第二章 C++程序设计[1]

一、程序基本概念

【大纲内容】

【难度 1】标识符、关键字、常量、变量、字符串、表达式的概念

【难度 1】常量与变量的命名、定义及作用

【难度 2】头文件与命名空间的概念

【难度 2】编辑、编译、解释、调试的概念

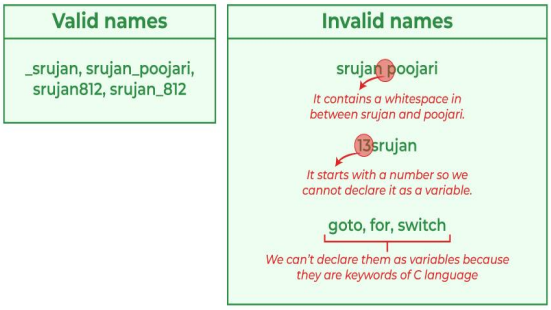
（一）、程序的构成元素（像拼图一样组合成一个完整的程序）

1、标识符（Identifier）

标识符是给程序里的“东西”**起的名字**，比如变量、函数、数组的名字。

就像你给自己的宠物取名字：“雪球”、“旺财”，在程序里也要给“数字”、“文字”等数据取名，这样电脑才能记住、使用它们。

命名规则



- 只能由**英文字母、数字、下划线**组成。
- 不能以数字开头。
- 不能和关键字一样。

☑C++例子：

```
int age;          // age 是标识符，表示“年龄”
double height; // height 是标识符，表示“身高”
```

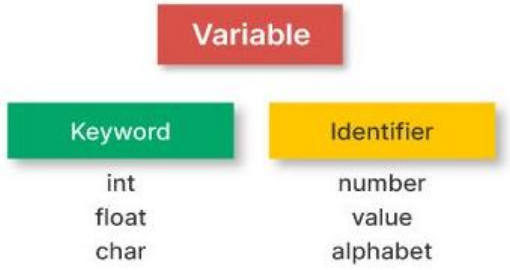
2、关键字（Keyword）

关键字是 C++ 语言里预先**保留**的特殊词语，它们有特别的意义，**不能用作变量名**。

就像“警察”、“校长”这些词不能随便用作人名一样，关键字是电脑规定“专用”的词。

常见关键字：

- int（整数）
- double（小数）
- return（返回）
- if（如果）
- while（当……时）



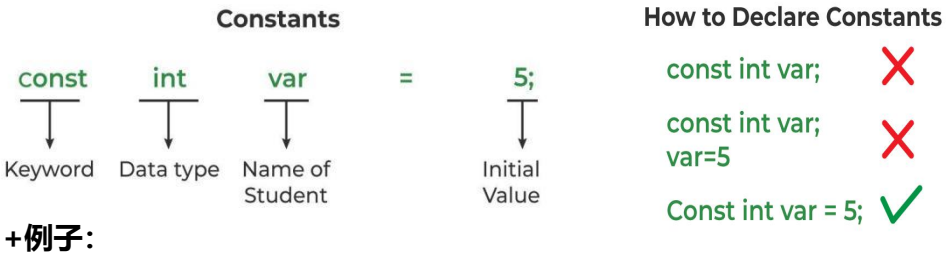
❌ 错误示例：

```
int int = 5; // 错误，int 是关键字，不能当变量名
```

3、常量（Constant）

常量是一个**固定不变的值**，在程序运行过程中不能改变。

比如“1 年有 12 个月”、“π是 3.14”，这些数永远不会变，就是常量。



☑ C++例子：

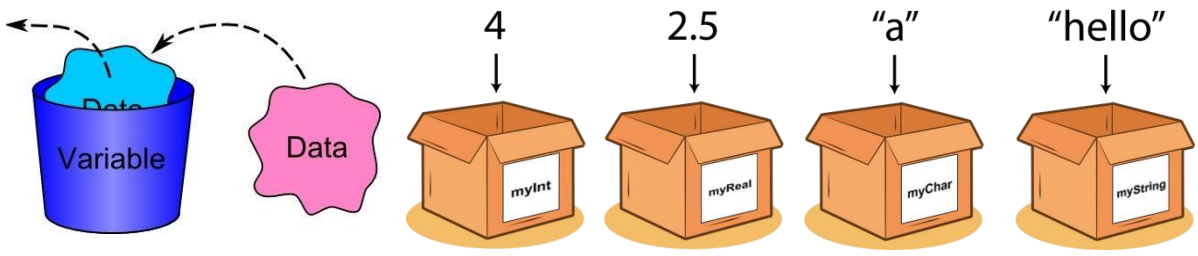
```
const double PI = 3.14159; // 常量，不能改变
```

- 使用 **const** 关键字来定义常量。

4、变量（Variable）

变量是用来**存储数据**的盒子，里面的值是**可以改变**的。

就像你的书包今天装铅笔，明天可以换成橡皮，这个“书包”就是变量。



☑ C++例子：

```
int age = 10;
age = 11; // 变量 age 变成了 11
```

5、字符串（String）

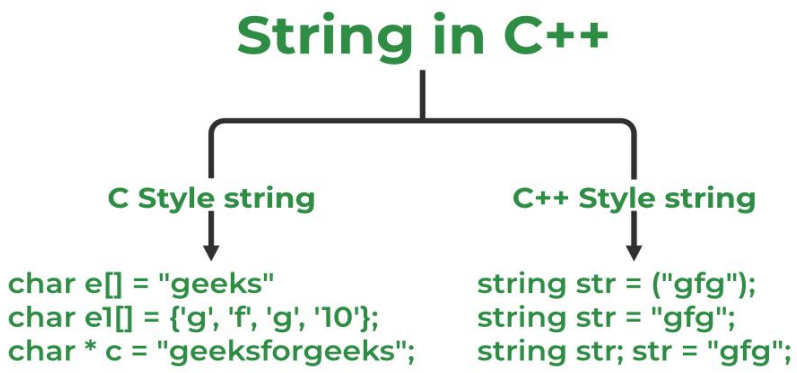
字符串是由一串字符组成的内容，通常用**英文双引号 " "** 包起来。

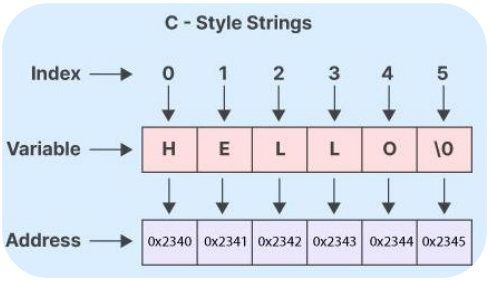
你在打字输入“原神启动”，这整句话就是一个字符串。

☑ C++例子：

```
string name = "Alice";
cout << "你好，" << name << "！" << endl;
```

- 注意：字符串要 #include <string>，并使用 string 类型。

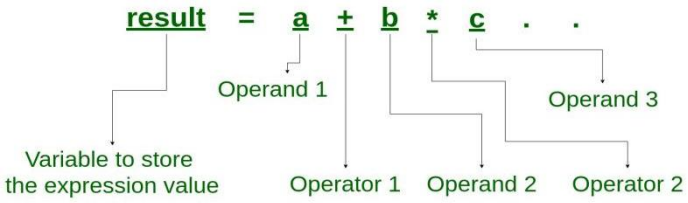




6、表达式（Expression）

表达式是由变量、常量和运算符组成的**可以计算出一个结果的式子**。

比如“3 + 4”是个式子，它等于 7，就是一个表达式。



☑ **C++例子：**

```
int a = 5;
int b = 3;
int sum = a + b; // a + b 是表达式，结果是 8
```

（二）常量与变量的命名、定义及作用

1、命名规则（常量和变量都适用）

项目	要点说明
只能用**英文字母、数字、下划线**	如：student_name, score1
不能以数字开头	如：1age 错误
不能使用关键字	如：int, if, for 不可用
最好有意义	用 age 表示年龄，而不是 a1

2、定义格式（C++标准写法）

☑ **变量定义：**

```
int age = 10;         // 整数变量
double height = 1.5; // 小数变量
char grade = 'A';      // 字符变量
```

☑ **常量定义：**

```
const double PI = 3.14159;
const int DAYS_IN_WEEK = 7;
```

- 使用 **const** 开头表示这个变量是“常量”，不能再修改！

3、作用是什么?

类型	用处
变量	存储“会变化”的数据（输入、计算中间值）
常量	存储“固定不变”的数（规则、公式）

☑ **举个例子说明变量和常量的区别：**

```
const int DAYS = 7;     // 一周固定 7 天（常量）
int homework = 5;       // 今天布置了 5 个作业（变量）
homework = 3;           // 明天布置的作业变了（变量值可以改）
// DAYS = 6;            // ❌ 错误，常量不能改
```

（三）头文件（Header File）与命名空间（Namespace）

1、头文件（Header File）

头文件（Header File）是在 **C++ 程序中提前写好的一些功能代码集合**，我们可以通过 #include 把它们“借”过来使用。

它的作用是：提供别人已经写好的函数、类、常量等内容，让**我们不用从头写起**。

就像你搭积木，如果有现成的轮子、窗户模块，你只要把它拼上去就好了，不用自己刻轮子了。头文件就是这些“**现成的功能模块**”。

☑ **C++中常见头文件：**

头文件	包含内容说明	常用函数/类示例
<iostream>	输入输出流	cin, cout, endl
<cstdio>	C 风格输入输出	scanf, printf, getchar
<cmath>	数学函数	sqrt, pow, abs, fabs, ceil, floor
<cctype>	字符处理函数	isalpha, isdigit, tolower, toupper
<cstring>	C 风格字符串处理	strlen, strcmp, strcpy, memset
<algorithm>	标准算法库	sort, max, min, swap, find
<vector>	动态数组（向量）	vector, push_back, pop_back, size
<queue>	队列和优先队列	queue, priority_queue, push, pop,
<stack>	栈	stack, push, pop, top
<set>	集合和多重集合	set, multiset, insert, find, count
<map>	映射和多重映射	map, multimap, insert, find,
<utility>	工具类，如 pair	pair, make_pair
<string>	字符串类	string, substr, find, length, append
<climits>	整数类型的极限值	INT_MAX, INT_MIN, LONG_MAX
<cstdlib>	通用工具，包括动态内存管理、随机数等	malloc, free, rand, srand, exit
<bits/stdc++.h>	**万能头文件，包含上述大部分头文件**	

示例：

```
#include <iostream> // 引入输入输出功能
#include <cmath>    // 引入数学计算功能
```

2、命名空间（Namespace）

命名空间（Namespace）是用来**区分不同变量或函数的“班级”**，防止重名冲突。

在 C++里，很多变量/函数可能重名，命名空间就像是给它们加姓氏，分清楚谁是谁家的。

就像两个班级都有叫“小明”的同学，一个是 1 班小明，一个是 2 班小明，为了不混淆，就说“1 班的小明”，“2 班的小明”，这就是命名空间的作用！

☑最常用的命名空间：std

- **C++标准库的所有功能**都放在 **std** 这个“班级”里。
- 比如 cout 实际是 std::cout，意思是“标准库里的 cout”。

示例：

```
#include <iostream>
int main() {
    std::cout << "你好，命名空间！" << std::endl;
    return 0;
}
```

小技巧：你也可以加一句：

```
using namespace std;
```

这样就可以省略 std:: 前缀啦！

(四) 编辑 (Edit) 、编译 (Compile) 、解释 (Interpret) 、调试 (Debug)

1、编辑（Edit）

编辑是指我们在电脑上**写程序**代码的过程，通常是在代码编辑器里完成的。

就像你在写作文、画画，编辑就是“写代码作文”的阶段。

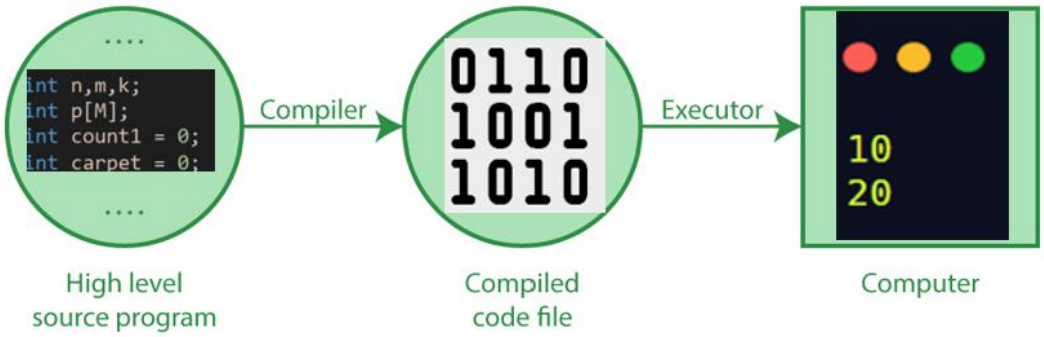
2、编译（Compile）

编译是将我们写的 **C++源代码（.）**转换成机器能读的 **“0 和 1”**代码的过程。

编译器（Compiler）就像翻译官，把我们写的“人话”代码翻译给电脑听。

你写了中文说明书（代码），电脑只懂二进制，编译器帮你把它翻译成“电脑语”。

编译后的文件叫**“可执行文件”**。



3、解释（Interpret）

解释是指程序**边运行边翻译**。不像编译要全部翻译后再运行，解释是一边读一边翻译一边做。

C++ 是编译型语言，但解释型语言比如 Python 就是靠“解释器”运行的。

解释就像是老师读书给你听，一边读一边解释意思；而编译是老师先翻译整本书再让你自己读。

4、调试（Debug）

调试就是**发现和改正程序中的错误（bug）**过程。

程序出错时，我们要像小侦探一样查找出错的地方，然后修好它！

你写数学作业的时候错了一道题，检查发现哪里错，改对它，这个过程就叫“调试”。

☑ C++中常见的错误类型：

类型	举例	含义
语法错误	忘写分号；拼错关键字	程序根本无法翻译运行
逻辑错误	算错了（比如乘错写成加）	程序能运行但结果不对
运行时错误	除以 0、数组越界等	运行时才发现的问题