Chapter 1 开始

1.1 编译运行程序

1. 在大多系统中，main的返回值被用来指示状态，返回值0表示成功，非0的返回值则有系统来定义，通常用来指出错误类型。

2. 常见的后缀名为cpp, cxx, cp, cc, C

1.2 初识输入输出流

1. cerr来输出警告和错误信息，也被称为标准错误(standard error)。而clog用来输出程序运行时的一般性信息

2. <<运算符左侧必须是ostream对象，右侧是要打印的值。连续使用<<例如

   ```
   cout <<  "Enter two numbers" << endl;
   ```

   此时<<返回一个ostream对象，也就等价于：

   ```
   (cout <<  "Enter two numbers") << endl;
   ```

3. endl是操纵符(manipulator)的特殊值，写入endl的效果是结束当前行，并将与设备关联的**缓冲区**(buffer)中的内容疏导设备中。endl可以保证到目前为止程序所产生的输出都真正写到输出流中，而不是仅在buffer区等待输出。**注意：在调试时添加打印语句，这类语句应当保证一直刷新流，否则如果程序崩溃,输出可能还留在缓冲区内从而导致我们错误推断程序崩溃的位置**

4. std是命名空间的名字，所有标准库中定义的名字都存在std中。

1.3 注释简介

1. 单行注释以(//)开始，换行符为结束。这种注释可以包含任何文本，包括额外的双斜线(// //)

2. 另一种继承自c语言的两个界定符(/* 和 */)，可以包含除了 */ 以外的任何内容，也包含换行符

3. 注释界定符不能嵌套：

   ```
   /*
    * 注释/* */不能嵌套
    *
    *
    */
   ```

1.4 控制流

1. 当我们使用istream对象作为条件时，其效果是检测流的状态。如果流是有效的，即未遇到错误，那么检测成功，当遇到文件结束符(end-of-file)或遇到一个无效输入时（例如读入的值不是一个整数）那么istream的状态会变为无效。此时条件会变为假。

```
int value;
while(std::cin >> value){
    //code block
}
```

2. 编译错误：
    - 语法错误（syntax error）：C++语言文法上的错误。例如缺少括号、漏掉引号等等
    - 类型错误（type error）：数据项的类型赋了错误类型的值，比如string类型的变量赋值了int型的值
    - 声明错误（declaration error）：即出现未定义的变量。**C++每个名字都要先声明后使用。**
3. 每次修复一个bug就立即重新编译，按照**“编辑-编译-调试”（edit-compile-debug）**周期来工作
4. 类似于while，if语句也可以用文件输入来作为条件判断

```
int value;
if(std::cin >> value){
    //code block
}
```

1.5 类简介

1. 习惯上，头文件根据其中定义的类的名字来命名，以.h为或.hpp、.H、.hxx来作为后缀名。
2. 标准库头文件通常不带后缀名
3. 来自标准库的头文件用<>来包围，而不属于标准库的我们使用""来包围
4. 文件重定向:

```
addItems  <infile  >outfile
```

此处 <infile表示输入file中的内容， 而 >outfile则表示输出到文件outfile中

5. **成员函数(member function)是定义为类的一部分的函数，有时也被称为方法(method)**