通过对第二章的学习，我了解到C++语言是在C语言的基础上发展而来，同时它又支持面向对象的程序设计，它主要具有以下特点：

1．继承自C语言的优点：语言简洁、紧凑，使用方便、灵活；拥有丰富的运算符；生成的目标代码质量高，程序执行效率高；可移植性好等。

2．对C语言进行的改进：编译器更加严格，引入引用的概念，引入const常量和内联函数，取代宏定义等。

3 ．同时支持面向过程和面向对象的方法：在C++环境下既可以进行面向对象的程序设计，也可以进行面向过程的程序设计。因此它也具有数据封装和隐藏、继承和多态等面向对象的特征。

(1)支持面向对象编程机制，如信息隐藏、封装函数、抽象数据类型、继承、多态、函数重载、运算符重载、乏型编程（模板）

(2)兼容C语言

(3)扩充C语言，如内联函数、函数重载、名字空间、更灵活、方便的内存管理（new、delete）、引用。

(4)团队开发更简单

C++语言既保留了C语言的有效性、灵活性、便于移植等全部精华和特点，又添加了面向对象编程的支持，具有强大的编程功能，可方便地构造出模拟现实问题的实体和操作；编写出的程序具有结构清晰、易于扩充等优良特性，适合于各种应用软件、系统软件的程序设计。用C++编写的程序可读性好，生成的代码质量高，运行效率仅比汇编语言慢10%～20%。

C++语言具有以下特点：

C++是C语言的超集。它既保持了C语言的简洁、高效和接近汇编语言等特点，又克服了C语言的缺点，其编译系统能检查更多的语法错误，因此，C++比C语言更安全。

C++保持了与C语言的兼容。绝大多数C语言程序可以不经修改直接在C++环境中运行，用C语言编写的众多库函数可以用于C++程序中。

支持面向对象程序设计的特征。C++既支持面向过程的程序设计，又支持面向对象的程序设计。

C++程序在可重用性、可扩充性、可维护性和可靠性等方面都较C语言得到了提高，使其更适合开发大中型的系统软件和应用程序。

C++设计成静态类型、和C同样高效且可移植的多用途程序设计语言。

C++设计成直接的和广泛的支援多种程序设计风格（程序化程序设计、资料抽象化、面向对象程序设计、泛型程序设计）。

C++设计成给程序设计者更多的选择，即使可能导致程序设计者选择错误。

C++设计成尽可能与C兼容，籍此提供一个从C到C++的平滑过渡。

C++避免平台限定或没有普遍用途的特性。

C++不使用会带来额外开销的特性。

C++设计成无需复杂的程序设计环境。

C++使用inline来声明内联函数，这是为了在函数调用处能够直接展开函数代码而避免函数语句较少却因参数传递等代价带来的效率损失，但要注意的是，inline的声明只是用户希望这个函数内联，而编译器有权忽略这个请求。比如当函数语句十分长时，这时候函数的参数传递等开销对函数执行本身而言微不足道，如果还是在调用处展开会造成代码体积的膨胀，从而增大指令的存储空间，得不偿失，因此，对于教长的函数，编译器可能会忽略inline的请求，事实上内联函数也不应该用于较多语句的函数。

如果内联函数的声明在头文件中则头文件中不仅要包含 inline 函数的声明，而且必须包含定义，且在定义时必须加上 inline 。【关键字 inline 必须与函数定义体放在一起才能使函数成为内联，仅将 inline 放在函数声明前面不起任何作用】

inline 函数可以定义在源文件中，但多个源文件中的同名 inline 函数的实现必须相同。一般把 inline 函数的定义放在头文件中更加合适。