



# 基于 Linux 的 C++

## 第一讲 C/C++ 基本语法元素

# ■ 提 纲

程序设计的基本概念

简单 C/C++ 程序介绍

程序设计流程与常用命令

数据类型

量与表达式

语 句

基本输入输出

程序设计风格

# ■ 程序设计的基本概念

## 程 序

一系列遵循一定规则并能正确完成特定功能的代码或指令序列

通常包括数据结构与算法两部分

# ■ 程序设计的基本概念

## 程序设计与程序设计语言

**程序设计：**按照任务需要，设计数据结构与算法，编写代码并测试其正确性，得到正确运行结果的过程

**程序设计方法学：**程序设计应遵循一定的方法与原则，养成良好的编程风格

**程序设计语言：**编写程序代码的规范，具有特定的语法规则、意义与使用环境

# ■ 程序设计的基本概念

## 算 法

问题的求解方法与步骤

不允许存在二义性

算法设计过程是逐步求精的

常使用流程图描述算法



# ■ 程序设计的基本概念

## 数据与数据结构

**数据：程序操作的对象**

**数据结构：数据对象之间的相互关系及构造方法**

**与算法关系密切，良好的数据结构可使算法更简单，  
恰当的算法可使数据结构更易理解**

# ■ 程序设计的基本概念

## 结构化程序设计

数据结构、程序流程控制、函数与算法、程序组织

## 面向对象程序设计

对象：行为与属性的集合

类与对象、对象构造、继承与派生

## 范型编程

标准模板库：迭代器、容器、抽象算法

# 简单 C 程序介绍

## 编写程序，显示字符串 "Hello World! "

```
#include <stdio.h>
int main()
{
    printf( "Hello World!\n" );
    return 0;
}
```

预编译指令，包含标准头文件

主函数的函数头，全局唯一

函数体，花括号内为语句序列

函数结束执行，程序退出

标准库函数，在输出设备（一般为屏幕）上输出一行文本函数调用，需要了解该函数原型，即函数如何使用的细节信息；printf 函数原型位于"stdio.h"中，故需包含该头文件双引号内为字符串内容；'\n' 为转义字符，表示换行分号表示语句结束



# ■ 简单 C++ 程序介绍

## 编写程序，显示字符串 "Hello World!"

```
#include <iostream>
```

预编译指令，包含标准头文件

```
using namespace std;
```

使用标准名空间

```
int main()
```

主函数的函数头，全局唯一

```
{
```

```
    cout << "Hello World!" << endl;
```

函数体，花括号内为语句序列

```
    return 0;
```

函数结束执行，程序退出

```
}
```

标准输出对象，在输出设备（一般为屏幕）上输出一行文本；连续的双左尖括号表示向左侧输出对象提供右侧的信息；双引号内为字符串内容；‘\n’为转义字符，表示换行；分号表示语句结束

# ■ 简单 C++ 程序介绍

## 编写程序，计算用户输入的两个整数之和

```
#include <iostream>
using namespace std;
int main()
{
    int a, b, sum;
    cout << "a: ";
    cin >> a;
    cout << "b: ";
    cin >> b;
    sum = a + b;
    cout << a << " + " << b << " = " << sum << endl;
    return 0;
}
```

## ■ 数据的输入与存储

**a**



**b**



**sum**



## ■ 数据的输入与存储

**a**

1

**b**

**sum**

## ■ 数据的输入与存储

**a**

1

**b**

2

**sum**



## ■ 数据的输入与存储

**a**

**1**

**b**

**2**

**sum**

**3**

## ■ C/C++ 程序结构特点

由一个或多个函数组成

具有惟一一个主函数 main

程序执行由主函数开始，其他函数由主函数调用或间接调用

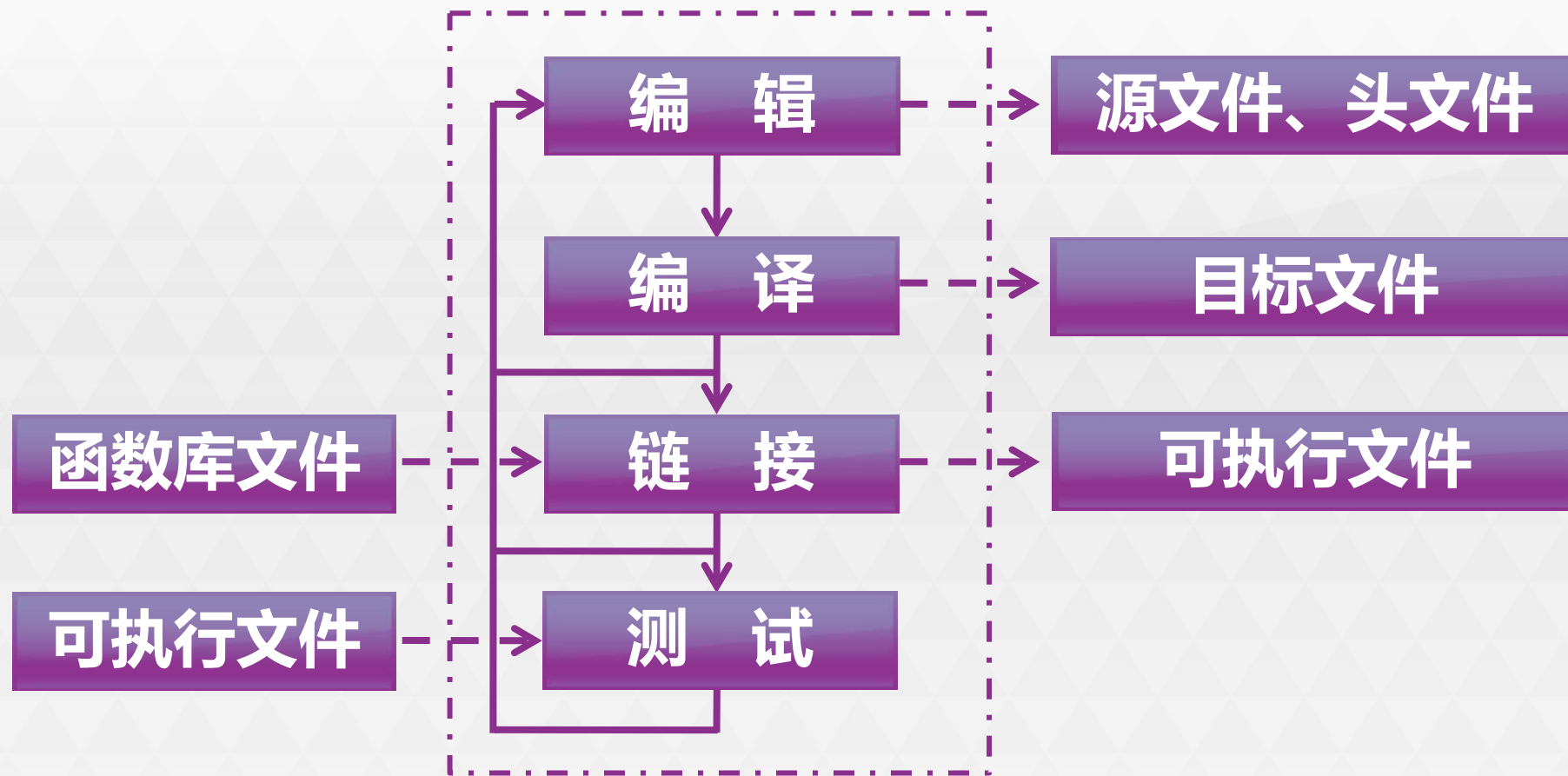
程序执行流程与函数定义顺序无关

每个函数包含函数头与函数体两部分

C/C++ 包含大量库函数和算法，可直接使用

库函数原型位于相应头文件中，使用函数前必须包含该头文件

# ■ 程序设计的基本流程



# ■ 编辑与编译

## 编 辑

行文本编辑器：vi 编辑器或其他编辑器

窗口文本编辑器：gedit 编辑器或类似编辑器

## 编 译

gcc：编译 C 程序

g++：编译 C++ 程序

示例：g++ main.cpp

## 执 行

缺省可执行文件：./a.out

# 数据类型

## 整数类型

使用 `int` 定义整数类型的数据对象

整数类型具有取值范围，仅能表示特定区间内整数值

## 浮点数类型

使用 `double` 或 `float` 表示小数

同样具有取值范围

首选 `double`，精度更高



# ■ 整数类型

## 整数类型

定义 a 为整数：`int a;`

整数的符号：确定是否表示负数

signed、unsigned

定义 b 为无符号整数：`unsigned int b;`

整数的长短：确定整数取值范围的大小

long、short

定义 c 为短整数：`short int c;`

# ■ 量与表达式

表达式

变 量

文字与常量

赋值与初始化

操作符与操作数

# ■ 表达式

## 表达式的定义

操作数与操作符序列，表达运算过程

操作数：文字、量、函数调用、括号

操作符：运算逻辑

表达式值：运算结果

表达式求值：计算表达式值的过程

类比：按普通四则运算理解表达式求值过程，逐一计算表达式中所有操作符，直至完成

# ■ 变 量

## 先定义再使用

定义格式：**变量类型** **变量名称** [, **变量名称**];

单独定义例：**int a;**

同时定义例：**double b, c;**

**变量名称应既有惟一性：不允许重名**

## **变量的四大基本特征（VANT）**

**变量总是具有值（value）**

**变量总是与内存中的地址（address）相对应**

**变量总是具有名称（name）**

**变量总是具有确定的类型（type）**

# ■ 文 字

**定义：程序中单独出现的值**

**例：3.14、100 等**

**整数类型文字**

**十进制、十六进制、八进制**

**浮点类型文字**

**字符串类型文字**



# 整数类型文字

十进制表示：普通整数表示

十六进制表示：逢十六进一，无正负号

以“0x”或“0X”开头的 0~9、A/a~F/f 序列

例：0x55、0X1F、0x2a

$$0x55_{16} = 85_{10}$$

八进制表示：逢八进一，无正负号

以“0”开头的 0~7 序列

例：055

$$055_8 = 45_{10}$$

注意不要与十进制混淆

## ■ 浮点数类型文字

### 一般形式：小数表示法

由整数部分、小数点与小数部分组成

例：3.1416、-2.718281828

### 指数形式：科学计数法

由尾数部分、字符 'E' / 'e' 与指数部分组成

例：3.14e3 =  $3.14 \times 10^3$ 、-2.7E-2 =  $-2.7 \times 10^{-2}$

# 字符串类型文字

## 双引号引起来的字符序列

例：“Programming in C++”、“Hello World!\n”

双引号是字符串文字的界定符，不是字符串的内容

## 转义序列

字符串不仅可以包含可打印字符，也可以包含表示特殊活动的特殊字符

格式：使用反斜杠开始转义序列，后跟一个或多个字符

转义序列整体作为单个字符进行处理，样式固定

例：“\n”、“\\”

# ■ 预定义的转义序列

转义序列	功能与意义
<code>\a</code>	响铃
<code>\b</code>	退格
<code>\f</code>	换页
<code>\n</code>	换行
<code>\r</code>	返回到当前行首
<code>\t</code>	水平制表键
<code>\v</code>	垂直制表键
<code>\0</code>	ASCII 码 0
<code>\\</code>	反斜杠字符自身
<code>\'</code>	单引号字符（仅在字符常数中需要使用此方式）
<code>\"</code>	双引号字符（仅在字符串常数中需要使用此方式）
<code>\ddd</code>	“ddd” 表示八进制的该字符 ASCII 码值
<code>\xhh</code>	“hh” 表示十六进制的字符 ASCII 码值，x 为十六进制标志

# ■ 常 量

**常量的定义：程序执行期间其值不可改变的量**

格式：**const 数据类型 常量名称 = 初始值;**

例：**const int zero = 0;**

## 常量的意义

解决直接出现的文字无法解释其意义的问题

文字？魔数也！

## 常量与变量

常量不可改变值，其他与变量同

**定义常量时必须进行初始化**

常量初始值必须能够在编译期间计算出来



# ■ 赋值与初始化

## 赋值表达式

格式：变量名称 = 表达式

例：a = 0

## 赋值语句：赋值表达式加分号

格式：变量名称 = 表达式;

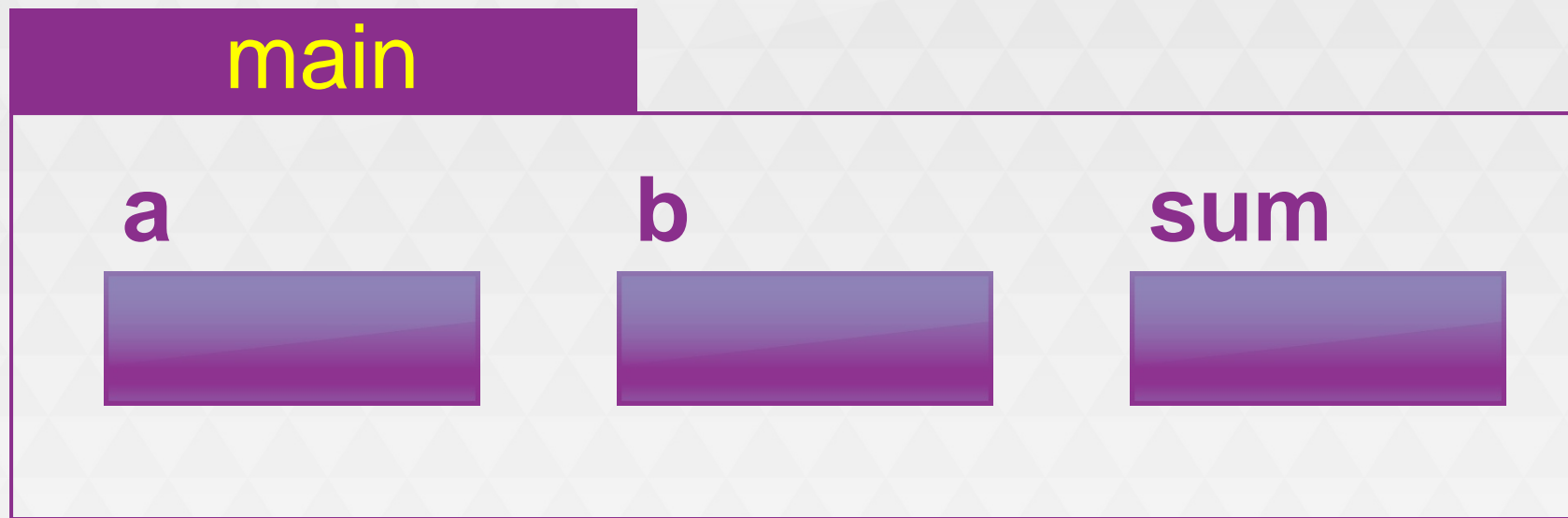
例：a = b + c;

## 变量的初始化

在定义变量时直接设定初始值

例：int a = 0;

# ■ 赋值流程



## ■ 赋值流程

**a = 1;**

**main**

**a**

**1**

**b**

**sum**

## ■ 赋值流程

**a = 1;**

**b = 2;**

**main**

**a**

**1**

**b**

**2**

**sum**

## ■ 赋值流程

**a = 1;**

**b = 2;**

**sum = a+b;**

**main**

**a**

**1**

**b**

**2**

**sum**

**3**

# ■ 操作符与操作数

## 操作符分类

一元操作符：只有单个操作数，例如负号

二元操作数：带有两个操作数，例如加减乘除等

三元操作数：带有三个操作数，例如条件表达式

## 操作符的优先级与结合性

优先级确定计算顺序，结合性确定计算方向



# ■ 语 句

## 简单语句

格式：表达式后跟分号

例一：`sum = n + d;`

例二：`cout << "Hello, World!" << endl;`

## 复合语句

格式：花括号对括起来的语句序列

例：`{ a = 1; b = a + 1; }`

## 空语句

格式：单独出现的分号

目的：满足程序特定语法规则的要求；作为未来添加程序代码的占位标记

# 基本输入输出

## 输出：cout

输出目的地：cout

输出（插入）操作符：<<

输出数据对象

示例：`cout << "Hello, World!" << endl;`

## 输入：cin

输入来源：cin

输入（提取）操作符：>>

输入数据对象

示例：`cin >> a >> b;`

# ■ 程序设计风格

**注 释**

**命名规范**

**宏与常量**

**赋值语句的简写形式**

**源代码的排版**

# ■ 注 释

注释的目的：增强程序的可读性、可维护性

## 注释的格式

格式一：`“/*”` 与 `“*/”` 对，注释其中的部分，可出现在程序代码的任意地方

例一：`int /* 此处内容为注释 */ a;`

格式二：`“//”`，注释从此字符序列开始直至本文本行结束

例二：`int a; // 此处内容为注释`

## 特别说明

同一种注释标记不可以嵌套

`“//”` 不跨行，`“/*”` 与 `“*/”` 对可跨行

# ■ 命名规范

## 字符集：ASCII字符集

大小写英文字母、数字、部分特殊符号

## 标识符

以下划线或字母开头，由下划线、字母、数字组成，表示常量、变量、函数、类型等名称，**区分大小写**

**关键字：**系统专用的具有特定意义的标识符

**预定义标识符：**编译命令与库函数名，具有特定意义，一般不能重新定义

**用户定义标识符：**用户根据需要定义，主要是为了使程序容易阅读、理解与维护

# ■ 宏与常量

## 宏定义

#define 指令：预处理命令

例：**#define PI 3.14**

一旦定义，可以代替常量使用

宏定义不是常量

程序中应优先使用常量而不是宏

## 目的与意义

增强程序可读性

使用有意义的名字命名

增强程序的可维护性

将值定义为符号常量，则程序中仅需要修改一次



## ■ 赋值语句的简写形式

加赋： $x += a$  等价于  $x = x + a$

减赋： $x -= a$  等价于  $x = x - a$

乘赋： $x *= a$  等价于  $x = x * a$

除赋： $x /= a$  等价于  $x = x / a$

除法作用于整数上时，结果也为整数

余赋： $x \% = a$  等价于  $x = x \% a$

%：两个整数进行整数除法，结果为余数

### 特别说明

$x *= a + b$  等价于  $x = x * (a + b)$

# ■ 源代码排版

递进层次应使用左缩进格式

每行代码不能过长，不超过80个字符

函数代码不超过60行

使用空行区分不同功能代码

复合语句书写格式要统一

除非特别必要，否则不要在一行上书写多条语句

命名规范要一致

无论采用什么标准，都一定要一直按照该标准执行

# 编程实践

1.1 编写程序，接受用户输入的10个整数，输出它们的和。

1.2 编制程序完成下述任务。接受两个数，一个为用户一年期定期存款金额，一个为按照百分比格式表示的一年期定期存款年利率。程序计算一年期满后本金与利息总额。

说明：（1）存款金额以人民币元为单位，精确到分；

（2）输入利率时不需要输入百分号，例如一年期定期存款年利率为2.52%，用户输入2.52即可；

（3）输出数据如何精确到分？