

第一部分 上机实验

一、上机环境介绍

1. VS2010 集成环境和程序开发过程简介

在 vs2010 集成开发环境中集成了编辑器、编译器、连接器以及程序调试环境，覆盖了开发应用程序的整个过程，用户在这个环境中可以开发出完整的应用程序。

程序开发的一般步骤是：输入源程序文件(即编辑)、编译、连接、运行、调试和修改，如图 1-1 所示：

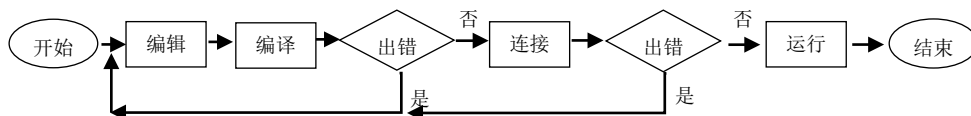


图 1-1

上图最后一步，若运行出错，需要调试和修改程序。


上机实验步骤如下：

1. 建立文件夹，存放自己的实验内容

一般来说，机房 D 盘开放给同学使用，为了防止病毒，D 盘当天晚上 12 点清空。在开发程序之前，在 D 盘建立一个以自己的学号姓名为名字的文件夹，如“051910199 张三”，用于存放本次实验内容。

2. 启动 VS2010 应用程序

可用下述两种方法之一，启动后界面如图 1-2 所示。

(1) 双击桌面应用程序图标  启动。根据机房设置，此图标可能在桌面某个子文件夹中，比如在“应用程序”子文件夹中。

(2) 从任务栏“开始”菜单，找到“Microsoft Visual Studio 2010”应用程序，启动之。

3. 通过 VS2010 项目管理，开发 C++ 语言控制台程序的过程

在图 1-2 中，可以通过下述四种方法之一，启动新建项目：

- ① “文件”菜单 | “新建” | “项目”
- ② 工具栏上的“新建项目”按钮
- ③ 起始页上的“新建项目”按钮
- ④ 快捷键 Ctrl+Shift+N

然后进入图 1-3 所示界面。

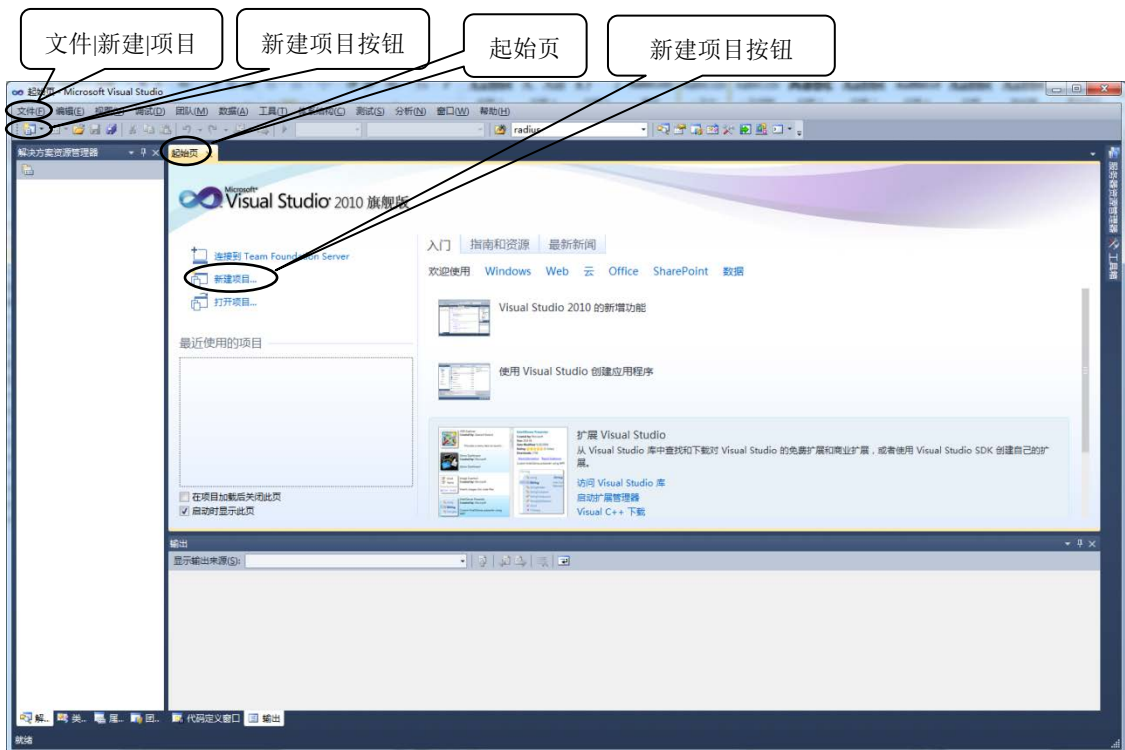


图 1-2

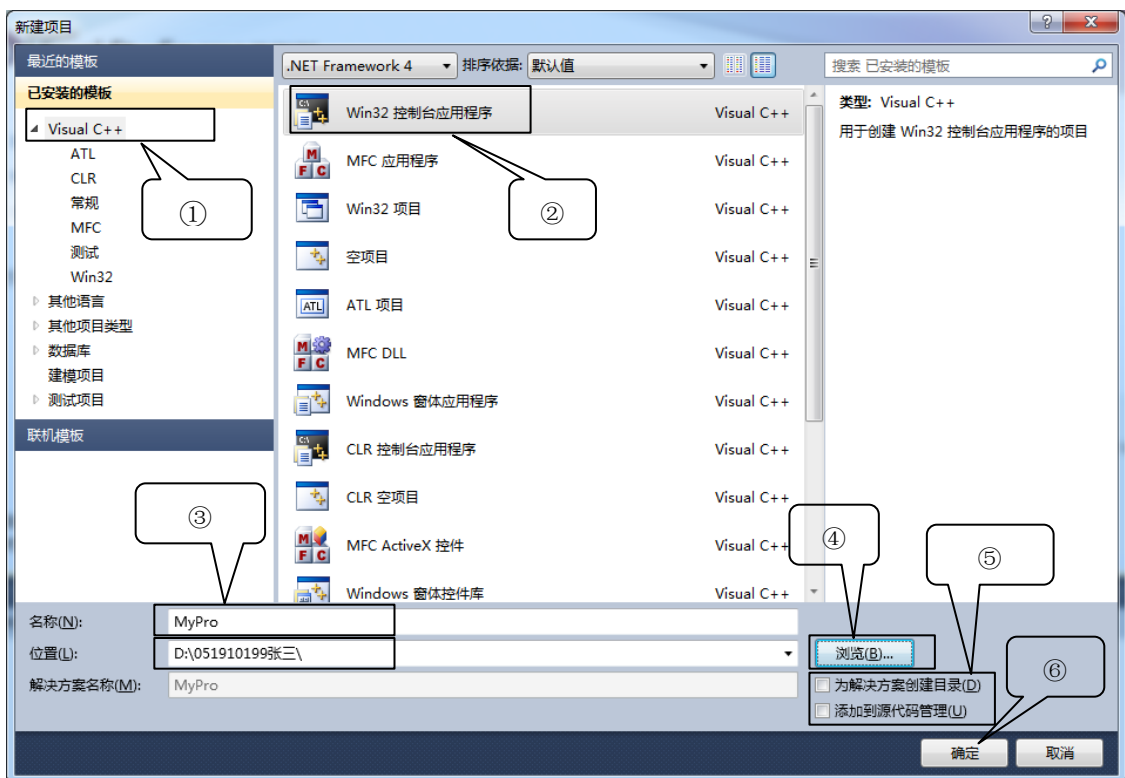


图 1-3

在图 1-3 中，按照如下步骤做：

- ① 选择“VisualC++”
- ② 选择“Win32 控制台应用程序”
- ③ 输入项目“名称(N)” MyPro，“名称”根据自己的需要取名，“项目方案名称(M)”会随着项目“名称(N)”而变化
- ④ 单击“浏览”按钮，选择项目文件夹 MyPro 的存储“位置(L)”，一般选择开始实验时第 1 步已创建的文件夹即 D 盘的文件夹“051910199 张三”。项目 MyPro 中所包含的跟项目有关的多个文件均存储于 MyPro 文件夹中，而文件夹 MyPro 存储在文件夹“D:\051910199 张三”中。
- ⑤ 取消勾选“为解决方案创建目录”“添加到源代码管理”，即把前面方形勾选项取消。
- ⑥ 按“确定”按钮，进入图 1-4。

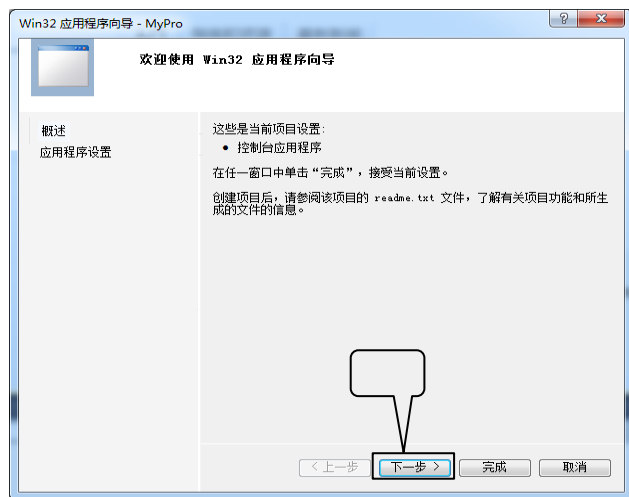


图 1-4

单击图 1-4 中的“下一步”按钮，进入图 1-5。



图 1-5

在图 1-5 中，①勾选“空项目”②点击“完成”按钮，进入图 1-6。

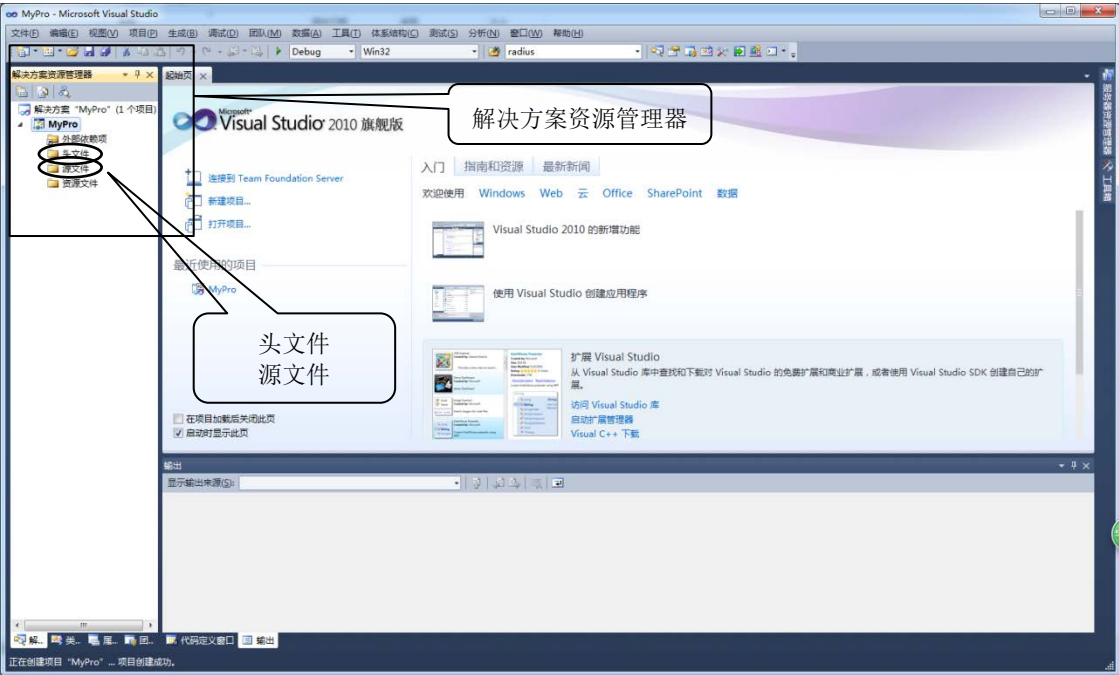


图 1-6

在图 1-6 窗口左上角的“解决方案资源管理器”中，可以看到已生成的项目解决方案 MyPro，初始时该项目是空的，需要给项目添加源程序，步骤是，右击“源文件”可添加 .cpp 源文件，右击“头文件”可添加 .h 头文件（现在暂时不考虑头文件，后续章节会使用）。

如图 1-7 所示，右击“源文件”选“添加”“新建项”，进入图 1-8。

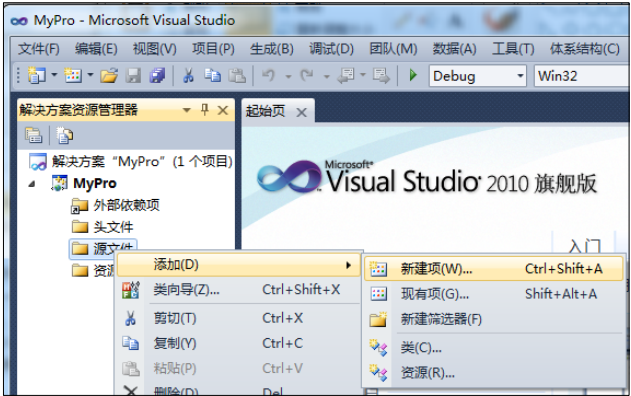


图 1-7

在图 1-8 中，①选择“代码”，②选择“C++文件(.cpp)”。注意若此步选择下一行“头文件(.h)”，则创建头文件。③输入 C++源程序名 ex0101，表示创建实验一第 1 题的源程序，因默认扩展名是 .cpp，所以完整的源文件名是 ex0101.cpp。源文件的存储“位置(L)”默认在文件夹“D:\051910199 张三\MyPro”中，建议④点击“浏览”按钮改变源文件的存储位置，把所有的源文件都存储在“D:\051910199 张三”中，这样便于完成实验后上交作业或拷贝“带走”源程序。⑤点击“添加”按钮，进入图 1-9。

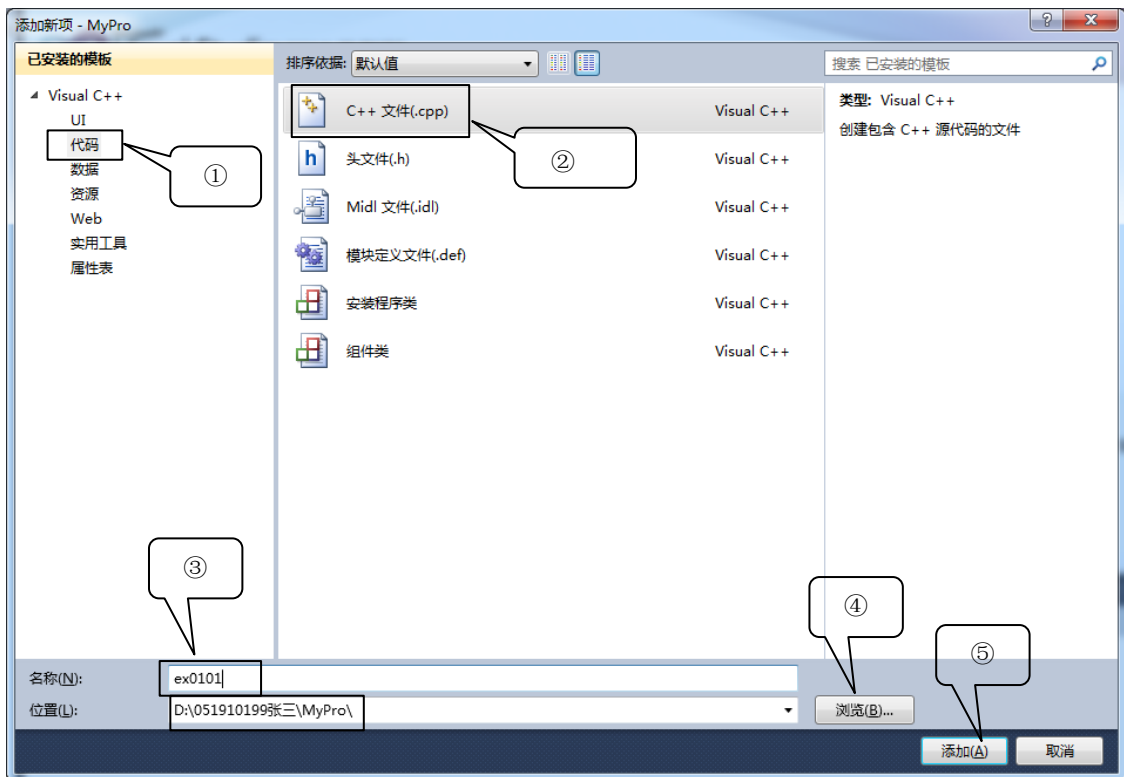


图 1-8

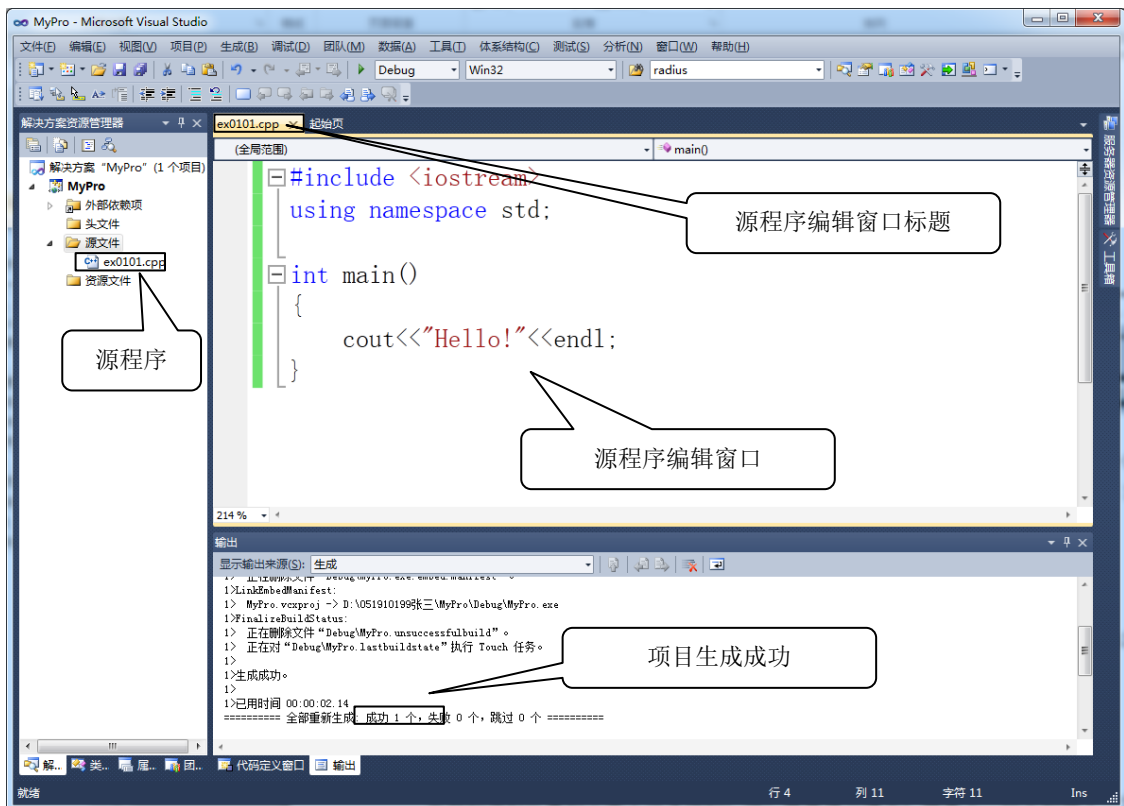


图 1-9

在图 1-9 中,观察,“项目方案资源管理器”的“源文件”下有源程序“ex0101.cpp”,现在可在“源程序编辑窗口”中输入如图 1-9 中所示的源程序(注意该窗口标题也是 ex0101.cpp),然后执行菜单“生成(B)”中的“生成解决方案(B) (快捷键 F7)”或“重新生成解决方案(R) (快捷键 Ctrl+Alt+F7)”命令,产生可执行程序,最后执行程序,即执行菜单“调试(D)”中的“开始执行(不调试)(H) (快捷键 Ctrl+F5)”命令,程序执行结果如图 1-10 所示,该图背景及文字颜色已调整,实际上背景是黑色、文字是白色。可按任意键关闭该窗口。

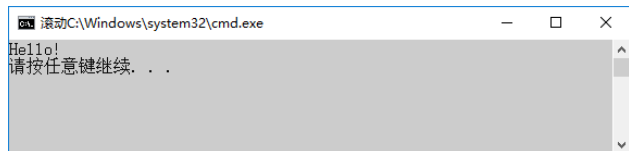


图 1-10

一个项目完成后,可以执行图 1-9 窗口中的“文件(F)”菜单中的“关闭解决方案(T)”命令关闭当前项目,再从头开始创建另一个新项目。但是这样处理非常麻烦,因为每次都要创建新项目。

下面介绍一个简单办法,每次上机实验只需创建一个项目,当完成一个源程序后,将当前源程序从项目中移除,然后再在当前项目中新建另一源程序,如此循环,可在一次实验时,仅创建一个项目,完成多个源程序。

如何从一个项目中移除一个源程序,如图 1-11 所示:右击解决方案“源文件”下的源程序名,如 ex0101.cpp,选择“从项目中排除”即可。

然后,可以从图 1-6 开始,在一个“空”的项目中,右击“源文件”添加一个新的.cpp 源文件,开始创建一个新的源程序。当然,对以前已经做好的源程序,如果需要修改,也可以将当前项目清空(即移除所有源程序)后,再重新将以前的源程序添加进来,此时如图 1-7 所示,右击“源文件”时,需要选择“添加”“现有项”,选择已有程序添加进来即可。

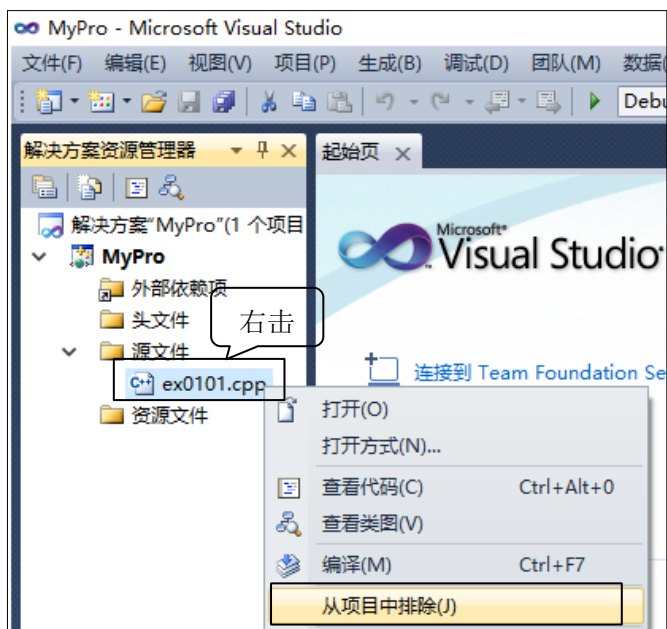


图 1-11

仿照前面的过程，又创建了一个新源程序 `ex0102.cpp`，并且在图 1-8 这一步，选择将源程序均放置在文件夹“D:\051910199 张三”中，此时在操作系统中观察文件夹“D:\051910199 张三”中有两个源程序文件 `ex0101.cpp` 和 `ex0102.cpp`，如图 1-12 所示。将一次实验的源程序放在一个文件夹中，便于上交或“带走”全部源程序。文件夹 `MyPro` 中存放的是与项目相关的多个文件，可以暂时不用关心。

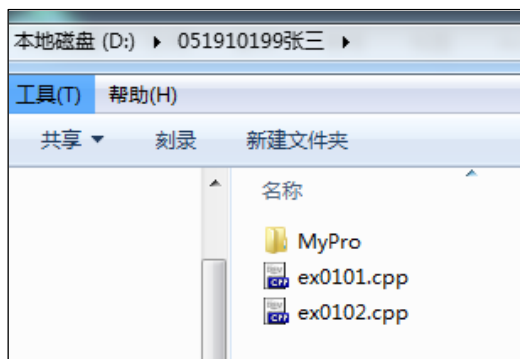


图 1-12

VS2010 若干使用技巧：

(1) 公共机房开发的源程序“带走”后，下次再来修改，可以先建立一个项目，然后依次添加、排除需要修改的源程序进行修改。注意，公共机房开放给学生使用的 D 盘，一般每天都是清空的，即第 1 天在 D 盘保存的文件，第 2 天就没有了，所以必须备份自己的程序。下次来上机，需要重新建立项目。

(2) “起始页”中的内容如图 1-13 所示，可以新建、打开项目，可以打开最近使用的项目。使用“视图”菜单“起始页”命令，可以打开和关闭起始页的显示。如果想让起始页

始终存在，可以取消勾选“设置”中的“在项目加载后关闭此页”。



图 1-13

(3) 如果不是在公共机房，而是使用自己的电脑开发程序，已建立的项目下次开机后可以直接打开，方法有两个①在“起始页”上点击最近使用的项目名称，如 MyPro ②在“文件”菜单的“最近使用的项目和解决方案”中选择打开。

2. VS2010 程序调试

程序的开发步骤是：输入源程序、编译、连接、运行，编译连接阶段如果出错，需要修改错误，直至将一个完整、正确的程序设计出来。程序调试是指对程序的查错和排错工作。

程序的错误一般分为语法错误和逻辑错误两种。编译时出现的错误一般是语法错误，可根据编译时给出的“出错信息”，判断出现

错误的语句。逻辑错误一般指程序算法流程设计和处理上的错误，程序运行结果不正确一般是程序逻辑错误造成的，所以需要根据算法流程来查找出错原因，这不是一件简单的事情，要求对算法的各个环节以及实现算法的各条语句有一个全面充分的认识，才能找出原因。程序的运行时错误，往往是由于程序对系统资源的使用不当造成的，一般是由程序的逻辑错误引起的。注意，一般地，编译只能查出语法错误，而程序的逻辑错误在编译阶段无法查出，一般只有当运行出错时，才会去查逻辑错。

调试程序一般经过以下几个步骤：

(1) 首先进行人工检查，即静态检查。作为一个程序员应当养成严谨的科学作风，每一步都要严格把关，不把问题留给后面的工序。

为了更有效地进行人工检查，所编写的源程序应力求做到以下几点：

①采用缩进式书写格式，每行写一条语句，以增强程序的可读性。例如复合语句花括弧的匹配，若采用缩进式书写格式，就不容易出错。

②在程序中尽可能多地加注释，以帮助理解每段程序的作用。

③在编写复杂程序时，不要将全部语句都写在 main 函数中，而要多使用函数，用一个函数来实现一个独立的不可分的功能。这样既易于阅读也便于调试。

(2) 在人工（静态）检查无误后，进入调试。通过编译、运行发现错误的过程称为动态检查。在编译时系统会给出语法错误的信息（包括哪一行有错以及错误类型），可以根据提示信息具体找出程序中的出错之处并改正。应当注意的是：有时提示的出错行并不是真正的出错行，如果在提示的出错行中找不到错误，应当到可能的相关行再找。另外，有时提示的出错的类型并非绝对准确，由于出错的情况繁多而且各种错误互有关联，因此要善于分析，找出真正的错误，而不要死抱住提示的出错信息不放。

如果系统提示的出错信息有很多条，应当从第一条开始，由上到下逐一改正。有时显示多条错误信息往往使人感到问题严重，无从下手。其实可能只有 1~2 个错误。例如，若某一变量未定义，编译时就会对所有使用该变量的语句发出错误信息。此时只要在前面增加一个变量定义，所有的错误都消除了。因此，在程序中的第一处错误修改完成之后，应立刻重新编译该程序。

(3) 在改正语法错误（包括“错误 error”和“警告 warning”）后，程序经过连接（link）就得到可执行程序，即生成了项目解决方案。运行程序，输入程序所需数据，就可得到运行结果。运行时如果出现“运行时错误”，则应着重查找程序中的逻辑错误。

在查找错误时，应该对运行结果进行认真分析，判断是否符合题目要求。有的初学者看到针对一组数据输出了正确结果，就认为程序没有问题了，不作认真分析，这是危险的。有时程序比较复杂，难以根据一组数据判断结果是否正确，因为有可能针对另外一组数据程序的运行结果就不对了。所以应事先设计好一批全面的“实验数据（测试数据）”，以验证在各种情况下程序的正确性。

事实上，当程序复杂时，很难把所有的可能的输入数据方案全部都测试一遍，选择典型数据即可。

若程序的运行结果不对，大多属于逻辑错误。对这类错误往往需要仔细检查和分析程序逻辑才能发现。可以采用以下方法：

- 先检查流程图有无错误，即算法有无问题，如有错则改正之，接着修改程序。将程序与流程图（或伪代码）仔细对照，如果流程图正确，程序的错误会很容易发现。

- 采取“分段检查”的方法。在程序不同的位置加入几个输出（cout）语句，输出有关变量的值，逐段往下检查。直到找到在某段中的变量值不正确为止。这时就把错误局限在这一段中了。不断缩小“查错区”，就可能发现错误所在位置。

下面以一个具体实例，给大家介绍与“分段检查”法类似的简单的程序调试方法。

在学习了函数和数组后，程序变的越来越复杂了，这时可使用 vs2010 提供的强有力的调试工具，跟踪程序的执行找出程序中的错误。

下列程序的功能是：输入数组的 5 个元素值，将数组元素逆序存储后输出。

```
#include <iostream>
using namespace std;

void reverse(int a[], int n)
{   int t,i,j;
    for(i=0, j=n-1; i<j; i++, j--)
    {   t=a[i];
        a[i]=a[j];
        a[j]=t;
    }
}

int main()
{   int a[5], i;
    cout<<"Please input 5 elements: ";
    for(i=0; i<5; i++)
        cin>>a[i];           //A
    reverse(a,5);             //B
    for(i=0; i<5; i++)        //C
        cout<<a[i]<<" ";
    cout<<endl;
    return 0;
}
```

输入以上程序并生成项目解决方案后，按 F10 键，启动并进入调试状态，界面如图 1-14 所示，图中程序左侧有一个黄色的向右的箭头，表示当前正在执行的语句，每按一次 F10 键（F10 可实现**单步跟踪执行**），执行一条语句。在执行过程中，把鼠标放在某个变量名上可观察变量的当前值。假定本程序运行时输入的数组 5 个元素值是 1、2、3、4、5，如图 1-14 所示，若将鼠标放在数组名上并单击下面的“+”号，则可观察数组全部元素的值。又如，若将鼠标放在变量 i 上，可以观察 i 的值。同时在下方的“自动窗口”中也可以观察数组元素和变量的值。在调试过程中，可以随时结束调试，方法是执行“调试”菜单“停止调试”命令（或按快捷键 shift+F5）。

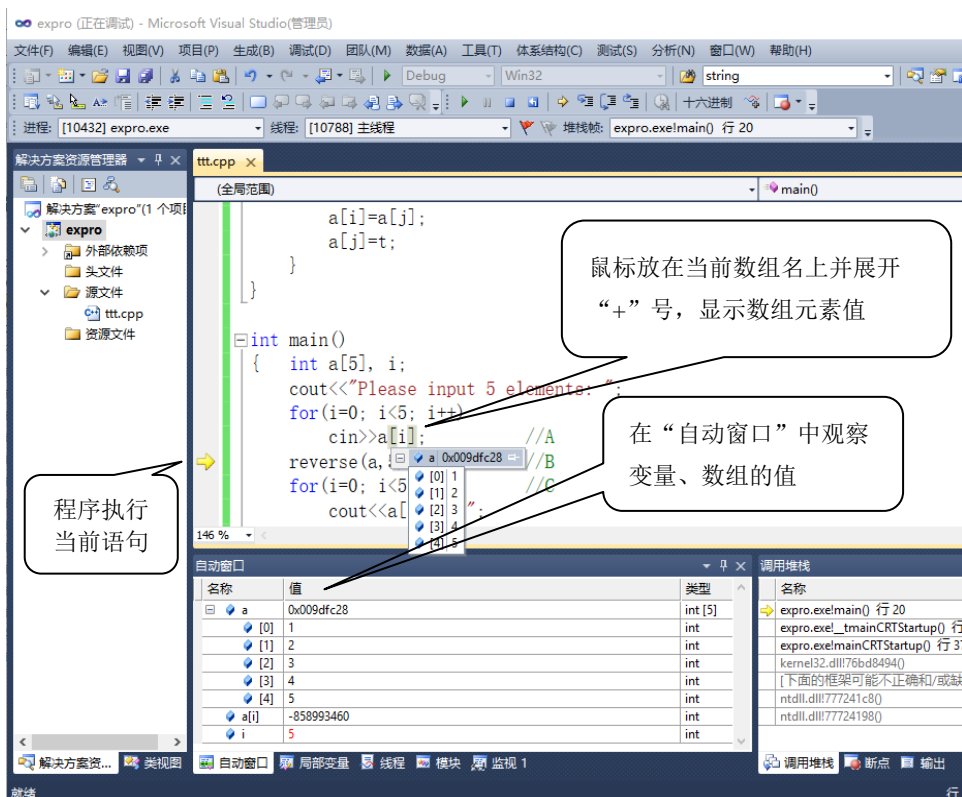


图 1-14

在调试阶段，各常用快捷键及功能如表 1-1 所示：

表 1-1 常用调试快捷键及功能

快捷键	功能	菜单命令
F10	单步跟踪执行，不进入被调用函数	调试 逐过程
F11	单步跟踪执行，进入被调用函数	调试 逐语句
Shift+F5	退出调试状态	调试 停止调试
F5	执行到下一个断点处	调试 启动调试/继续
F9	在光标所在行设置/取消断点	调试 切换断点

在图1-14中，调试上述程序时，按一次F10键，执行一条语句，当执行到A行时，无法继续执行，因为程序等待输入数据，可以在Windows的任务栏中切换到程序的数据输入窗口，在提示信息“Please input 5 elements:”后输入5个任意数据，例如输入1 2 3 4 5，按<回车>键后，再从任务栏回到程序调试窗口。当执行到B行，遇到reverse()函数调用语句时，如果不想跟踪进入被调函数，则按F10键（被调函数一次执行完毕），如果想跟踪进入被调函数，则按一次F11键，进入被调函数后再按F10键单步跟踪执行。

在调试程序时，有时感觉程序的某些部分不需要单步执行，例如循环输入数组元素时，或者已经确认一段程序是正确的，此时可以在程序的某处设置一个断点，让程序跳过不需要单步跟踪的部分，一次性执行到断点处，再观察程序执行到断点处各变量的变化情况。

设置、取消断点的方法是，在光标所在当前行按 **F9** 键，**F9** 是一个开关键，按一次设置，再按一次取消。若将某行设置为断点，则此行的左边出现一个棕红色的圆点，取消断点后，棕红色圆点消失。若程序中设置了断点，可以按 **F5** 键将程序运行到第一个断点处，暂停，等待调试的下一动作，程序员观察完毕，可以继续按 **F5** 键执行到下一断点，也可以按 **F10** 键继续单步执行。

例如，调试上述程序时，可在 **B** 行和 **C** 行分别设置第 1 个和第 2 个断点。按 **F5** 键开始调试，程序执行到第 1 个断点（期间需要输入数据）处暂停，程序员观察 **a** 数组中的值，然后按 **F5** 键，执行到第 2 个断点处，再次观察 **a** 数组中的值，以确认 **reverse()** 函数是否执行正确。

再次提醒，在调试过程中，可以随时结束调试，方法是执行“调试”菜单“停止调试”命令（或按快捷键 **shift+F5**）。

```
=====  END  =====
```