**第2章 变量和基本类型**

**2.1 基本内置类型**

·一个char的空间应确保可以存放机器基本字符集中任意字符对应的数字值，其大小和机器字节一样。

·wchar\_t类型用于确保可以存放机器最大扩展字符集中的任意一个字符，类型char16\_t和char32\_t则为Unicode字符集服务（Unicode是用于表示所有自然语言中字符的标准）。

·大多数计算机以2的整数次幂个比特bite作为块来处理内存，可寻址的最小内存块称为字节byte，存储的基本单元称为字word，它通常由几个字节组成。

·大多数机器的字节由8个比特构成（1byte==8bite）

·32位：1word==4byte==32bite

·64位：1word==8byte==64bite

·为了赋予内存中某个地址明确的含义，必须首先知道存储在该地址的数据的类型。类型决定了数据所占的比特数以及该如何解释这些比特的内容。

·通常float: 1word(32bite)、double: 2word(64bite)、long double: 3或4word(96或128bite)

·当我们赋值给无符号类型一个超过它表示范围的值时，结果是初始值对无符号类型表示数值总数取模后的余数。（a = b x q + r，其中 |r|<|a|。负数取模：r = a - (a / b) x b, 其中a/b在c++中是向0取整）

·当我们赋给带符号类型一个超出它表示范围的值时，结果是未定义的，程序可能继续工作、可能崩溃、也可能生成垃圾数据。

·程序应该尽量避免依赖与实现环境的行为。

·切勿混用带符号类型和无符号类型。

·我们可以将整型字面值写作十进制、八进制、十六进制数的形式。以0开头的整数代表八进制，以0x或者0X开头的代表十六进制数。

·如果一个字面值连与之关联的最大的数据都放不下，将产生错误。类型short没有对应的字面值。

·由单引号括起来的一个字符称为char型字面值，双引号括起来的零个或多个字符则构成字符串型字面值。

·如果两个字符串字面值位置紧邻且仅由空格、缩进和换行符分隔，则它们实际上是一个整体。

·转义序列escape sequence

**2.2 变量**

·

**2.3 复合类型**

·

**2.4 const限定符**

·

**2.5 处理类型**

**·**

**2.6 自定义数据结构**

·