

3 处理数据

数据的3种基本属性

- 信息存储在何处
 - 存储的值为多少
 - 存储的信息是什么类型
-
- 策略：定义一个变量
 - 声明语句指出了值的类型和符号名
 - 让程序为值分配内存
 - 并在内部跟踪该内存单元。

cout格式

➤ 整数16进制

```
cout << hex; // 程序3.4 hexoct2.cpp P46
```

```
cout<<dec; //oct
```

➤ cout.put('C');

➤ 转义序列的编码, 表3.2 P50

格式控制 (17.2.4)

```
cout.setf(ios::fixed, ios_base::floatfield);  
cout.width(12);
```

➤ 浮点数

```
cout.precision(2); // 显示精度  
cout.setf();      // 表17.1, 表17.2
```

[P17.10 iomanip.cpp](#)

const限定符

➤ `const type name = value;`

➤ 常量

➤ 符号名称指出了常量表示的内容

➤ 如果程序在多个地方使用同一个常量，则需要修改该常量时，只需修改一个符号定义即可

➤ `#define` 语句的加强！

4 C++算术运算符

➤ 自动类型转换

- 将一种算术类型的值赋给另一种算术类型的变量时，C++将对值进行转换；
- 表达式中包含不同的类型时，C++将对值进行转换；
- 将参数传递给函数时，C++将对值进行转换。

1. 初始化和赋值进行的转换
2. 以{ }方式初始化时进行的转换(C++11)
3. 表达式中的转换
4. 传递参数时的转换
5. 强制类型转换

C++11中的auto声明

- 编译器能够根据初始值的类型推断变量的类型
- 在初始化声明中，如果使用关键字auto，而不指定变量的类型，编译器将把变量的类型设置成与初始值相同
- 处理复杂类型如标准模块库(STL)中的类型时，自动类型推断的优势才能显现出来