C++语言程序设计

第零章: 绪论

宋霜

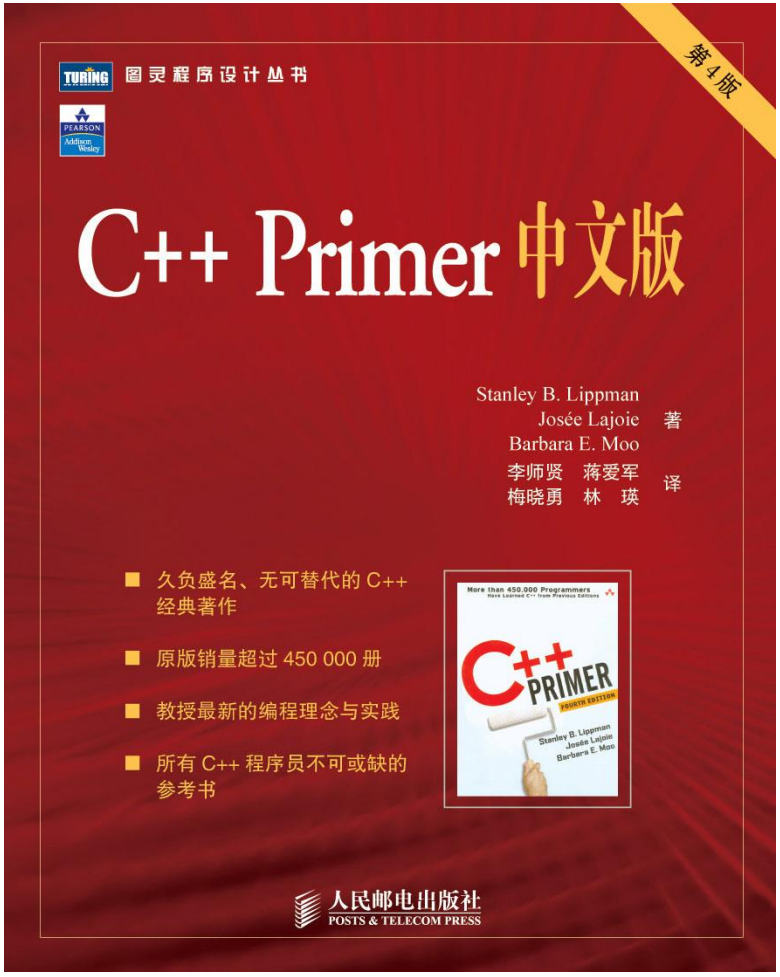
哈尔滨工业大学（深圳）

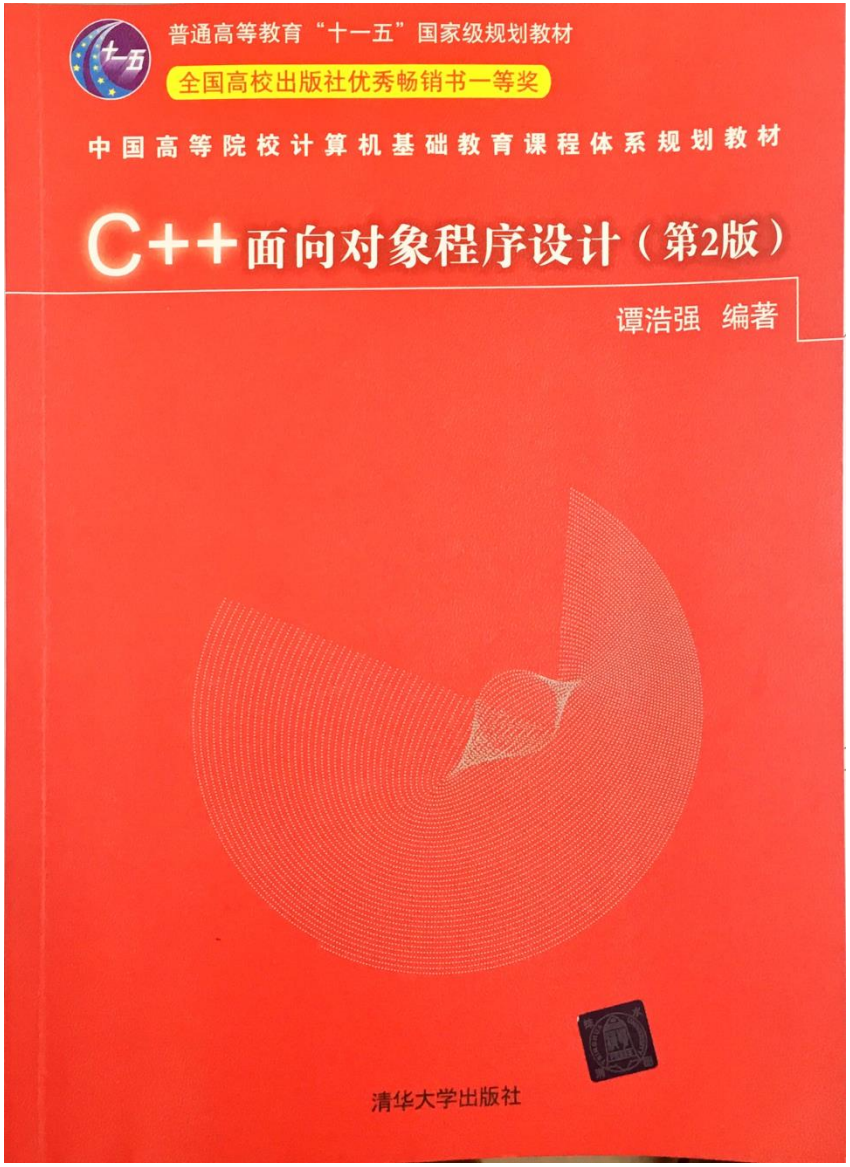
机电工程与自动化学院

邮箱: songshuang@hit.edu.cn

第零章：绪论

教材


TURING 图灵程序设计丛书
第4版
C++ Primer 中文版
Stanley B. Lippman
Josée Lajoie 著
Barbara E. Moo
李师贤 蒋爱军
梅晓勇 林 瑛 译
久负盛名、无可替代的 C++ 经典著作
原版销量超过 450 000 册
教授最新的编程理念与实践
所有 C++ 程序员不可或缺的参考书
人民邮电出版社
POSTS & TELECOM PRESS


普通高等教育“十一五”国家级规划教材
全国高校出版社优秀畅销书一等奖
中国高等院校计算机基础教育课程体系规划教材
C++面向对象程序设计（第2版）
谭浩强 编著
清华大学出版社

第零章：绪论

群名称:2020秋季C++语言程序设计

群　号:694442601

第零章：绪论

参考

- 数据结构
- 算法导论
- 软件工程
- 操作系统
- 编译原理
- 微机原理

第零章：绪论

课程内容

- C++初步知识
- 类与对象
- 运算符重载
- 继承与派生
- 多态与虚函数
- 编程规范
- Windows GUI 编程

第零章：绪论

实验内容

- 基本输入输出-算术表达式识别
- 类和对象-简易计算器
- 运算符重载-矩阵运算
- GUI编程-科学计算器

第零章：绪论

课程成绩

- 实验：10分*4
  - 实验签到：2分*4
  - 实验代码：5分*4
  - 实验报告：3分*4
- 期末考试：60%
  - 选择题
  - 填空题
  - 编程题

C++语言程序设计

第一章: C++新特征

宋霜

哈尔滨工业大学（深圳）

机电工程与自动化学院

邮箱: songshuang@hit.edu.cn

第一章：C++新特征

C

- 结构化和模块化
- 面向过程

C++

- 增加了面向对象机制
- 标准模版库

第一章: C++新特征

Hello World

```cpp
//main.cpp
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

```c
/*main.c*/
#include <stdio.h>

int main()
{
    printf("Hello world!\n ");
    return 0;
}
```

示例程序1

第一章: C++新特征

iostream

- istream：输入流
- ostream：输出流
- 流：指要从某种IO设备上读入或写出的字符序列
- 标准库中的IO对象
  - cin：标准输入
  - cout：标准输出
  - cerr：标准错误，输出警告和错误信息
  - clog：产生程序执行的一般信息

第一章: C++新特征

iostream

```
#include <iostream>
int main()
{
    std::cout << "Enter two numbers:" <<std::endl;
    int v1,v2;
    std::cin>>v1>>v2;
    std::cout<< "The sum of " <<v1<< " and " <<v2
             << " is " <<v1+v2<<std::endl;
    return 0;
}
```

endl：操纵符（manipulator），将它写入输出流时，具有换行的效果，并刷新与设备相关的缓冲区（buffer）

第一章：C++新特征

命名空间

```
#include <iostream>

std::cin

std::cout
```

- 前缀std::表明cin和cout是定义在命名空间std中
- :: 作用域操作符
- 使用命名空间，可以避免使用了与程序中所定义名字相同的名字而引起冲突

```
#include <iostream>

using namespace std;

cout << “Hello world!” << endl;
```

```
#include <iostream>

using std::cout;

cout << “Hello world!” << endl;
```

第一章：C++新特征

注释

两种注释方法：

- 单行注释：　//
- 成对注释：　/*　*/
  - 注释对不可嵌套

- cout<<“a+b+c”<<endl;
- cout<<“/*a+b*/+c”<<endl;
- cout<</*　“a+/*b+c*/”　*/<<endl;

第一章: C++新特征

注释

两种注释方法:

- 单行注释: //
- 成对注释: /* */
  - 注释对不可嵌套

```
cout<< "a+b+c" <<endl;
cout<< "/*a+b*/+c" <<endl;
cout</* "a+/*b+c*/" */<<endl;
```

第一章：C++新特征

引用

- Reference
- 对象的别名，实际应用中引用主要用作函数的形式参数
  - int a;
  - int &b=a;
  - a=1;
  - cout<<a<<" "<<b<<endl;
  - b=2;
  - cout<<a<<" "<<b<<endl;
- 引用定义必须初始化

第一章: C++新特征

引用

- Reference
- 对象的别名，实际应用中引用主要用作函数的形式参数

```
void swap(int a, int b)
{
    int tmp;
    tmp=a;
    a=b;
    b=tmp;
}
```

```
int a=4,b=6;
swap(a,b)
cout<<a<< " "<<b<<endl;
```

第一章: C++新特征

引用

- Reference
- 对象的别名，实际应用中引用主要用作函数的形式参数

```
void swap(int *a, int *b)
{
    int tmp;
    tmp=*a;
    *a=*b;
    *b=tmp;
}
```

```
int a=4,b=6;
swap(&a,&b)
cout<<a<< " "<<b<<endl;
```

第一章: C++新特征

引用

- Reference
- 对象的别名，实际应用中引用主要用作函数的形式参数

```
void swap(int &a, int &b)
{
    int tmp;
    tmp=a;
    a=b;
    b=tmp;
}
```

```
int a=4,b=6;
swap(a,b)
cout<<a<< " "<<b<<endl;
```

第一章: C++新特征

引用

- 引用与指针

Reference	Pointer

int a=10;
int &r=a;
r=5;
	int a=10;
	int *r=&a;
	*r=5;

1. 引用必须在创建时初始化，指针可以随时初始化
2. 引用被初始化到一个变量后，其不能再更改为其他变量的引用，指针可以随时指向其他变量。
3. 不能有空引用，引用必须关联到一个合法的内存空间。指针可以指向空。

第一章: C++新特征

字符串变量

- String
- 字符串赋值操作
- 字符串内字符操作
- cin ； cout
- 字符串连接操作： +
- 字符串比较操作： ==, >, >=, <, <=, !=

第一章：C++新特征

函数重载

- 函数重载的规则
  - 函数名称必须相同。
  - 参数列表必须不同（个数不同、类型不同、参数排列顺序不同等）。
  - 函数的返回类型可以相同也可以不相同。
  - 仅仅返回类型不同不足以成为函数的重载。

第一章: C++新特征

默认参数

```
double volumn(double r=1, double h=1)
{
    return pi*r*r*h;
}
```

- 必须在函数调用前将默认值的信息通知编译系统
- 注意二义性

函数模板

- 模板语言实现MAX函数

```
template<typename T>

T Max( T a, T b)

{

        return  a > b ? a : b;

  }
```

```
Max(10, 5);

Max(10.5, 5.5);

Max( ‘a’ , ‘c’ );
```

顺利运行

示例程序2

第一章：C++新特征

变量

- 变量提供了程序可以操作的有名字的存储区
- 变量都有一个类型，该类型决定了变量的内存大小和布局、取值范围、操作集合。
- 左值
  - 可以出现在赋值语句的左边或者右边
- 右值
  - 只能出现在赋值的右边
- 定义：用于为变量分配存储空间，变量必须且仅能定义一次
- 声明：用于向程序表明变量的类型和名字。extern。可以多次声明;

第一章：C++新特征

作用域

- 作用域用大括号来界定
- 名字从其声明点开始直到其声明所在的作用域结束处可见
- 全局作用域
  - 定义在所有函数外部的名字具有全局作用域
- 局部作用域
  - 如定义在函数内部
- 语句作用域
  - 如定义在for语句的作用域中
- 通常把一个对象（变量）定义在它首次使用的地方

示例程序3

第一章: C++新特征

作用域

- 作用域用大括号来界定
- 名字从其声明点开始直到其声明所在的作用
- 全局作用域
  - 定义在所有函数外部的名字具有全局作
- 局部作用域
  - 如定义在函数内部
- 语句作用域
  - 如定义在for语句的作用域中
- **通常把一个对象（变量）定义在它首次使用的地方**

```
#include <iostream>

using namespace std;

int global_val;

int main()
{
    int local_val;

    for (int i=0;i<10;i++)
    {
        int val=10;

        cout << i*val << endl;
    }

    return 0;
}
```

示例程序3

第一章: C++新特征

声明 与 定义

```
double volumn(double r=1, double h=1);  声明

int main()
{
    cout<< volumn()<<endl;
    cout<< volumn(2)<<endl;
    cout<< volumn(2,2)<<endl;
    return 0;
}

double volumn(double r, double h)  定义
{
    return pi*r*r*h;
}
```

第一章: C++新特征

声明 与 定义

```
//in file a
extern double r;
```

声明

```
//in file b
double r;
```

定义

- 使用前需声明，可多次声明，但只能一次定义
- 变量定义会申请内存空间，也可能为变量赋一个初始值。
- 变量显式初始化的声明也会成为定义
- 函数声明与定义区别在于后者多个函数体
- 声明放在头文件中，定义放在cpp文件中，方便调用。

new与delete

- 动态的创建和释放空间函数

```
int *p=new int;
delete p;
int *p=new int(10);
delete p;
int *p=new int[10];
delete [] p;
```

- 执行delete操作后，要

```
p=NULL;
```

示例程序4