C++入门

第一个 C++程序：

```
#include <iostream>

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

观察这个程序：

头文件使用的是#include <iostream>

C++标准库中的头文件不需要.h 作为后缀

C 中的头文件能否使用？

仍然可以，以 stdio.h 为例，可以#include <stdio.h>或者#include <cstdio>

用户自定义头文件如何使用？

使用#include “XXX.h”

如何运行程序：

如何编译程序？

g++ -o main.o –c main.cpp

如何链接程序？

g++ -o main.exe main.o

命名空间：

在这个程序中，使用的是 std::cout 而不是 cout，原因在于 cout 位于标准（std）名称空间中。

那么什么是名称空间呢？

假设调用 cout 时没有使用名称空间限定符，且编译器知道 cout 存在于两个地方，编译器应该调用哪个呢？当然，这回导致冲突，从而导致编译失败。这就是命名空间的用武之地。名称空间是给代码指定的名称，有助于降低命名冲突的风险。通过使用 std::cout，可命令编译器调用名称空间 std 中独一无二的 cout。

如果在代码中频繁添加 std 限定符，会显得很繁琐。为避免添加该限定符，可以使用声明 using namespace std;

如下：

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

int main() {

    cout << "hello world" << endl;

    return 0;
}
```

使用 cin 和 cout 执行基本的输入输出操作：

要将简单的文本数据写入到控制台，可使用 std::cout，要从终端读取文本，可以使用 std::cin。

例如：

C++入门 郭春阳

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

int main() {

    int input_number;
    cout << "Please input a number: ";
    cin >> input_number;

    cout << "Enter your name: ";
    string input_name;    //字符串
    cin >> input_name;

    cout << input_name << " " << input_number << endl;
    return 0;
}
```

如何输入未知数目的元素：

```
#include <iostream>
int main()
{
    int sum = 0, value;
    while(std::cin >> value)
    {
    sum += value;
    }
    std::cout << "Sum is: " << sum
              << std::endl;
    return 0;
}
```

练习：

输入一个正整数 n，然后求从 1 到 n 的和，并打印输出。

数据类型：

C++入门 郭春阳

C++新增了 bool 类型来表示真假，C++支持的数据类型有：bool char int float double long 等。

练习：

使用 sizeof 打印每种数据类型的大小

左值和右值：

左值：可以放在赋值语句的左边或者右边

右值：只可以放在赋值语句的右边

练习：

举几个左值和右值的例子

使用 typedef 简化类型定义

typedef unsigned int UINT;

const 变量

看下面的代码：

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

int main() {

    const int a = 10;
    a = 33; // ERROR
}
```

引用类型：

引用是变量的别名。声明引用时，需要将其初始化为一个变量，

因此引用只是另一种访问相应变量存储数据的方式。

要声明引用，可使用引用运算符（&），如下面的语句所示：

int num = 99;

int &num_ref = num;

当然也可以用于其他类型，例如之前见过的字符串：

string s = "test";

string &s_ref = s;

看下面的程序，观察输出结果：

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

int main() {

    int a = 30;
    cout << "a = " << a << endl;
    cout << "a is at address: " << &a << endl; //打印地址

    int &ref = a;
    cout << "ref is at address: " << &ref << endl;

    int &ref2 = ref;
    cout << "ref2 is at address: " << &ref2 << endl;
    cout << "ref2 gets value, ref2 = " << ref2 << endl;
}
```

输出表明：无论将引用初始化为变量还是其他引用，它都指向相应变量所在的内存单元。因此，引用是真正的别名，即相应变量的另一个名字。