

面向对象方法与C++程序设计

第1章 面向对象概述

大连理工大学
主讲人-赵小薇



简单C++程序结构



```
// my first program in C++  
#include <iostream>  
using namespace std;  
int main()  
{  
    cout << "Hello, C++ !";  
    return 0;  
}
```

单行注释

预处理指令

命名空间

cout为输出对象



注释



- 有两种形式：单行注释和多行注释
- 单行注释以双斜杠开头 (//)。例如：
- 多行注释以/*开始到*/结束，多行注释如下：

```
// my first program in C++  
float pi; //数学中的 $\pi$ 
```

```
/*注释开始
```

```
项目功能：贪吃蛇
```

```
作者：张三
```

```
完成日期：2011.4.5
```

```
修改记录：无
```

```
注释结束*/
```



预处理指令与头文件



```
# include <iostream>
```

- 头文件包含指令：头文件的代码可以是C++标准程序库中的文件，也可以是程序员创建自己的头文件。
- 如果此程序没有此行，编译就会出现错误。

```
cout << "Hello, C++ !";
```

- 在这个程序中，名称cout在头文件iostream中定义，这是一个标准的头文件它提供了在C++中使用标准输入和输出功能所需要的定义。<iostream>头文件包含了cout的定义，没有它，编译器就不知道cout是什么。



命名空间



- 命名空间：程序中的元素可以选择使用任何合乎规范的名称，如果两个或多个程序员为同一个大型工程的不同部分工作，也会有潜在的名称冲突。命名空间就解决了这个问题。

```
using namespace std;
```

- C++ 标准库中的实体名都用std来限定。cout的全名就是std::cout，其中的两个冒号称作范围解析运算符，std为命名空间，cout为对象名，std::cout表示命名空间std中的对象名cout。如果省略using指令，就必须把输出语句写为：

```
std::cout << " Welcome to C++ World!";
```



输入与输出



*/*C输入输出*/*

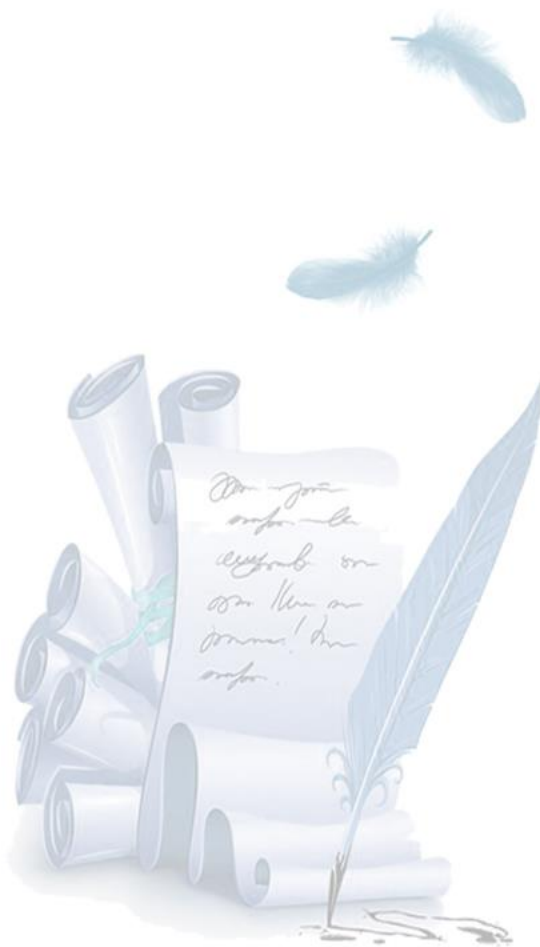
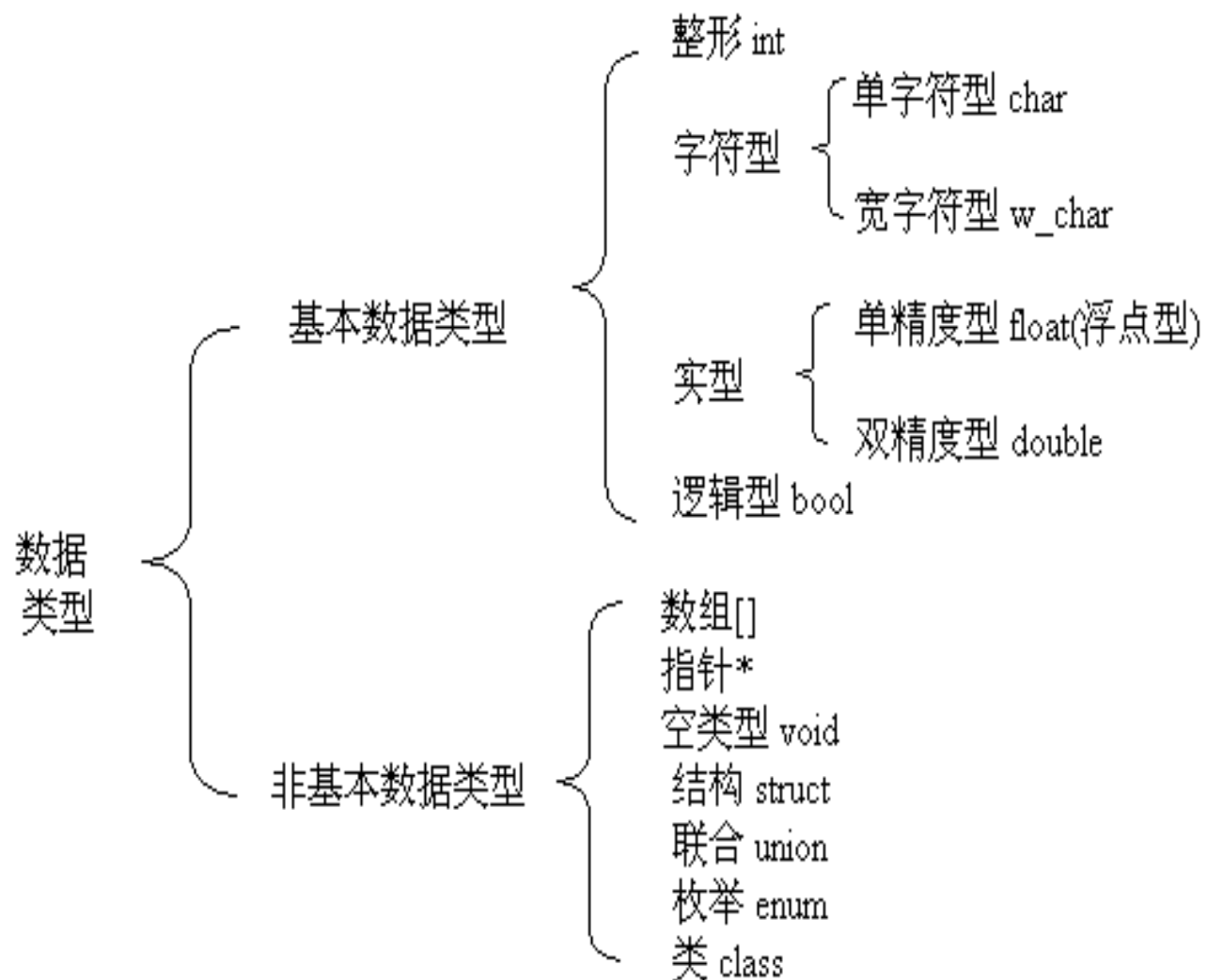
```
int i=5;  
double d=3.6;  
char c='A';  
scanf("%d,%lf,%c", &i,&d,&c);  
printf("%d,%lf,%c", i,d,c);
```

//C++输入输出

```
int i=5;  
double d=3.6;  
char c='A';  
cin>>i>>d>>c; //插入运算符  
cout<<i<<d<<c; //提取运算符
```



数据类型



变量



- 变量定义与C语言相同
举例：

```
int age;           //年龄变量  
float height, weight; //身高和体重
```

- 在C语言中，变量的声明只允许出现在程序块的开始；而在C++语言中变量声明可以出现在任何位置。



常量



- C++常量定义与C语言不同，C++中常量需要通过const关键字定义。相对来说，const更加现代，它是C++的一个关键字，其一般形式为

```
const 标识符=常量;
```

- 常量只能读不能修改，并且定义时必须初始化。

```
const double PI=3.1415926;
```

- 推荐用const，而不是#define预处理指令

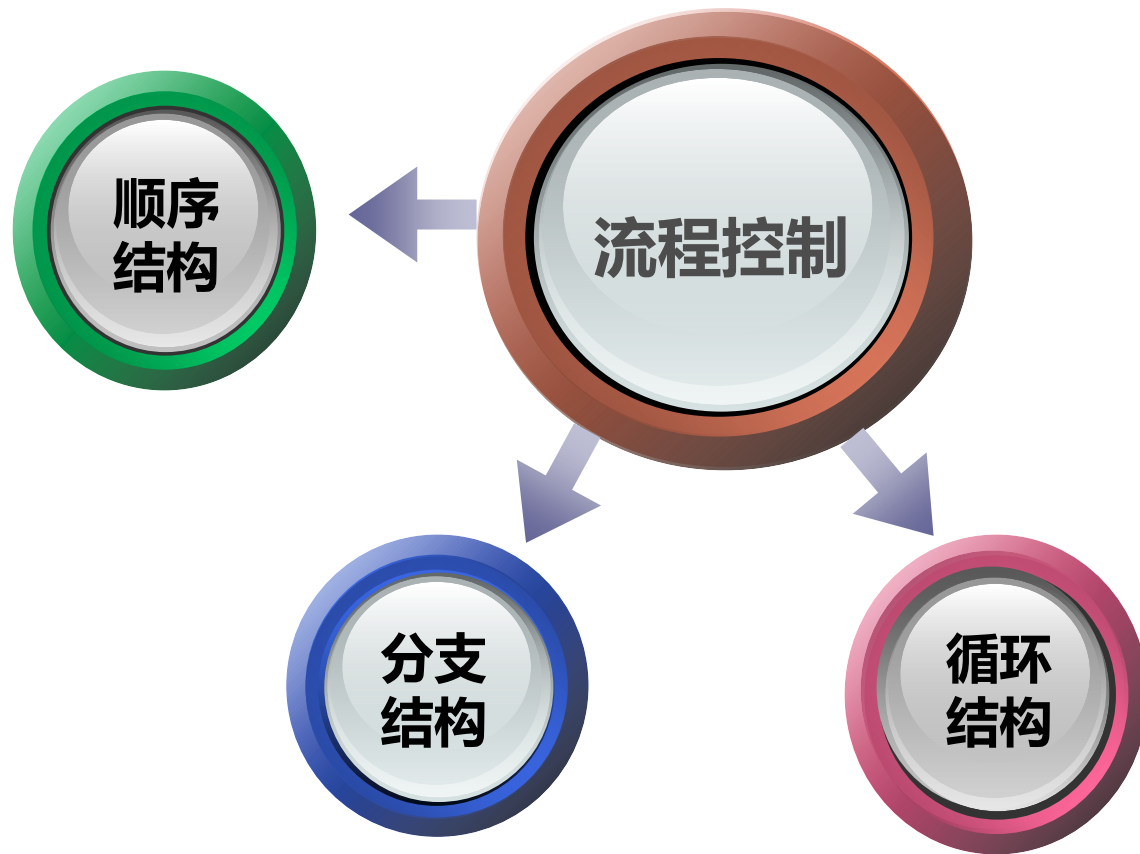


常量定义的举例



```
#include <iostream>
using namespace std;
const double PI=3.1415926;
double area(int radius){
    return PI*radius*radius;
}
int main(){
    cout<< area(2)<<endl;
    return 0;
}
```







- 字符集：C++程序区别大小写字母
- 标识符：C++不限制长度，但只有前31个字符有效
- 运算符：C++更加丰富，按功能可分为
 - 算数运算符
 - 关系运算符
 - 逻辑运算符
 - 其他

