

第二章

第二章 开始：29

本章内容包括：

- 创建 C++ 程序。
- C++ 程序的一般格式。
- `#include` 编译指令。
- `main()` 函数。
- 使用 `cout` 对象进行输出。
- 在 C++ 程序中加入注释。
- 何时以及如何使用 `endl`。
- 声明和使用变量。
- 使用 `cin` 对象进行输入。
- 定义和使用简单函数。

c语言 面向过程 c++ 面向对象

泛型 独立于特定的数据类型

c++

main函数 花括号括起来

```
#include <iostream>
using namespace std;
int main(void) //不接受任何参数 里面是有参数的
{
    return 0; //函数主体语句
}
/* zhushi*/
```

int就是函数的类型 return 返回的是返回值；

```

#include <iostream>
using namespace std;
int main(void)//不接受任何参数 里面是有参数的
{
    cout<<"Come up and c++ me some time";
    //输入的就是流
    //换行的两种方式
    cout<<"come up and c++ me some time.\n";
    cout<<"come up and c++ me some time ."<<endl;
    cout<<" You won't regret it !"<<endl;
    return 0;//函数主体语句
}
/* zhushi*/

```

用了解对象的内部情况，就可以使用它。只需要知道它的接口，即如何使用它。`cout` 对象有一个简单的接口，如果 `string` 是一个字符串，则下面的代码将显示该字符串：

```
cout << string;
```

对于显示字符串而言，只需知道这些即可。然而，现在来看看 C++ 从概念上如何解释这个过程。从概念上看，输出是一个流，即从程序流出的一系列字符。`cout` 对象表示这种流，其属性是在 `iostream` 文件中定义的。`cout` 的对象属性包括一个插入运算符 (`<<`)，它可以将其右侧的信息插入到流中。请看下面的语句（注意结尾的分号）：

```
cout << "Come up and C++ me some time.";
```

它将字符串 “Come up and C++ me some time.” 插入到输出流中。因此，与其说程序显示了一条消息，不如说它将一个字符串插入到了输出流中。不知道为什么，后者听起来更好一点（参见图 2.2）。



```

#include <iostream> /* 实例三 引入变量*/
int main(void) {
    using namespace std;
    int acrots; //声明变量carrots
    carrots = 25; //赋值给carrots
    cout<<"I have ";
    cout<<carrots<<endl; //这个是打印变量carrots
    cout<<" carrots "<<endl; //这个是打印字符串carrots
    carrots=carrots -1; //修改变量
    cout<<"Crunch,Crunch,Now I have "<<carrots<<" carrots "<<endl;
    return 0;
} //存储的时候是按照二进制存储的 打印出来的时候会按照你定义的函数输出出来

```

捕获用户输入

```
#include <iostream>
int main(void) {
    using namespace std;
    int acrots;
    cout<<"How many carrots do you have ?"<<endl;
    cin>>carrots;
    cout<<"Here are two more."<<endl;
    carrots=carrots+2;
    cout<<"Now you have " <<carrots<<"carrots"<<endl;
    return 0;
}
```

类

类里面什么信息和操作数据的方法 是用户自己定义的数据类型 对象是具体实例化

比如类如果是红富士苹果 那么对象就是指的某一个红富士苹果

cin cout 就是iostream类里面的两个实例化的对象

函数

可以分为有返回值 和 有返回值的 的

有返回值的 比如sqrt () 平方 x=sqrt (6.25) ;

```
#include <iostream>
#include <cmath>
int main(void)
{
    using namespace std;
    double area;
    double side;
    cout<<"Enter the floor area ,in square feet,of your home;";
    cin>>area;
    side = sqrt(area);
    cout<<"That's the equivalent of a square"<<side<<"feet"<<endl;
    return 0;
}
```

案列

```
#include <iostream>
void simon(int n); //使用之前必须先要声明
int main(void)
{
    using namespace std;
    simon(3)
    return 0;
}

void simon(int n) {
    using namespace std;
    cout<<"Simon says touch yur toes "<<n<<"tomes"<<endl;
} //这个是固定次数
```

```
#include <iostream>
void simon(int n); //使用之前必须先声明
int main(void)
{
    using namespace std;
    int count;
    cout<<"Please enter your number"<<endl;
    cin>>count;
    simon(count);
    cout<<"Done!!"<<endl;
    return 0;
}

void simon(int n) {
    using namespace std;
    cout<<"Simon says touch yur toes "<<n<<"tomes"<<endl;
}
```

- 进制转换

```

#include <iostream>
int stonetolb(double n);
int main(void) {
using namespace std;
cout<<"Enter the weight in stone :"<<endl;
int stone;
cin>>stone;
int pounds;
pounds = stonetolb(stone);
cout << stone <<"stone =";
cout << pounds <<"pound"<<endl;
return 0;
}
int stonetolb(int sts) {
return 14*sts;//case 1
/**
int pounds=14*sts;
return pounds;
**/
}

```

using namespace std;或者这样

```

#include <iostream>
using namespace std;
void simon(int n);//使用之前必须先要声明
int main(void)
{
//using namespace std;
simon(3)
return 0;

}
void simon(int n) {
//using namespace std;
cout<<"Simon says touch yur toes "<<n<<"tomes"<<endl;
}//这个是固定次数

```

总结

C++程序由一个或多个被称为函数的模块组成。程序从 `main()` 函数（全部小写）开始执行，因此该函数必不可少。函数由函数头和函数体组成。函数头指出函数的返回值（如果有的话）的类型和函数期望通过参数传递给它的信息的类型。函数体由一系列位于花括号（`{}`）中的 C++ 语句组成。

有多种类型的 C++ 语句，包括下述 6 种。

- 声明语句：定义函数中使用的变量的名称和类型。
- 赋值语句：使用赋值运算符（`=`）给变量赋值。
- 消息语句：将消息发送给对象，激发某种行动。
- 函数调用：执行函数。被调用的函数执行完毕后，程序返回到函数调用语句后面的语句。
- 函数原型：声明函数的返回类型、函数接受的参数数量和类型。
- 返回语句：将一个值从被调用的函数那里返回到调用函数中。

类是用户定义的数据类型规范，它详细描述了如何表示信息以及可对数据执行的操作。对象是根据类规范创建的实体，就像简单变量是根据数据类型描述创建的实体一样。

C++ 提供了两个用于处理输入和输出的预定义对象（`cin` 和 `cout`），它们是 `istream` 和 `ostream` 类的实例，这两个类是在 `iostream` 文件中定义的。为 `ostream` 类定义的插入运算符（`<<`）使得将数据插入到输出流成为可能；为 `istream` 类定义的抽取运算符（`>>`）能够从输入流中抽取信息。`cin` 和 `cout` 都是智能对象，能够根据程序上下文自动将信息从一种形式转换为另一种形式。

C++ 可以使用大量的 C 库函数。要使用库函数，应当包含提供该函数原型的头文件。

复习题

1. C++ 程序的模块叫什么？

函数 实现特定的功能 函数先声明 在调用

2. 下面的预处理器编译指令是做什么用的？

```
#include <iostream>
```

包含 `iostream` 头文件 将头文件内容添加到编码

3. 下面的语句是做什么用的？

```
using namespace std;
```

使用 `std` 的命名空间 `using` 是预编译器指令

4. 什么语句可以用来打印短语 “Hello, world”，然后开始新的一行？

```
std::cout<<“Hello,world”<<std::endl;
```

5. 什么语句可以用来创建名为 `cheeses` 的整数变量？

6. 什么语句可以用来将值 32 赋给变量 `cheeses`？

```
int cheeses;
```

```
cheeses=32;
```

7. 什么语句可以用来将键盘输入的值读入变量 `cheeses` 中？

```
std::cin<<cheese;
```

8. 什么语句可以用来打印 “We have X varieties of cheese,”，其中 X 为变量 `cheeses` 的当前值。

```
cout<<“We have”<<cheese<<“varieties of cheese”<<endl;
```

9. 下面的函数原型指出了关于函数的哪些信息？

```
int froop(double t);  
void rattle(int n);  
int prune(void);
```

函数名 函数类型 形参

10. 定义函数时，在什么情况下不必使用关键字 `return`？

没有返回值的时候

11. 假设您编写的 `main()` 函数包含如下代码：

```
cout << "Please enter your PIN: ";
```

而编译器指出 `cout` 是一个未知标识符。导致这种问题的原因很可能是什么？指出 3 种修复这种问题的方法。

没有添加 `std::` ； 或者 `using namespace std;` `using std::cout;`

编程练习

1. 编写一个 C++ 程序，它显示您的姓名和地址。

1

```
#include <iostream>  
using namespace std;  
int main(void) {  
    cout<<"My name is hjs"<<endl;  
    cout<<"My home in The China"<<endl;  
    return 0;  
}
```

2. 编写一个 C++ 程序，它要求用户输入一个以 `long` 为单位的距离，然后将它转换为码（一 `long` 等于 220 码）。

```
#include <iostream>  
using namespace std;  
int main(void) {  
    int long, ma;  
    cout<<"Please enter nuber"<<endl;  
    cin>>long;  
    ma = long *220;  
    cout<<"long to ma ="<<ma<<endl;  
    return 0;  
}
```

3. 编写一个 C++ 程序，它使用 3 个用户定义的函数（包括 main()），并生成下面的输出：

```
Three blind mice
Three blind mice
See how they run
See how they run
```

```
#include <iostream>
using namespace std;
void use1(void);
void use2(void);
int main(void) {
    use1();
    use1();
    use2();
    use2();
    return 0;
}
void use1(void) {
    cout<<"Three blind mice"<<endl;
}
void use2(void) {
    cout<<"See how they run"<<endl;
}
```

4. 编写一个程序，让用户输入其年龄，然后显示该年龄包含多少个月，如下所示：

```
Enter your age: 29
```

```
#include <iostream>
using namespace std;
int main(void) {
    int age;
    cout<<"Enter your age:"<<endl;
    cin>>age;
    int month;
    month=age*12;
    cout<<"you age include in "<<month<<"month"<<endl;

    return 0;
}
```


5. 编写一个程序，其中的 `main()` 调用一个用户定义的函数（以摄氏温度值为参数，并返回相应的华氏温度值）。该程序按下面的格式要求用户输入摄氏温度值，并显示结果：

```
Please enter a Celsius value: 20
20 degrees Celsius is 68 degrees Fahrenheit.
```

下面是转换公式：

华氏温度 = $1.8 \times \text{摄氏温度} + 32.0$

```
#include <iostream>
double tmep(double a);
using namespace std;
int main(void) {
    double t, T;
    cout<<"Please enter a Celsius value"<<endl;
    cin<<t;
    T = tmep(t);
    cout<<t<<"degrees Celsius is"<<T<<"degrees Fahrenheit"<<endl;
    return 0;
}
double tmep(double a) {
    int b;
    b = 1.8*a+32.0;
    return b;
}
```

6. 编写一个程序，其 `main()` 调用一个用户定义的函数（以光年值为参数，并返回对应天文单位的值）。该程序按下面的格式要求用户输入光年值，并显示结果：

```
Enter the number of light years: 4.2
4.2 light years = 265608 astronomical units.
```

天文单位是从地球到太阳的平均距离（约 150000000 公里或 93000000 英里），光年是光一年走的距离（约 10 万亿公里或 6 万亿英里）（除太阳外，最近的恒星大约离地球 4.2 光年）。请使用 `double` 类型（参见程序清单 2.4），转换公式为：

1 光年=63240 天文单位

```

#include <iostream>
double use(double a);
using namespace std;
int main(void) {
    double light, ast;
    cout<<"Enter the number of light years:"<<endl;
    cin>>light;
    ast = use(light);
    cout<<light<<"light years ="<<ast<<"astronomical units."<<endl;
    return 0;
}
double use(double a) {
    double b;
    b = 63240*a;
    return b;
}

```

7. 编写一个程序，要求用户输入小时数和分钟数。在 main() 函数中，将这两个值传递给一个 void 函数，后者以下面这样的格式显示这两个值：

```

Enter the number of hours: 9
Enter the number of minutes: 28
Time: 9:28

```

```

#include <iostream>
using namespace std;
int main(void) {
    int hour, minute;
    cout<<"Enter the number of hours:"<<endl;
    cin>>hour;
    cout<<"Enter the number of minutes:"<<endl;
    cin>>minute;
    return 0;
}
void time(int a, int b) {

    cout<<"Time: "<<a<<":"<<b<<endl;
}

```

end:54