

C++ Primer Notes

mingzailao

2016-9-11

Contents

1	Begin	1
1.1	Writing a Simple C++ Program	1
1.1.1	Example 1_1_1	1
1.1.2	Compiling and Executing Our Program	2
1.1.3	Exercise	2
1.2	A First Look at Input/Output	2
1.2.1	Standard Input and Output Objects	2
1.2.2	A Program That Uses the IO Library	3
1.2.3	Writing to a Stream	3
1.2.4	Using Names from the Standard Library	3
1.2.5	Reading from a Stream	3
1.2.6	Completing the Program	4
1.2.7	EXercise	4

1 Begin

1.1 Writing a Simple C++ Program

1.1.1 Example 1_1_1

```
#include<iostream>
using std::cout;
using std::endl;
int main()
{
    cout<<"Hello world"<<endl;
    return 0;
}
```

1.1.2 Compiling and Executing Our Program

1. Comliling

```
#!/bin/bash
cd Code
g++ hello.cpp -o Hello
```

In the next, I just use Automake to compile.

2. Executing ./Hello

1.1.3 Exercise

1. Exercise 1.2: Change the program to return -1. A return value of -1 is often treated as an indicator that the program failed. Recompile and rerun your program to see how your system treats a failure indicator from main.

- (a) Answer

```
#include<iostream>

int main()
{
    return -1;
}
```

1.2 A First Look at Input/Output

1.2.1 Standard Input and Output Objects

The library defines four IO objects:

1. istream:cin
2. ostream:cout
3. cerr
4. clog

1.2.2 A Program That Uses the IO Library

```
#include<iostream>
int main()
{
    std::cout<<"Enter two numbers : "<<std::endl;
    int v1=0,v2=0;
    std::cin>>v1>>v2;
    std::cout<<"The sum of "<<v1<<"and "<<v2
        <<" is "<<v1+v2<<std::endl;
    return 0;
}
```

1.2.3 Writing to a Stream

```
std::cout<<"Enter two number"<<std::endl;
```

The << operator takes two operands: The left-hand operand must be an ostream object(std::cout); the right-hand operand is a value to print. The operator writes the given value on the given ostream

1. Notes for std::endl Writing endl has the effect of ending the current line and flushing the buffer.

1.2.4 Using Names from the Standard Library

The prefix std:: indicates that the names cout and endl are defined inside the namespace named std.

1.2.5 Reading from a Stream

```
std::cin >> v1 >> v2;
```

The input operator (the >> operator) behaves analogously to the output operator. It takes an istream as its left-hand operand and an object as its right-hand operand. It reads data from the given istream and stores what was read in the given object. The input operator returns its left-hand operand as its result.

1. Notes All the operators just like a function which can return its left-hand operand or its right-hand operand.

1.2.6 Completing the Program

```
std::cout << "The sum of " << v1 << " and " << v2  
<< " is " << v1 + v2 << std::endl;
```

It prints each of its operands on the standard output. What is interesting in this example is that the operands are not all the same kinds of values. Some operands are string literals, such as "The sum of ". Others are int values, such as v1, v2, and the result of evaluating the arithmetic expression $v1 + v2$. The library defines versions of the input and output operators that handle operands of each of these differing types.

1.2.7 EXercise

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

(a) Answer

```
#include<iostream>  
int main()  
{  
    std::cout<<"Hello world"<<std::endl;  
    return 0;  
}
```

2. 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.

(a) Answer

```
#include<iostream>  
int main()  
{  
    std::cout<<"Enter two numbers : "<<std::endl;  
    int v1=0,v2=0;  
    std::cin>>v1>>v2;  
    std::cout<<"The product of "<<v1<<"and "<<v2  
        <<" is "<<v1*v2<<std::endl;  
    return 0;  
}
```