> 由 KiOii (_*EM_Cpper*)整理。 （KiOii (_*EM_Cpper*) makes this note.）

Chapter 1. Getting Started

1.1 Writing a Simple C++ Program

Function

- *define*: ***ReturnType   Function-Name   (Parameter-List)   {Body}***
- *main-function*:***int _cdecl main (int argc[1],char *argv[][2], char *argv[][3]) {return 0x0;}***

Type

- *define*: **the contents of a data element、operation that are on those data**
- *say*: **v** has type **T**[4]

1.1.1 Compiling and Executing Our Program

Compile

- *depend*: **operating system** and **compiler**

1.2 A First Look at Input/Output

IO

- *define:* **Input** and **Output**
- *C++ IO:* **not statement** but **standard library**[5]
- *stream:* **a sequence of characters**
- *iostream:* <***istream***> and <***ostream***>

Standard IO Objects

- *such that:* **cin、cout、cerr、clog**
- *cin:* **standard input** and has type **istream**
- *cout:* **standard output** and has type **ostream**
- *cerr:* **standard error** and has type **ostream**
- *clog:* **information about the execution of the program**

associate

- *do-what:* opearating system **associate** the **window** with **Standard IO Objects**
- So, when we read from cin, data are read from the window in which the program is executing, and when we write to cout, cerr, or clog, the output is written to the same window.

A Program That Uses the IO Library

```
#include <iostream>                         //写在同一行
int main(int argc,char *argv,char *envp[])
{
    std::cout << "Enter two numbers:" << std::endl;
    int v1,v2;
    std::cin >> v1 >> v2;
    std::cout << "The sum of"<<v1 << "and" << v2<< "is" << v1 + v2 << std::endl;
    return 0x0;
}
```

Writing to a stream

- *operator << :*
  - *two operands :* **left-hand** operand must be an **ostream** object; the **right-hand** operand is a **value to print.**
  - *the result:* the **ostream** on which we wrote the given value. (**left-hand**)*
- *mainpulator:*
  - *such that:* **std::endl**
  - *endl:* **ending the current line** and **flushing the buffer**(associated with that device)[6]

using Names from the Standard Library

namespace

- *std-namesapce:* **include all the names defined by the standard library**

Reading from a Stream

- *operator >>:*
  - *two operands:* left-hand operand is an **istream object**; the right-hand operand is **object**.
  - *the result:* **left-hand**

Exercises Section 1.2

Exercise 1.3: Write a program to print Hello, World on the standard output.

Answer

```
#include <iostream> // using IO standard library
int main(int argc,char *argv[],char *envp[])
{
    std:: cout << "Hello,World" << std:: endl;
    // std::cout << "Hello" <<"," << "World" << std::endl;
    return 0x0;
}
```

Exercise 1.4: Our program used the addition operator, +, to add two numbers. Write a program that uses the multiplication operator, *, to print the product instead.

Answer

```
#include <iostream> // using IO standard library
int main(int argc,char *argv[],char *envp[])
{
    std::cout << "Enter two numbers" << std:: endl;
    int v1,v2;
    std::cin >>v1 >> v2;
    std::cout <<"The multiplication of "<< v1 << "and" << v2 << "is"<<v1*v2<<std::endl;
    return 0x0;
}
```

Exercise 1.5: We wrote the output in one large statement. Rewrite the program to use a separate statement to print each operand.

Answer

```
#include <iostream> // using IO standard library
int main(int argc,char *argv[],char *envp[])
{
    std::cout << "Enter two numbers" << std:: endl;
    int v1,v2;
    std::cin >>v1 >> v2;
    std::cout << " The Multiplication of";
    std::cout << v1;
    std::cout << "and";
    std::cout << v2;
    std::cout << "is";
    std::cout << v1 * v2;
    return 0x0;
}
```

1.3 A Word about Comments

Comments

- *warning:* **the compiler ignores comments** (no effect on program)
- *use:* **help the human readers of our program.**
- *such that: /* comments */* and // comments
- *important:* **comments pairs do not nest**[7]

1.4 控制流

Statement

- *execute sequentially:* **from top to buttom**
- *flow-of-control statements:* allow for **more** complicated **execution paths**.

1.4.1 The while Statement

```
#include <iostream>
/*
*   sum the numbers form 1 through 10
*/
int main(int argc,char *argv[],char *envp[])
{
    int sum = 0,val = 1;
    while(val <= 10){
        sum += val;
        ++val;
    }
    return 0x0;
}
```

- *define:* ***while***(*condition*) {*statement*}

Exercist Section 1.4.1

Exercise 1.9: Write a program that uses a while to sum the numbers from 50 to 100.

Answer

```
#include <iostream>
int main(int argc,char *argv[],char *envp[])
{
    int sum = 0,val =50;
    while(val < 101)
    {
        sum += val;
        val++;
    }
    return 0x0;
}
```

Exercise 1.10: In addition to the ++ operator that adds 1 to its operand, there is a decrement operator (--) that subtracts 1. Use the decrement operator to write a while that prints the numbers from ten down to zero.

Answer

```
#include <iostream>
int main(int argc,char *argv[],char *envp[])
{
    int val =10;
    while(val >= 0)
    {
        std::cout << val << std::endl;
        val--;
    }
    return 0x0;
}
```

Exercise 1.11: Write a program that prompts the user for two integers. Print each number in the range specified by those two integers.

Answer

```
#include <iostream>
int main(int argc,char *argv[],char *envp[])
{
    int min = 0,max = 0;
    std::cout  << "Enter two number,min and max:"<<std::endl;
    std::cout << "Enter the min:";
    std::cin >> min;
    std::cout << "Entet the max";
    std::cin >> max;
    while(min != max)
    {
        std::cout << min << std::endl;
        min++;
    }
    return 0x0;
}
```

1.42 The for Statement

```cpp
#include <iostream>
/*
*   sum sum the numbers form 1 through 10
*/
int main(int argc,char *argv[],char *envp[])
{
    int sum = 0;
    for(int val = 1;val <= 10; ++val)
    {
        sum += val;
    }
    return 0x0;
}
```

1.4.3 Reading an Unknown Number of Inputs

```cpp
#include <iostream>
/*
*   read until end-of-file,calculating a returnning total of all values read
*/
int main(int argc,char *argv[],char *envp[])
{
    int sum = 0,val = 0;
    while(std::cin >> val)
      sum += val;
    std::cout << "The sum of all numbers is:" << sum << std::endl;
    return 0x0;
}
```

condition: ***cin***

- *warning:* When we use **an istream as a condition**, the effect is to test **the state of the stream.**
- *end:* when we hit **end-of-file**(Ctrl + Z or Ctrl + D) or encounter an **invalid input**, *such as* reading a value that is not an integer.

1.4.4 The if Statement

```cpp
#include <iostream>
/*
*   count how many consecutive times each distinct value appears in the input
*   such that: 1 1 1 2 2 2 3 3 3 4 5 5
*
*/
int main(int argc,char *argv[],char *envp[])
{
    int currVal = 0;
    if(std::cin >> currVal)

    {
        int count = 1;
        int val = 0;
        while(std::cin >> val)
        {
            if(val == currVal)
              count++;
            else
            {
              std::cout << currVal << "occurs " << count <<" times"<< std::endl;
              currVal = val; // reset pre-value
              count = 1;     // reset count
            }
        }
        std::cout << currVal << "occurs " << count <<" times"<< std::endl;
    }
    return 0x0;
}
```

1.5 Introucing Classes

class

- *define:* **Type** and **a collection of operations** that related to that Type (**Data Structure**)
- *three-points*
  - What is its name?
  - Where is it defined?
  - What operations does is support?

1.5.1 The Sales_item Class

name : ***Sales_item***

Data: *total revenue*、 *number of copies*、 *average sales price*

> First to know: ***what operations***

operation:

- Call a **function** named **isbn** to fetch the IBSN number from **Sales_item** object
- Use the **input(>>)** and **output(<<) operation** to read and write objects of type **Sales_item**
- Use the **assignment operator(=)** to assign one Sales_item object to another
- Use the **addition operator(+)** to add two **Sales_item** object (*the same ISBN*)
- Use the **compound assignment operator(+=)** to add one **Sales_item** object into another

Reading and Writing Sales_item

input(>>) and output(<<)

```
#include <iostream>
#include "Sales_item.h" // using Sales_item that we have defined.
int __cdecl main(int argc,char *argv[],char *envp[])
{
    Sales_item book;
    std::cin >> book;                 // using input( >> ) operator
    std::cout << book << std::endl;  // using output( << ) operator
    return 0x0;
}
```

addition operator(+)

```
#include <iostream>
#include "Sales_item.h" // using Sales_item that we have defined.
int __cdecl main(int argc,char *argv[],char *envp[])
{
    Sales_item book1,book2;
    std::cin >> book1 >> book2;
    std::cout << book1 + book2 << std::endl; // using addition operator( + )
    return 0x0;
}
```

1.5.2 A first Look at Member Functions

Our program that adds two Sales_items should check whether the objects have the same ISBN. We'll do so as follows:

```
#include <iostream>
#include "Sales_item.h" // using Sales_item that we have defined.
int __cdecl main(int argc,char *argv[],char *envp[])
{
    Sales_item book1,book2;
    std::cin >> book1 >> book2;
    // check
    if(book1.isbn() == book2.isbn())
        std::cout << book1 + book2 << std::endl; // using addition operator( + )
    {
        std::cerr << "Data must refer to same ISBN" << std:: endl;
        return -1;
    }
    return 0x0;
}
```

Member Function

- *define :* **is a function that is defined as part of a class**
- *such that:* ***book.isbn()*** use the **dot operator(.)** to say that we want "The isbn member of the object named book",use the **call operator(())** to call function.

Chapter Summary

- compile and execute simple C++ programs
- main function
- define variables
- IO operator
- if、for、while statement
- use easy class

---

1. 命令行参数的数量(arg count) ↩
2. 命令行参数数组↩
3. 环境参数数组↩
4. 如果一个名为v的变量的类型是T，我们称"v具有类型T"↩
5. C++没有提供输入输出语句，而是用标准库来实现IO↩
6. 换行，并且将与设备关联的缓冲区内容刷新到设备中，保证所有输出都写入到了输出流中，而不是停留在内存中等待写入流。↩
7. 注释界定符不能够嵌套↩