1.1 计算机程序设计语言的发展

计算机语言通常是能完整、准确和规则地表达人们意图，并用于指挥或控制计算机工作的“符号系统”。当使用计算机解决问题时，首先将解决问题的方法和步骤按照一定的顺序和规则用计算机语言描述出来，形成指令序列，然后由计算机执行指令，完成所需的功能。

计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的历程。

1.1.1 机器语言阶段

众所周知，在计算机内部采用二进制表示信息。机器语言（Machine Language）是用二进制代码表示的、计算机能直接识别和执行的一种机器指令的集合。它是面向机器的语言，是计算机唯一可直接识别的语言。用机器语言编写的程序称为机器语言程序(又称目标程序)。每一条机器指令的格式和含义都是由设计者规定的，并按照这个规定设计、制造硬件。一个计算机系统全部机器指令的总和称为指令系统。不同类型的计算机的指令系统不同。

例如，某种计算机的指令如下：

```
10110110 00000000      //表示进行一次加法操作
10110101 00000000      //表示进行一次减法操作
```

它们的前 8 位表示操作码，而后 8 位表示地址码。从上面两条指令可以看出，它们只是在操作码中从左边第 0 位算起的第 6 和第 7 位不同。这种机型可包含 256（即 2^8）个不同的指令。

用机器语言编写的程序能直接在计算机上运行，运行的速度快，效率高，但机器语言难以记忆，也难以操作，代码编程烦琐、易出错，而且编写的程序紧密依赖计算机硬件，程序的可移植性差。

机器语言是第一代计算机语言。

1.1.2 汇编语言阶段

为了克服机器语言的缺点，使语言便于记忆和理解，人们采用能反映指令功能的助记符来表达计算机语言，称为汇编语言（Assembly Language）。汇编语言采用的助记符比机器语言直观、容易记忆和理解。汇编语言也是面向机器的程序设计语言，每条汇编语言的指令对应了一条机器语言的指令，不同类型的计算机系统一般有不同的汇编语言。

例如，用汇编语言编写的程序如下：

```
MOV    AL   10D      //将十进制数 10 送往累加器
SUB    AL   12D      //从累加器中减去十进制数 12
 ……
```

用汇编语言编写程序比用机器语言要容易得多，但计算机不能直接执行汇编语言程序，必须把它翻译成相应的机器语言程序才能运行。将汇编语言程序翻译成机器语言程序的过程叫做汇编。汇编过程是由计算机运行汇编程序自动完成的，如图 1-1 所示。

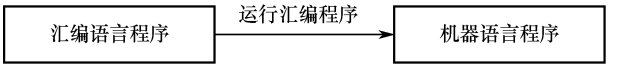

图 1-1　汇编过程

在计算机语言系统中，汇编语言仍然列入“低级语言”的范畴，它也依赖于计算机的硬件，可移植性差，但汇编语言比机器语言在很多方面都有优越性，如编写容易、修改方便、阅读简单、程序清楚等。针对计算机硬件而编制的汇编语言程序，能准确地发挥计算机硬件的功能和特长，程序精练且质量高，所以至今仍是一种常用的程序设计语言。

汇编语言是第二代计算机语言。

1.1.3　高级语言阶段

机器语言和汇编语言都是面向机器（计算机硬件）的语言（低级语言），受机器硬件的限制，通用性差，也不容易学习，一般只适用于专业人员。人们意识到，应该设计一种语言：它接近于数学语言或自然语言，同时又不依赖于计算机的硬件，编出的程序能在所有计算机上通用。高级语言（High-Level Language）是这样的语言。例如，用 C++语言编写的程序片断如下：

```
int i , j , k ;                    // 定义变量 i , j , k
cin >> i >> j ;                    // 输入 i , j 的值
k=i*j ;                            // 将变量 i , j 的值相乘，结果赋给变量 k
cout << k ;                        // 输出求积结果
```

如上例，使用高级语言编写程序时，不需要了解计算机的内部结构，只要告诉计算机“做什么”即可。至于计算机用什么机器指令去完成（即“怎么做”），编程者不需要关心。高级语言是面向用户的。

用高级语言编写的程序叫做高级语言源程序，计算机无法直接执行，必须翻译或解释成机器语言目标程序才能被计算机执行。翻译过程分为两步，即编译和连接，翻译过程如图 1-2 所示。

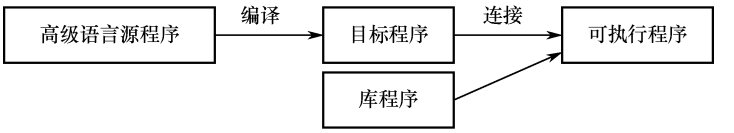

图 1-2　翻译过程

在图 1-2 中，高级语言经过编译后，得到目标程序（.obj），再与库程序连接生成可执行程序（.exe）。

程序设计语言从机器语言到高级语言的抽象，带来的主要好处如下。

（1）高级语言接近自然语言，易学、易掌握，一般工程技术人员只要几周时间的培训就可以胜任程序员的工作。

（2）高级语言为程序员提供了结构化程序设计的语法，使设计出来的程序可读性好、可维护性强、可靠性高。

（3）高级语言远离机器语言，与具体的计算机硬件关系不大，因而所编写出来的程序可移植性好，代码重用率高。

（4）由于把繁杂琐碎的事务交给了编译程序去做，所以自动化程度高，开发周期短，并且程序员得到解脱，可以集中时间和精力去设计算法和从事更重要的创造性劳动，以提高程序的质量。

高级语言是第三代计算机语言。目前广泛应用的高级语言有多种，如 BASIC、FORTRAN、PASCAL、C、C++、JAVA 及 C#等。

1.1.4 从 C 到 C++

C 语言是 AT&T 贝尔实验室的 Dennis Ritchie 在 B 语言的基础上开发出来的，1972 年在一台 DEC PDP-11 计算机上实现了最初的 C 语言。C 语言最初用做 UNIX 操作系统的开发语言，UNIX 操作系统 90%的代码由 C 语言编写，10%的代码由汇编语言编写。由于 UNIX 的成功和广泛使用，也使 C 语言成为一种普遍使用的程序设计语言。

C 语言具有如下优点。

（1）语言简洁、紧凑、使用方便、灵活。C 语言只有 32 个关键字，程序书写形式自由。

（2）运算符丰富，数据结构丰富，具有现代化语言的各种数据结构。

（3）具有结构化的控制语句（如 if…else 语句、while 语句、for 语句）。

（4）语法限制不大严格，程序设计自由度大。

（5）C 语言允许直接访问物理地址。

（6）生成目标代码质量高，程序执行效率高。

（7）用 C 语言编写的程序可移植性好。

C11 标准是 C 语言标准的第三版，前一个标准版本是 C99 标准。

但是，C 语言也有它的局限性。

（1）C 语言数据类型检查机制较弱，这使程序中的一些错误不能在编译时被自动发现。

（2）当程序的规模大到一定程度时，复杂性很难控制。

为了解决这些问题，研制 C++语言的一个首要目标就是使 C++语言突破 C 语言的局限性。同时，在 C++中引入了类等机制来支持面向对象的程序设计。所研制的这个语言最初被称为“带类的 C”，1983 年取名为 C++（C Plus Plus）。C++的喻义是对 C 语言进行“增值”。1994 年制定了 ANSI C++ 草案。后来又经过不断完善和发展，曾有 C++98、C++03、C++11、C++14 等标准。其中 C++98 是第一个正式 C++标准，C++03 在 C++98 上面进行了小幅修订，C++11（2011 年发布，包含语言的新机能并且拓展 C++标准程序库）则是一次全面的大进化，历经多次修订成为今天的 C++，且 C++仍在不断的发展中。

同样 C 语言也经历 C89、C99、C11 标准，新标准提高了对 C++的兼容性，并将新的特性增加到 C 语言中。

C++是由 C 语言发展而来的，与 C 语言兼容。C++包含了 C 语言的全部特征、属性和优点，是 C 语言的超集，同时 C++添加了面向对象编程的完全支持，是一种功能强大的面向对象程序设计语言。

1.2 过程化程序设计

程序设计的基本目标是用算法对问题的原始数据进行处理，从而获得所期望的效果。

但这仅仅是程序设计的基本要求，要全面提高程序的质量，提高编程效率，使程序具有良好的可读性、可靠性、可维护性及良好的结构，就必须掌握正确的程序设计方法和技术。

在面向对象的方法出现以前，人们都采用面向过程的程序设计方法。例如，计算炮弹的飞行轨迹。为了完成计算，就必须设计一个计算方法或解决问题的过程。由于处理的问题日益复杂，程序也就越来越复杂和庞大。20 世纪 60 年代产生的结构化程序设计方法为使用面向过程方法解决复杂问题提供了有力的手段。结构化程序设计的基本程序结构为顺序结构、选择结构和循环结构。

过程化程序设计方法的主要思想：将任务按功能进行分解，自顶向下、逐步求精。当一个任务十分复杂以至无法描述时，可按功能划分为若干个基本模块，各模块之间的关系尽可能简单，在功能上相对独立，如果每个模块的功能实现了，则复杂任务也就得以解决。

例如，一个简单的学生成绩管理系统是一个较复杂的任务，可以采用过程化设计思想完成，如图 1-3 所示。

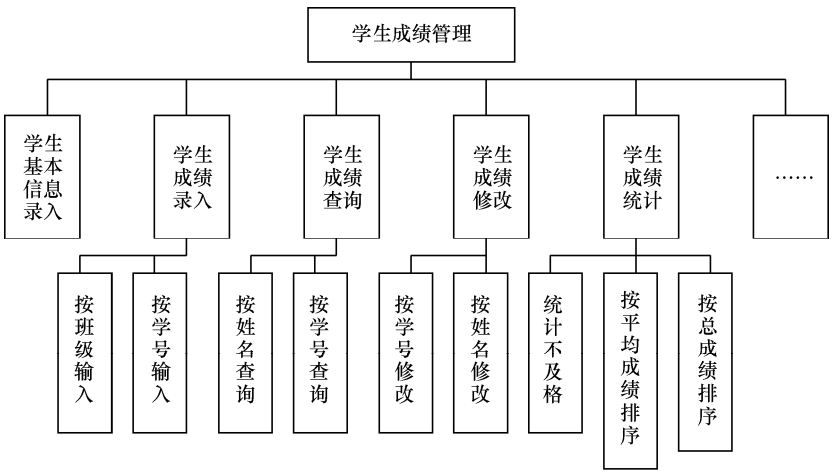

图 1-3　学生成绩管理系统设计

过程化程序设计方法中，数据与处理数据的算法是分离的，重用性差。编程的主要技巧在于追踪过程调用及哪些数据发生了变化。过程化编程思想是设计数据结构和算法，即

程序=数据结构+算法

过程化程序设计能够较好地解决一些复杂的问题，但也有很多缺点。例如，当需要处理的数据量较复杂时，数据与处理这些数据的方法之间的分离使程序变得越来越难以理解和维护。当数据结构发生变化时，必须对程序进行修改，代码的重用性差，而面向对象的程序设计方法能较好地解决这些问题。

1.3　面向对象的程序设计

面向对象的程序设计不仅吸取了结构化程序设计的优点，还考虑了现实世界与面向对象的映射关系，从而提出一种新思想。它所追求的目标是将现实世界的问题求解尽可能简化，使程序设计更加贴近现实世界，用于开发较大规模的程序，以提高程序开发的效率。面向对象的程序设计的实现需要数据封装、继承和多态技术。

1.3.1 基本概念

1. 对象

对象又称实例，是客观世界中一个实际存在的事物。它既具有静态的属性（或称状态），又具有动态的行为（或称操作）。所以，现实世界中的对象一般可以表示为：属性+行为。例如，一个盒子就是一个对象，它具有的属性为该盒子的长、宽和高等；具有的操作为求盒子的容量等。

2. 类

在面向对象的程序设计中，类是具有相同属性数据和操作的对象的集合，它是对一类对象的抽象描述。例如，将所有盒子的共同属性抽象出来就是盒子类。

类是创建对象的模板，它包含所创建对象的属性描述和方法定义。一般先定义类，再由类创建其对象，按照类模板创建一个个具体的对象（实例）。

3. 面向对象程序设计（Object Oriented Programming，OOP）

面向对象程序设计是将数据（属性）及对数据的操作算法（行为）封装在一起，作为一个相互依存、不可分割的整体来处理。面向对象程序设计的结构如下所示：

对象=数据（属性）+算法（行为）

程序=对象+对象+…+对象

面向对象程序设计的优点表现在可以解决软件工程的两个主要问题——软件复杂性控制和软件生产效率的提高。另外，它还符合人类的思维方式，能自然地表现出现实世界的实际问题。

1.3.2 面向对象程序设计的特点

面向对象程序设计具有封装、继承、多态三大特性。

1. 封装性

封装是一种数据隐藏技术，在面向对象程序设计中可以把数据和与数据有关的操作集中在一起形成类，将类的一部分属性和操作隐藏起来，不允许用户访问；将另一部分作为类的外部接口，允许用户访问。类通过接口与外部发生联系、沟通信息，用户只能通过类的外部接口使用类提供的服务，发送和接收消息；而内部的具体实现细节则被隐藏起来，对外是不可见的，增强了系统的可维护性。

2. 继承性

在面向对象程序设计中，继承是指新建的类从已有的类那里获得已有的属性和操作。已有的类称为基类或父类，继承基类而产生的新建类称为基类的子类或派生类。由父类产生子类的过程称为类的派生。继承有效地实现了软件代码的重用，增强了系统的可扩充性，

同时也提高了软件开发效率。

3. 多态性

在面向对象程序设计中，多态性是面向对象的另一重要特征。

所谓多态性是指当不同的对象收到相同的消息时，产生不同的动作。其好处是，用户不必知道某个对象所属的类就可以执行多态行为，从而为程序设计带来更大方便。利用多态性可以设计和实现一个易于扩展的系统。

1.4 简单的 C/C++程序介绍

下面通过一个简单的程序来说明 C/C++程序的基本结构。

【例 1.1】 一个简单的 C++程序。

程序如下：

```
/* -----------ch1_1.cpp：
输出一行字符："This is a C++ program."----------*/
#include <iostream>                          //预处理命令
using namespace std;                         //标准命名空间 std
void main( )
{
    cout<<"This is a C++ program. ";         //在屏幕上输出一行文字
}
```

程序运行结果：

```
This is a C++ program.
```

上述简单的 C++程序由注释语句、编译预处理命令和主函数构成。

1. 注释语句

注释是程序员为读者做的说明，用来提高程序的可读性。C++程序在编译过程中忽略注释，注释的内容不转换为目标代码。注释一般分为两种：序言注释和注解性注释。前者用于程序开头，说明程序的名称、用途、编写时间、编写人及输入/输出说明等，后者用于对程序难懂的地方做注解。注释的形式有两种，一种以“//”开头，从它开头到本行末尾之间的内容都作为注释，称为行注释；另一种是在“/*”与“*/”之间的内容，这种形式的注释可以跨多行书写，称为块注释。

2. 编译预处理命令

以符号“#”开头的行是编译预处理行，如“#include”称为文件包含预处理命令。因此，“#include <iostream>”不是 C++的语句，而是 C++的一个预处理命令，它的作用是在编译之前将系统定义的头文件“iostream”的内容包含到程序 ch1-1.cpp 中。“iostream”代表“输入/输出流头文件”，在此头文件中设置了 C++的 I/O 相关环境，定义了与数据的输入/输出有关的 I/O 流对象 cout 和 cin 等。一般来说，如果在程序中使用系统预先定义的标准函数、符号或对象，则需要在程序的头部，用“#include”预处理命令将相应的头文件包含

进来。在一些老版本 C++中 iostream 头文件是 iostream.h，在 C++新标准中用<iostream>头文件来代替。

在 C 语言中，头文件后面往往有.h，例如：

```
#include <stdio.h>
void main( )
{
printf("This is a C program. " );        //在屏幕上输出一行文字
}
```

由于 C++兼容 C 语言功能，所以 C 语言语法仍可以在 C++中使用，所以 C 的头文件 stdio.h 等依然可以继续使用，这是为了兼容 C 代码，但是它们依然有对应 C++版本的头文件，如<cstdio><cstdlib>等。

标准 C++库中的所有组件都定义在一个称为 std 的命名空间中，标准头文件如<iostream>、<string>、<exception>、<vector>、<list>等中声明的函数对象和类模板都在命名空间 std 中，因此 std 又称为标准命名空间。

3. 主函数 main()

main()函数是一个特殊的用户定义的函数，是程序执行的入口点。每个程序都必须有且仅有一个 main()函数。main 前面的 void 的作用是声明 main()函数没有返回值。

函数体用{ }括起来。在函数体中，按照算法写出语句，完成功能。每条 C++语句必须以“;”结束。本例中的主函数体中只有一个语句：cout<< " This is a C++ program. " ; cout 实际上是 C++系统定义的对象名，称为输出流对象。“<<”是“插入运算符”，与 cout 配合使用，在本例中它的作用是将运算符“<<”右边的字符串输出到显示器上。

注意，主函数名 main 全部都是由小写字母构成的。C++程序的标识符对大小写“敏感”，所以在书写标识符的时候要注意区分大小写。

例 1.1 中 void main()声明了 main()函数没有返回值。实际上，main 函数的返回值应该定义为 int 类型，C 和 C++标准中都是这样规定的。虽然在一些编译器中 void main 可以通过编译（如 Visual C++ 编译器），但并非所有编译器都支持 void main 函数，因为标准中从来没有定义过 void main 函数。在 GCC3.2 编译器中，如果 main 函数的返回值不是 int 类型，则根本无法通过编译，所以建议用 int main。main 函数的返回值用于说明程序的退出状态，如果返回 0，则代表程序正常退出，否则代表程序异常退出。

【例 1.2】 求 a1 和 a2 两个数的积。

程序如下：

```
//求两个数的积                                          //注释
#include <iostream>                                     //预处理命令
using namespace std;                                    //标准命名空间 std
int main()                                              //主函数
{                                                       //函数体开始
    int a1,a2,result;                                   //定义变量
    cout<<"please input two numbers :\n";               //输出提示信息
    cin>>a1>>a2;                                        //输入 a1,a2
    result=a1*a2;                                       //赋值语句
    cout<<"result is：  "<<result<<endl;                //输出语句
```

```
    return 0;
}                                                    //函数体结束
```

函数体中的第 1 行，定义 a1、a2 和 result 均为整型（int）变量。C++语言是强制语言，变量必须先定义再使用。定义变量后，系统给这些变量分配内存空间，用于存储变量的值。函数体中的第 2 行，字符串中的“\n”代表回车换行。函数体中的第 3 行使用了 cin， cin 是 C++系统定义的输入流对象，“>>”称为“提取运算符”，与 cin 配合使用，用于从键盘提取数据赋值给右边的变量。语句 cin>>a1>>a2;是要求用户从键盘输入两个数分别给变量 a1 和 a2，注意输入时用空格将两个数分割开。函数体中的第 4 行是将 a1 与 a2 的乘积的值赋给变量 result。函数体中的第 5 行是先输出字符串“result is:”，然后输出变量 result 的值。

如果程序运行时从键盘输入：

```
5  6  ↙
```

则输出为：

```
result is：30
```

【例 1.3】 给出两个数 x 和 y，求两数中的大者。

程序如下：

```
#include <iostream>                                  //预处理命令
using namespace std;                                 //标准命名空间 std
int max(int x,int y)                                 //max 函数定义
{
   int z;                                            //定义整型变量
   if (x>y)    z= x;                                 //如果 x>y，则将 x 的值赋值给 z
   else    z=y;                                      //否则，将 y 的值赋值给 z
   return z;                                         //返回 z
}                                                    //max 函数结束
int main( )                                          //主函数
{
   int a,b,c;                                        //定义整型变量
   cout<<"Input two numbers:\n";                     //输出提示信息
   cin>>a>>b;                                        //输入数据
   c=max(a,b);                                       //函数的调用
   cout<<" maximum number is"<<c<<endl;              //输出语句
   return 0;
}                                                    //主函数结束
```

本程序由两个函数组成：主函数和被调用的 max()函数。程序中第 2～8 行是 max()函数的定义，其功能是将 x 和 y 中的较大者赋值给 z，return 语句将 z 的值作为 max()函数的返回值。返回值通过函数名 max 带回到 main()函数的调用处。

max()函数虽然写在 main()函数的前面，但程序是从主函数 main()开始执行的。执行程序时，首先输入 a 与 b 的值；当执行到 c=max(a,b)语句时，调用 max()函数，将实参 a 和 b 分别赋给形参 x 和 y；程序转入执行 max()函数，函数 max()执行结束后将结果返回到主函数，并将计算结果赋给变量 c，然后主程序继续执行。

由上例可以看出，一个 C++程序由若干个函数组成，每个函数完成特定的功能。程序的执行总是从 main()开始，大多数函数是在程序运行时被调用的。程序按照顺序逐句执行，直到遇到函数调用语句，程序将执行被调用的函数。被调用的函数执行完成后，程序控制立即返回主调函数，并继续执行主调函数的下一行代码。

1.5 程序开发的过程

一个程序从编写源代码到最后得出运行结果一般要经历以下步骤。

1. 编写源代码

程序是一组计算机系统能识别和执行的指令，用于完成特殊的任务、功能和目的。用高级语言编写的程序称为“源程序”或“源代码”。用 C++编写的源程序的扩展名一般为.cpp，而用 C 语言编写的源程序的扩展名一般为.c。

2. 编译源代码

计算机不能直接识别和执行由高级语言编写的源代码，只能够识别和执行由 0 和 1 组成的二进制指令。因此，必须先用一种称为“编译器（Compiler）”的软件，将源代码翻译成二进制形式的目标文件（Object Program）。这种编译器软件又称为编译程序或编译系统。

编译的过程是以源代码文件为单位的。在实际程序设计过程中，程序是由一个或多个源文件组成的，编译系统分别对它们进行编译，生成多个目标文件。目标文件的扩展名一般为.obj。编译的作用是对全部源代码进行语法检查。编译结束后，如果有错则显示所有的编译错误信息。出错信息有两种，一种是错误（Error），这类错误必须改正后重新编译，否则不能生成目标文件；另一种是警告（Warning），这类错误是指一些不影响运行的小错误或程序不够优化，如定义一个变量而没有使用。

3. 连接成可执行文件

目标程序仍然不是一个可执行程序，因为目标程序只是一个个程序块，需要连接成一个适应一定操作系统环境的程序整体。为了把它们转换成可执行程序，必须进行连接。为此，系统提供的“连接程序（Linker）”将一个程序的所有目标程序和系统库文件及系统提供的其他信息连接起来，最终生成可执行程序。可执行程序的扩展名一般为.exe，可以直接执行。

4. 运行程序并分析运行结果

运行最终生成的可执行的二进制文件（.exe 文件）程序，得到程序运行结果。如果运行的结果不正确，则需要重新检查程序或算法存在的问题，并加以改正，直到运行结果正确为止。

在程序开发的编译、连接、执行等阶段都有可能出现错误，出现错误后，必须回到程序的编辑状态对源程序进行修改。C/C++程序的开发步骤如图 1-4 所示。

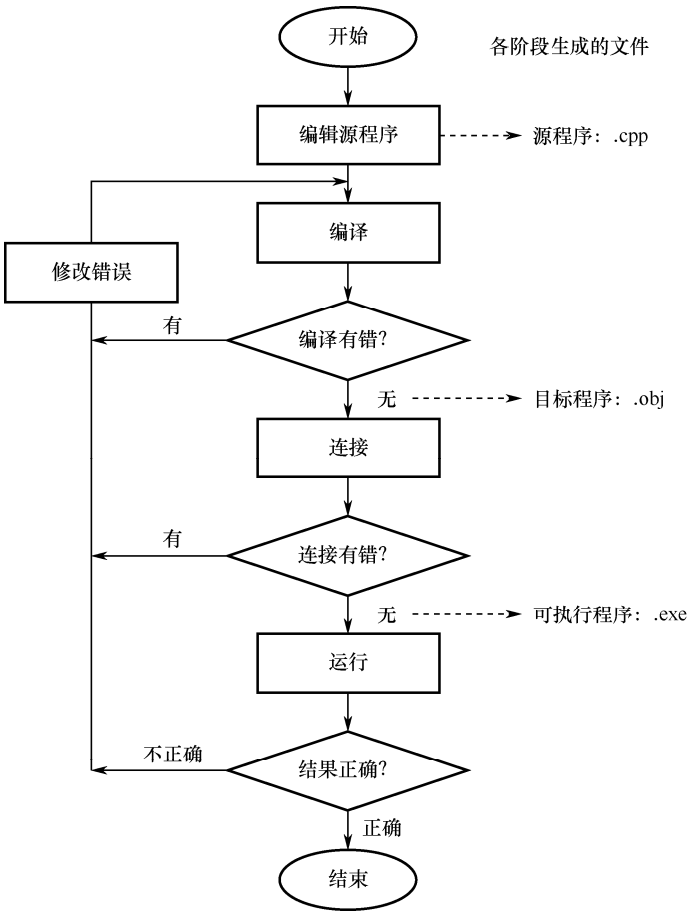

图 1-4　C/C++程序的开发步骤

1.6　C/C++上机实践

在前面已经了解到，C/C++程序设计包括编写源代码、编译、连接和运行等过程。这些过程都可以在集成开发环境中完成。集成开发环境（Integrated Developing Environment，IDE）是用于程序开发的应用程序，一般包括代码编辑器、资源编辑器、集成调试工具、调试器和图形用户界面工具等，是可以完成创建、调试、编辑程序等操作的编程环境。 C/C++集成开发环境有 Visual C++ 、Builder C++ 、Visual Studio .NET 等。Visual Studio 2013 是在 Windows 平台下运行 C/C++的集成开发环境，目前使用较广泛，本节主要介绍 Visual Studio 2013 的集成开发环境。

1.6.1　Visual Studio 2013 集成开发环境

Visual Studio 2013 提供了一个支持可视化编程，并且集界面设计、代码编写、程序调试和资源管理于一体的工作环境。用户可以依靠环境中提供的控件、窗口和方法进行各种应用程序的开发，减少了代码编写工作量，更注重程序逻辑结构的设计，大大提高了程序开发效率。

Visual Studio 2013 的开发环境主要包括菜单栏、工具栏、窗体设计器、工具箱、属性窗口、解决方案资源管理器和代码编辑器等。Visual Studio 2013 是多语言（C#、VB.NET、J#和 C++）的开发环境，C/C++程序员使用时选择 C++（Visual C++ 2013）。开发环境是程序员同 C/C++源程序的交互界面，通过它程序员可以完成程序设计的各种操作。对于集成开发环境的熟悉程度直接影响程序设计的效率，因此，希望读者能够熟练掌握 Visual Studio 2013 集成开发环境的使用，特别要掌握其调试工具的使用方法。Visual Studio 2013 的集成开发环境如图 1-5 所示。

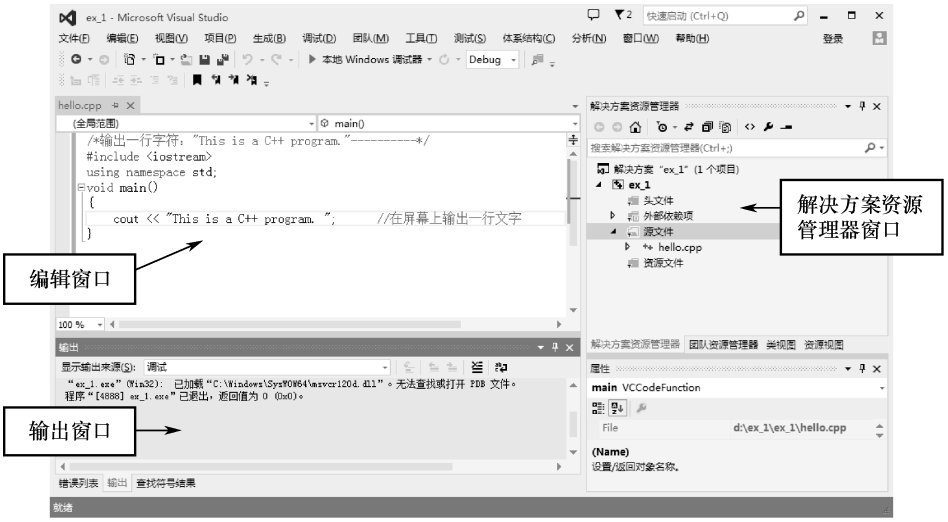

图 1-5　Visual Studio 2013 的集成开发环境

1. 菜单

集成开发环境中主要包括以下几种菜单选项。

（1）“文件”菜单。本菜单用于完成项目、解决方案及其他类型文件的相关操作，包括文件的建立、打开、保存和关闭等。

（2）“编辑”菜单。本菜单用于对控件对象和程序代码的编辑操作，如剪切、复制、粘贴、查找和替换等。

（3）“视图”菜单。根据当前的任务需要设置 Visual Studio 2013 的界面环境，通过“视图”菜单可以打开或关闭各个子窗口。

（4）“项目”菜单。本菜单用于对当前项目进行管理，如添加组件、模块和类等，并显示当前项目的结构及所包含的不同类型的文件。

（5）“生成”菜单。本菜单包括生成、重新生成、清理和发布项目。

（6）“调试”菜单。程序设计完成后，需要进行程序的调试。菜单中提供了调试程序的若干方法，如逐语句、逐过程和设置断点等。

（7）“工具”菜单。针对不同的操作，如连接到数据库、连接到服务器等，列出了 Visual Studio 2013 提供的各种不同工具。

（8）“测试”菜单。本菜单提供了和测试相关的一些功能，如加载数据文件、编辑测试运行配置等。

（9）“窗口”菜单。本菜单设置各类子窗口的显示方式和窗口之间的排列方式。

（10）“帮助”菜单。Visual Studio 2013 提供了一个基于 MSDN Library 的较完善的联机帮助系统，其中包含了.NET 支持的所有语言的信息内容及程序示例，可以通过搜索目录和查询关键词等多种方式进行检索，同时还可以和 Internet 上的相关站点进行链接，极大地方便了用户进行程序设计。

2．工具栏

工具栏以图标形式提供了常用命令的快速访问按钮，单击某个按钮，可以执行相应的操作。Visual Studio 2013 将常用命令根据功能的不同进行了分类，用户在完成不同的任务时可以打开不同类型的工具栏。标准工具栏如图 1-6 所示。标准工具栏各主要按钮的功能如表 1-1 所示。

图 1-6 标准工具栏

表 1-1 标准工具栏各主要按钮的功能

工具栏图标	名　称	功　能
	向前导航、向后导航	已打开设计窗口、代码窗口的切换
	新建项目	新建一个项目，在解决方案资源管理器中显示该项目的结构，也可以新建一个 ASP.NET 网站
	打开文件	打开 Visual Studio.NET 环境下建立的各种类型文件
	添加新项	打开右边的下拉列表，在当前项目中添加窗体、控件、各种组件和类等
	保存	保存当前项目中正在编辑的文件
	全部保存	保存正在编辑的项目的所有文件
	撤销和重做	撤销上次操作和恢复上次操作
	查找	打开“查找”对话框，查找相应的内容，包括快速查找、在文件中查找和查找符号等操作
	启动调试	开始运行当前的程序项目

3．解决方案资源管理器（Solution Explorer）

使用 Visual Studio 2013 开发的每一个应用程序叫解决方案（以.sln 为后缀名），每一个解决方案可以包含一个或多个项目（以.vcxproj 为后缀名）。一个项目（Project）通常是一个完整的程序模块，一个项目可以有多个文件项（.cpp，.c 或 .h 等）。“解决方案资源管理器”子窗口显示 Visual Studio.NET 解决方案的树型结构。在“解决方案资源管理器”中可以浏览组成解决方案的所有项目和每个项目中的文件，可以对解决方案的各元素进行组织和编辑。

解决方案资源管理器窗口一般位于屏幕右侧，如图 1-5 所示。单击窗口底部的标签可以从一个视图切换到另一个视图。每个视图都是按层次方式组织的，可以展开文件夹和其中的项查看其内容，或折叠起来查看其组织结构。

（1）类视图（ClassView）。ClassView 显示所有已定义的类及这些类中的数据成员和成员函数。在 ClassView 中，文件夹代表工程文件名。展开 ClassView 顶层的文件夹后，显示工程中所包含的所有类。展开类可查看类的数据成员和成员函数，以及全局变量、函数和类型定义。

（2）资源视图（ResourceView）。ResourceView 显示项目中所包含的资源文件。

1.6.2 开发 C/C++的程序过程

1. Visual Studio 2013 的启动

选择“开始”按钮→“所有程序”→“Visual Studio 2013”→“Visual Studio 2013”或桌面上 Visual Studio 2013 的快捷方式，启动 Visual Studio 2013 集成环境。

2. 创建 Windows 控制台应用程序

Visual Studio 2013（VS2013）不支持单个源文件的编译，必须先创建项目（Project）再添加源文件。

打开 Visual Studio 2013（VS2013），在菜单中选择“文件”→“新建”→“项目”命令，或者按 Ctrl+Shift+N 快捷键，弹出图 1-7 所示的对话框。

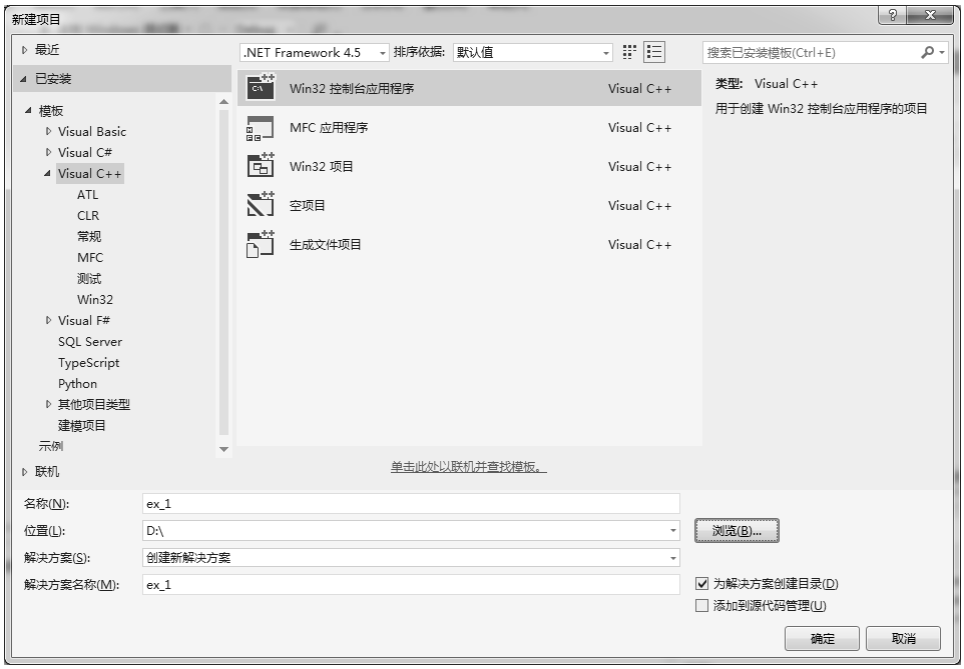

图 1-7 “新建项目”对话框

在上方的项目类型下拉列表中选择默认的“.NET Framework 4.5”。在左侧的“已安装的模板”中，选择“Visual C++”语言，在中间窗格选择“Win32 控制台应用程序”，在下面的“名称”框中输入要创建的项目名称（如 ex_1），“位置”框中输入要创建的项目保存位置（文件夹），单击“确定”按钮，弹出图 1-8 所示的向导对话框：

图 1-8　向导对话框——概述

单击“下一步”按钮，弹出图 1-9 所示的对话框。

图 1-9　向导对话框——应用程序设置

先取消“预编译头”，再勾选“空项目”，然后单击“完成”按钮就创建了一个新的项目。

3. 添加 C 或 C++源文件

在资源管理器窗口中，在新建项目 ex_1 下的“源文件”处右击鼠标，在弹出的菜单中选择“添加”→“新建项”，弹出添加源文件对话框，如图 1-10 所示。

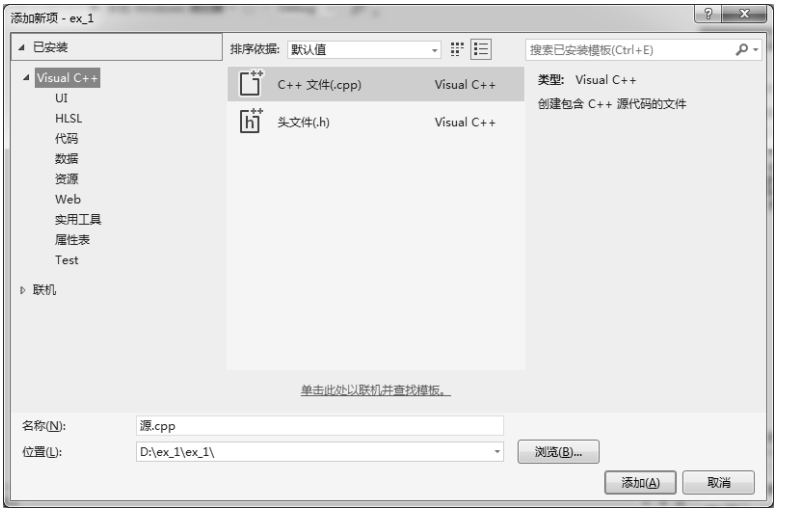

图 1-10　添加源文件对话框

在“代码”分类中选择“C++文件(.cpp)”，并输入文件名称（如 hello.cpp 或 hello.c），单击“确定”按钮，完成新建 C++源程序文件。

4. 完成 C++程序的编译、连接和运行

在出现的源代码编辑窗口中输入源程序代码。输入代码后，单击“启动调试”按钮，或者按下 F5 键，就可以完成程序的编译、连接和运行。

也可以在工具栏上单击右键，在快捷菜单中选择“调试”项，出现图 1-11 所示的“调试”工具栏，通过此工具栏可以调试程序的逻辑错误。如果运行程序，则单击图 1-11 中的  （启动调试）按钮运行（执行）程序，查看运行结果。在 Windows 下可能发现在运行程序的时候，还没来得及查看结果窗口就已经关闭了，可以加入 system(" pause ");来解决此问题。

图 1-11　“调试”工具栏

习题 1

一、选择题

1. C++源程序的扩展名是_____，C 语言源程序的扩展名是_____。

A. .cpp　　B. .c　　C. .obj　　D. .exe

2. 程序中主函数的名字为_____。

A. main　　B. MAIN　　C. Main　　D. 任意标识符

3. C++的合法注释是_____。

A. /* this is a c++ program /*　　B. // this is a c++ program

C. "this is a c++ program"　　D. / this is a c++ program /

4. C++程序设计的几个操作步骤依次是_____。

A. 编译、编辑、连接、运行　　B. 编辑、编译、连接、运行

C. 编译、运行、编辑、连接　　D. 编译、运行、连接、编辑

二、简答题

1. C 语言与 C++语言的关系是什么？
2. 面向过程的程序设计与面向对象的程序设计有什么异同？
3. 简述 C++程序的开发步骤。
4. 参阅资料简述 C++标准的变化历程。