C++ 编程讲义 — 初识C++

工欲善其事，必先利其器。

编程本质上是我们通过计算机向机器下达精确指令的过程。因此，在正式编写第一行代码之前，我们必须先熟练掌握并使用最核心的工具——电脑的基本操作与 Windows 操作系统。对于初次接触计算机的同学，我们需要从最基础的物理设备开始学习。

第一部分：计算机基础设备操作

一、鼠标的基本操作

鼠标是我们在图形界面中向计算机发送位置与选择指令的设备。

- **左键单击**：用食指按一下鼠标左键并立即松开。用于选定目标（例如选中一段代码或定位光标）。
- **左键双击**：快速、连续地按两次鼠标左键。通常用于打开软件或选中一整个单词。
- **右键单击**：用中指按一下鼠标右键。用于唤出快捷菜单，可以在其中找到复制、粘贴等功能。
- **拖拽**：按住左键不放，移动鼠标到目标位置后再松开。用于移动窗口或选中多行文本。

二、熟悉键盘按键

键盘是编写代码时最重要的输入设备。标准键盘一般包括26个英文字母键、10个数字键、12个功能键（F1 ~ F12）、方向键以及其他的一些功能键。

所有按键在物理布局上主要分为5个区：

1. **主键盘区**：最大的一块区域，包含字母、数字和常用标点符号，是我们书写代码的核心区域。
2. **功能键区（F1~F12）**：位于键盘最上方，在不同的软件中有不同的快捷功能。
3. **编辑/控制键区**：包含方向键、删除键（Delete）等，用于在代码中移动光标和修改文本。
4. **小键盘区**：位于最右侧，主要用于快速输入纯数字。
5. **键盘指示灯**：提示当前键盘的状态（如大小写锁定、数字键盘锁定等）。

编程必须掌握的按键指令：

- **Shift 键（上档键）**：用于切换中英文标点符号，或输入按键上半部分的符号（例如按住 Shift 再按数字 9，可以打出左括号 `(` ）。
- **Enter 键（回车键）**：在代码中用于换行，使指令结构清晰。
- **Backspace 键（退格键）**：用于删除光标前面的字符，是修改代码错误的常用键。

第二部分：什么是程序与计算机语言?

大家目前可能对软件中复杂的程序代码感到难以理解，但是，大家了解人类生活中的程序吗？有什么程序是大家在日常生活中经历过的呢?

例如：开学典礼的程序
主持人会宣布，今天典礼的程序如下：

- 第一项：升国旗；

- 第二项：领导介绍；
- 第三项：校长致辞；
- 第四项：教师代表发言；
- 第五项：学生代表发言。

这其实就是人类社会的程序。映射到计算机科学中，**计算机的程序被定义为：为了得到某种结果，通过计算机语言表达的指令序列。** 计算机通过严格执行你下达的指令，最终得出预期的结果。

假如我们要编写一个让计算机"把大象塞进冰箱"的程序，相当于依次下达以下指令：

1. 打开冰箱门；
2. 把大象塞进冰箱；
3. 关上冰箱门。

计算机语言的演变与分类

为了把上述指令传达给计算机，我们需要使用计算机语言。计算机语言主要分为三种：

1. **机器语言（二进制语言）**：由 `0` 和 `1` 组成，这是计算机底层唯一能直接听懂的语言，但人类极难阅读和编写。
2. **汇编语言**：使用一些简短的英文字母代替二进制代码，比机器语言稍易读，但依然非常晦涩。
3. **高级语言（接近英语/自然语言）**：使用人类容易理解的英文字母、单词和数学符号编写。

我们即将学习的 **C++ 课程就属于高级语言。**它的本质是由一套特定的英文单词和符号组成的指令集合。然而，计算机底层只能听懂二进制，这就意味着，**必须要将高级语言"翻译"成机器语言，计算机才能执行。**

第三部分：C++ 开发工具 (Dev-C++) 与执行过程

那我们应该在什么地方、如何去书写这些高级语言呢？这就需要用到专业的集成开发环境（IDE）。

一、初识 Dev-C++

对于初学者，我们使用 **Dev-C++。**这是一款专门用于书写和运行 C/C++ 高级语言的工具。它提供了一个干净的文本编辑区，并且内置了代码高亮等便捷功能，极大地规范和方便了我们的书写。

二、程序的编译（翻译）与报错机制

当我们在 Dev-C++ 中书写好高级语言代码后，此时代码还只是一堆英文字符，系统在等待着"翻译官"进行翻译。
这个"翻译"的过程，在计算机科学中被称为"编译 "。

- **严格的语法**：翻译官只能翻译它看得懂的、符合严格规则的代码。如果代码存在拼写错误，或者不符合语法要求，翻译官就会看不懂并拒绝工作。
- **编译报错（出现红色的 ×）**：如果下方窗口出现红色的 `×` 或错误提示，说明代码有错误，翻译失败。**只有在代码完全无误，翻译官顺利翻译后，程序才能被执行。**
- **标点符号规范**：因为 C++ 是一门基于英文环境诞生的语言，它的指令**无法识别中文标点符号。**如果误用了中文的逗号（，）或分号（；），就等同于你自创了一个不存在的单词，翻译官会直接报错。编写代码时，必须确保输入法处于**英文半角**状态。

三、编译并运行（快捷键 F11）

当确保代码书写完毕且无误后，我们需要让计算机去执行命令。
在 Dev-C++ 中，**按下键盘上的 `F11` 键，即可一键开始“编译并运行”程序。** 软件会自动先将代码翻译为机器语言，紧接着让计算机执行它。

四、控制台（黑框框）

按下 `F11` 并且编译成功后，屏幕上会弹出一个黑色的独立窗口。这个窗口被称为**控制台。**
控制台的作用是**展示程序的最终运行结果**，或者在程序需要用户输入数据时，**与程序进行交互。**当控制台弹出并显示出预期的结果时，就意味着你的指令被计算机成功执行了。

第四部分：初识 C++ —— 第一个程序 Hello World

了解了工具后，我们将正式开始编写全人类学习编程时的第一个经典程序：向世界问好。

一、代码示例

在 Dev-C++ 的代码编辑区中，输入以下代码：

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, World!" << endl;
    return 0;
}
```

二、代码解析

这段看似复杂的英文，其实每一行都有其含义：

- `#include <iostream>` **—— 引入输入输出流库**
  - **含义**：告诉计算机“我需要使用标准的输入和输出工具”。如果没有这一句指令，计算机就无法理解后续代码中用于在屏幕上显示文字的命令。
  - 就像做披萨前，要先准备好烤箱和工具，没有烤箱肯定无法制作披萨。
  - 如果没有这一句，电脑就不知道什么是 `cout`，没有用来输出文字的工具，自然就没法输出文字。

- `using namespace std;` **—— 使用标准命名空间**
  - **含义**：告诉计算机“本程序默认使用名为 `std` 的标准工具库”。
  - 比如：A 工具箱和 B 工具箱都有个锤子，你就要先说清楚是哪个工具箱的，不然计算机就会搞混。
  - `std` 就像 工具箱，`cout` 就像锤子。
  - 写了这一句，相当于告诉电脑：“我默认要锤子的时候都用的是 `std` 这个工具箱。”，我们以后要用的所有的工具基本都是在`std`这个工具箱里的。

- `int main() {` —— **主程序入口**
  - **含义**：这是所有 C++ 程序的起点。程序执行时，必定从 `main` 函数后方的大括号 `{` 开始，到对应的大括号 `}` 结束。一个完整的 C++ 程序必须有且仅有一个 `main` 。`int` 是单词 **integer** 的缩写，意思是“整数”，规定这个函数最终执行完毕后，必须向计算机汇报一个整数作为结果。
  - 好比一个“主任务清单”，电脑会从这里开始做事情。

- `cout << "Hello, World!";` —— **屏幕输出指令**
  - `cout`：用于输出数据的工具，相当于一个“喇叭”，把数据告诉计算机让它来输出。
  - `<<`：像传送带，把右边的东西送到左边。
  - `"Hello, World!"`：被英文双引号包裹的内容，即我们要原样显示在屏幕上的具体文字数据。
  - **整句话的意思**：把 "Hello, World!" 交给输出系统，让它把这句话输出在控制台上。
- *注意：这一行的末尾必须以英文分号 `;` 结束，C++ 里 `;` 代表一条指令的执行完毕。*
- `return 0;` —— **正常结束程序**
  - **含义**：返回一个数字 `0` 给系统。`0` 表示一切正常。
  - 好比告诉老师：“作业完成了，没有出错。”
- `}` —— **程序结束**
  - **含义**：与前文的 `{` 对应，标志着整个 `main` 函数的指令序列到此彻底结束。

以下是这段代码中涉及的英文单词与缩写的中文翻译对照表（理解英语含义后，对于代码的理解也会非常有帮助）：

第一个C++程序 基础词汇翻译表（务必记忆！）

代码单词	英文全拼	中文含义
include	include	包含、引入
iostream	input output stream	输入输出数据流
using	using	使用
namespace	name space	命名空间
std	standard	标准
int	integer	整数
main	main	主要的（程序的起点）
cout	console output	控制台输出
return	return	返回

第五部分：程序员的注释

一、什么是注释?

注释是专门写给人看的说明文字，计算机的翻译官（编译器）在将代码翻译成机器语言时，会**完全跳过并忽略**这些内容。

可以将注释理解为我们在课本空白处书写的学习笔记：老师（代表计算机）在批改作业时不会检查这些笔记，但这些文字能极大地帮助你自己或他人理清程序的逻辑与思路。

二、注释的两种标准写法

在 C++ 语言中，为了满足不同的说明需求，提供了两种添加注释的语法规范：

1. 单行注释 (`//`)

- **语法规则**：在代码中使用两个连续的斜杠 `//`。从这两个斜杠开始，直到本行结束的所有文字，都会被编译器自动忽略。通常用于对某单一行的代码进行简短解释。
- **代码示例**：

```
cout << "Hello"; // 这是一句单行注释，专门用于解释左侧这行代码的作用
```

2. 多行注释 (`/* ... */`)

- **语法规则**：从 `/*` 符号作为起点开始，直到 `*/` 符号作为终点结束。这两个符号中间包裹的所有文字内容（无论中间跨越了多少行），都会被系统统一认定为注释。此结构非常适合在程序开头书写长篇的整体功能说明。
- **代码示例**：

```
/* 这是多行注释块
   此处可以详细书写第一点、第二点...
   计算机在执行时会完全无视这部分内容
*/
cout << "World";
```

(注：在 Dev-C++ 等专业软件中，注释部分通常会被自动标记为绿色或其他浅色，以便与正常执行的代码做出明显的视觉区分。)

三、为什么在软件开发中要书写注释?

初学者往往为了节省时间而跳过书写注释的步骤，但书写注释是非常重要的，主要是因为：

- **防遗忘**：人类的短期记忆是有限的。经过数周或数月后再次阅读自己编写的代码时，如果没有注释辅助回忆，当时设计的复杂计算逻辑往往难以迅速被重新理解。
- **团队协作**：在真实的科技企业中，软件开发是一个团队项目，你的代码必定需要交接给其他工程师查阅或修改。清晰、规范的注释是保证团队成员之间能够快速读懂彼此逻辑。