级程序设计语言，在计算机语言发展史上具有划时代的意义；
- 引入了变量、数组、循环、分支等仍在使用的程序设计概念；
- 广泛应用于科学和工程计算中；

✚ 二十世纪50年代后期

- ❑ ALGOL

- ALGOL60提出了块结构的思想，由“Begin-End”实现块结构。避免同名变量相互混淆，这实际上是一种初级封装；

编程语言简史(续)

✚ 二十世纪60年代

❑ Simula 67

- 首先提出对象和类的概念，并支持类的继承，是面向对象语言的鼻祖；

✚ 二十世纪70年代

❑ Ada

- 支持数据抽象类型(ADT)的最重要的语言之一；
- 数据抽象类型是一种数据结构及作用在数据结构上操作组成的实体；
- 不能全面地支持继承，称为基于对象的语言；

❑ Smalltalk

- 并入了Simula 67中面向对象的特征，信息的隐藏更加严格，每种实体都是对象；
- 是一种弱类型化语言，也是最有影响的面向对象语言之一；

编程语言简史(续)

✚ 二十世纪80年代以后

出现了更多面向对象语言，归纳起来，大致可以分为以下两类：

❑ 开发全新的面向对象语言

- Smalltalk80
- Object-C
- Eiffel
- Java

❑ 对传统语言进行面向对象扩展

■ C++语言

- ✦ 混合型语言，在C语言的基础上增加了面向对象的支持；
- ✦ 既支持传统的面向过程程序设计，又支持新型的面向对象程序设计；
- ✦ 由于C语言的广泛普及和开发支持，C++语言得以广泛应用；

C++语言概述

C++的产生

- ❑ 1960年，出现ALGOL60语言是一种面向问题的高级语言，不适合用来编写系统程序；
- ❑ 1963年，英国剑桥大学推出CPL(Combined Programming Language)语言，该语言比ALGOL60离硬件更近一些，但规模较大，难以实现；
- ❑ 1967年，剑桥大学的Martin Richards对CPL语言进行了简化，推出BCPL(Basic CPL)语言；
- ❑ 1970年，美国贝尔实验室的Ken Thompson对BCPL语言进一步简化，设计了B语言(取BCPL第一个字母)，并用B语言写出了第一个UNIX操作系统；
- ❑ 1972年，D. M. Ritchie在B语言的基础上设计了C语言，既保持了BCPL和B语言的优点，又克服了它们的缺点；

C++的产生(续)

- ❑ 1977年，出现了不依赖于具体机器的C语言编译文本《可移植C语言编译程序》，这不仅大大提高了C程序的可移植性，同时也推动了UNIX操作系统的迅速发展；
- ❑ 1983年，美国国家标准化协会(ANSI)综合C的各种版本，制定了新的标准ANSI C，进而演化为87ANSI C；
- ❑ 1980年，美国贝尔实验室的Bjarne Stroustrup开始为C增加面向对象特性，最初为“带类的C”，1983年正式定名为C++；
- ❑ 1991年，C++3.0增加了模版，4.0增加了异常处理、命名空间和运行时类型识别等功能；
- ❑ 1998年，C++的国际标准由ISO/IEC14882文档定义，并由ANSI发布；
 - <http://webstore.ansi.org/ansidocstore/default.asp>

C++主要特点

✦ C语言主要特点

- ❑ 有丰富的运算符和数据类型;
- ❑ C语言具备高级语言特征, 又具有低级语言特征;
- ❑ 以函数为基础实现程序的结构化设计;
- ❑ 语言简洁、紧凑, 使用方便、灵活;

✦ C++主要特点

- ❑ C++表示对C的累加, 是一种基于C的面向对象语言, 既可进行过程化程序设计, 又可进行面向对象程序设计;
- ❑ C++实现了类的封装, 数据隐藏、继承及多态, 使其代码可重用并容易维护;
- ❑ C++与C完全兼容, 很多用C编写的库函数和应用程序都可以为C++所用;
- ❑ C++在标准库中提供了丰富的功能;

一个简单的C++程序

✚ 第一个C++程序

// prog01_01.cpp : 第一个C++程序。

```
#include <iostream>
```

```
// 引用命名空间std中的元素
```

```
using std::cout;
```

```
using std::endl;
```

```
int main() {
```

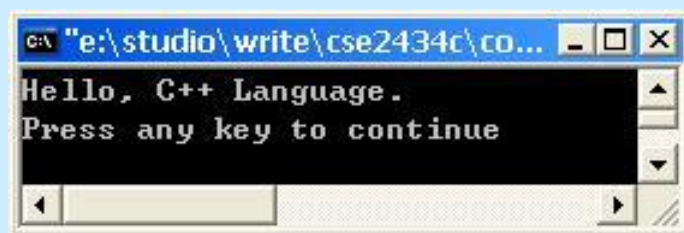
```
    // 向屏幕输出结果
```

```
    cout << "Hello, C++ Language." << endl;
```

```
    return 0;
```

```
}
```

运行结果



基本概念

✚ 字符集

- ❑ ASCII码字符集是计算机中的常用字符集。它包括英文字母及阿拉伯数字等128个字符，存储一个ASCII码占用一个字节单元；
- ❑ 汉字国标码用于汉字处理。国标码的存储占用两个字节单元；
- ❑ Unicode字符集对各类字符进行统一的双字节编码。它包括了世界上多种语言的基本字符，最多可容纳65536个字符。
 - ISO/IEC 10646标准指定；
 - ASCII码字符集、国标码字符集都是它的子集；
 - <http://www.unicode.org>

基本概念(续)

✚ C++语言基本字符

小写字母	abcdefghijklmnopqrstuvwxyz
大写字母	ABCDEFGHIJKLMNOPQRSTUVWXYZ WXYZ
阿拉伯数字	0 1 2 3 4 5 6 7 8 9
其他符号	+ - * / = , . _ : ; ? \ " ' ~ ! # % & () [] {} ^ < > 空格

基本概念(续)

✦ 标准C++关键字(74个)

数据类型说明符 与修饰符	bool char wchar_t class const double enum float int long short signed struct union unsigned void volatile
存储类型说明符	auto extern inline register static
访问说明符	friend private protected public
其它说明符	asm explicit namespace operator template this typedef typename using virtual
语句与标号	break case catch continue default do else for goto if return switch throw try while
运算符及逻辑值	delete false new sizeof true

基本概念(续)

- ✚ 标识符是程序员用来表示变量、常量、数据类型、函数、命名空间等的名称。
 - ❑ 由字母或下划线开始，由字母、数字、下划线组成，其有效长度为1-31个字符；
 - ❑ 使用有一定含义的英文单词或拼音序列作标识符，以提高程序可读性；
 - ❑ 尽量不用下划线或双下划线打头，以免与系统定义的关键字相冲突；
- ✚ 例
 - ❑ 合法标识符
 - MyOffice、amount、_sum、num_of_student等
 - ❑ 非法标识符
 - \$amount、f3.78、void、MySalary 2000等

基本概念(续)

✚ 变量

- 在程序中是指可以改变值的量。

✚ 常量

- 在程序中恒定不变的量，可分为字面常量和常变量。

✚ 语句

- 程序所处理数据元素的基本单位。

✚ 函数

- 自包含的、可命名的执行代码块。

✚ 命名空间

- 用一个集合名称标识程序中一组名称项的方法。

基本概念(续)

- + 注释主要用于解释代码含义，增加程序可阅读性，有两种注释形式：

- 单行注释

- `// the length of the box.`

- 多行注释

- `/*...*/`

- + 标准库

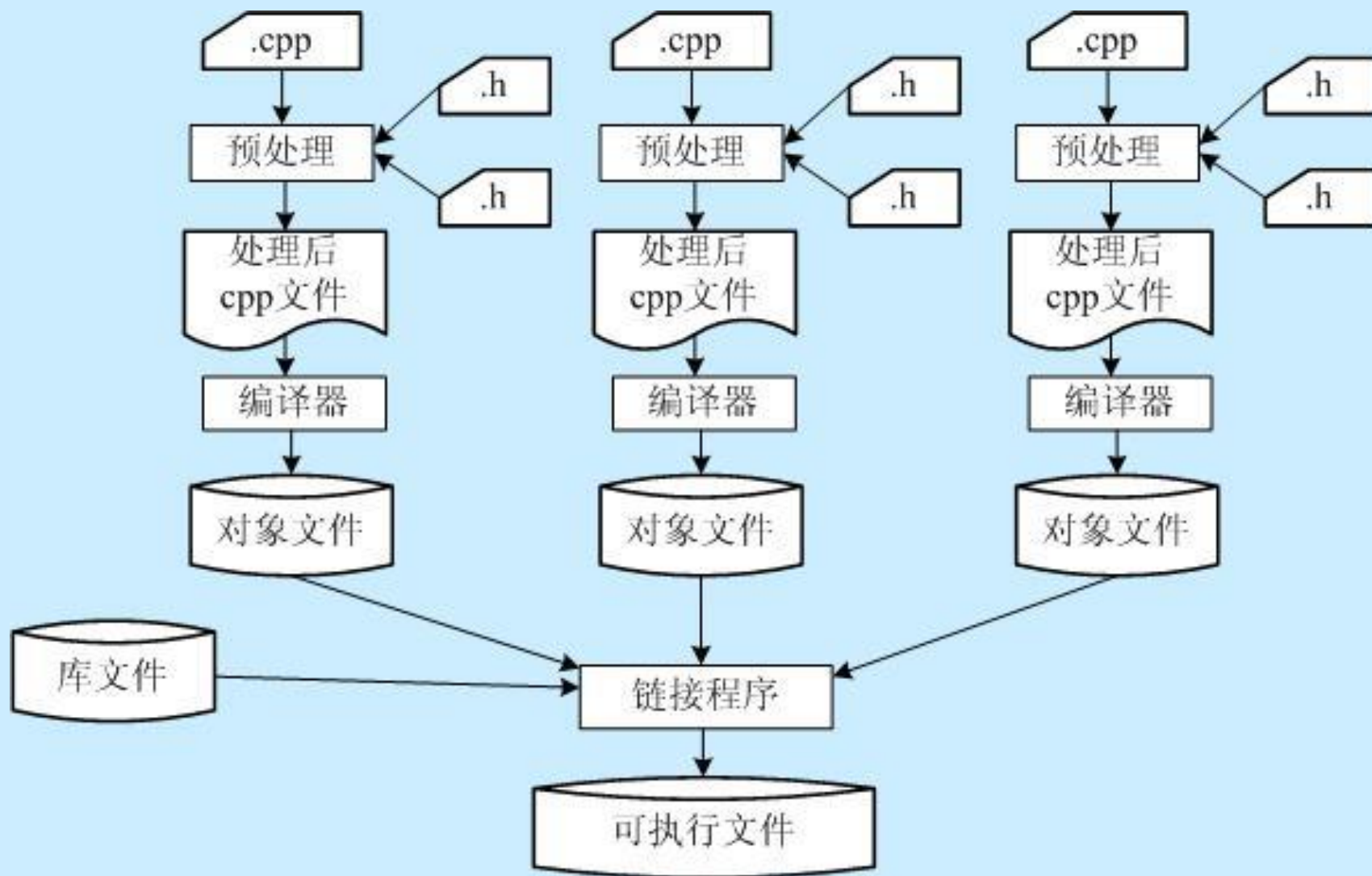
- 标准库是C++语言的重要组成部分，它提供了丰富的功能，大大扩展了C++语言的能力；
 - 标准库中的几乎所有内容都在命名空间std中定义，需要使用using指令引入；

编程风格

✚ 编程风格的主要作用是提高代码的可阅读性。

- ❑ 始终如一地坚持一种切合实际的约定；
- ❑ 函数采用作用性的名称；
- ❑ 全局变量使用具有说明性的名字，局部变量使用短名字；
- ❑ 使用一致的缩排和加括号风格；
- ❑ 加括号排除表达式二义性；
- ❑ 避免函数宏，给宏的体和参数都加上括号；
- ❑ 给函数和全局数据加上注释；
- ❑ 代码和注释要统一，不要相互矛盾；
- ❑ 不要注释差的代码，要进行重写；
- ❑

程序编译过程



本章小结

✦ 编程语言简史

✦ C++语言概述

- ❑ C++的产生
- ❑ C++的特点
- ❑ 一个简单的C++程序

✦ 基本概念

- ❑ 字符集
- ❑ 关键字
- ❑ 标识符
- ❑ 变量/类型/函数/命名空间/语句
- ❑ 注释
- ❑ 标准库

✦ 编程风格

✦ 程序开发过程

本章小结

✚ 课程网站

□ <http://www.jiaowu.buct.edu.cn>