

北 京 林 业 大 学

2019 学年—2020 学年第 1 学期 面向对象程序设计 B

实验指导书

专业名称： 地信 实验学时： 2 任课教师： 卢昊

实验题目： 实验一 从 C 到 C++

实验环境： Visual Studio

实验目的：

1. 复习 C 语言基本知识和 C 程序写法、结构体用法、文件读取方法；
2. 掌握 C++语言基本语法知识和 Visual Studio 环境下 C++程序编写、调试方法；

实验要求：

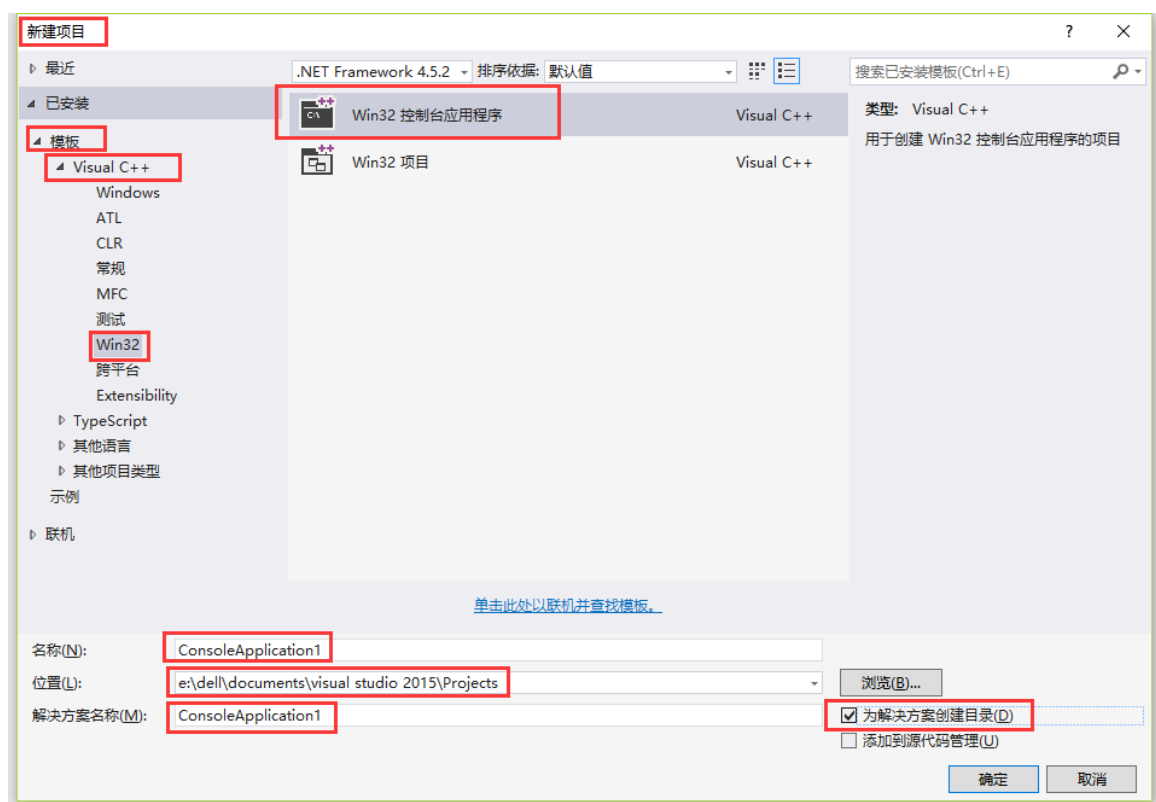
1. 熟悉 Visual Studio 开发工具的基本用法；
2. 按要求创建、命名和管理解决方案、工程目录；
3. 使用 C 结构体和 FILE 指针读取二进制图像文件；
4. 学会使用 C++语言重写 C 程序；
5. 学会使用 VS 的基本调试功能。

实验内容和步骤:

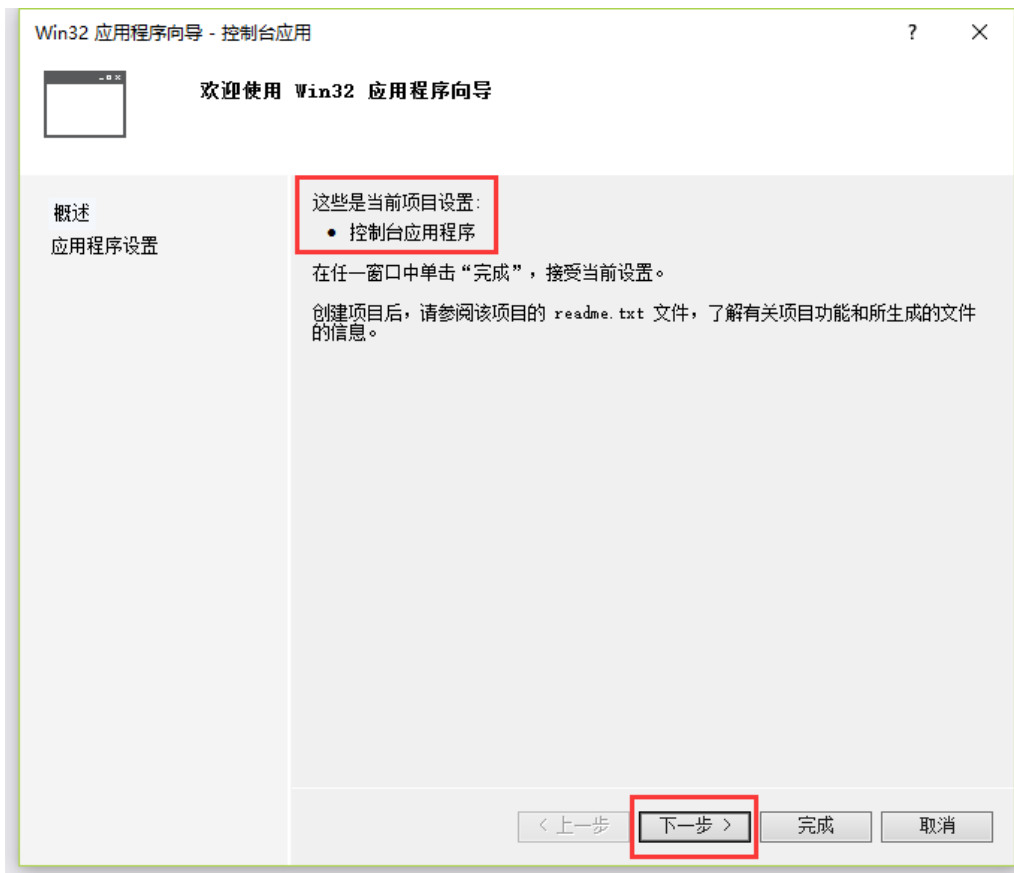
练习 1: 创建解决方案、工程

步骤 1: 运行 Visual Studio, 点击**文件-新建-项目**, 打开下图所示新建项目对话框。**注意**下图仅为示意, 标记的位置相关设置和文件名需根据你的实际情况**填写或修改**。

其中: 模板下选择 **Visual C++**, **Win32**, 中间选择 **Win32 控制台应用程序**。名称为工程的名字, 位置为工程存放的本地路径 (建议存放在非系统盘)。勾选**“为解决方案创建目录”**。



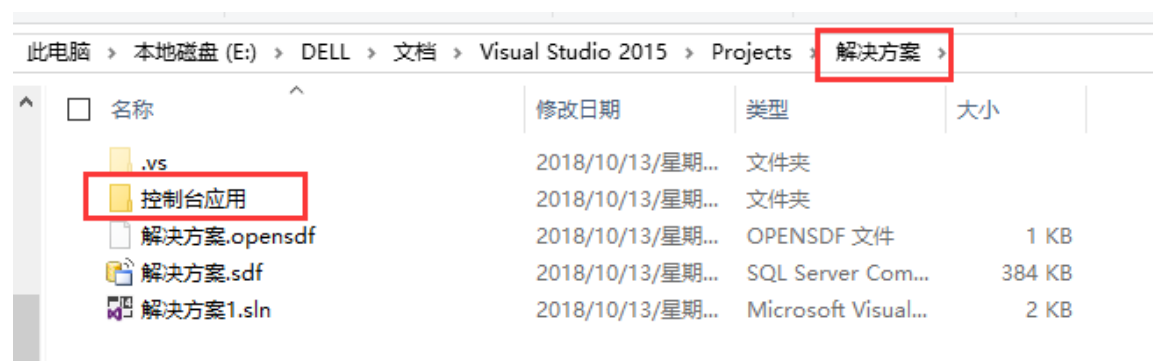
本次实验中, 解决方案名称修改为“你的学号_s1”, 如“180XXXXXX_s1” (注: 这是第 1 次实验, 后续实验改成相应编号)。名称一栏修改为“你的学号_c”。点击确定, 进入向导页, 如下图所示, 检查设置是否为控制台应用程序, 点击下一步。



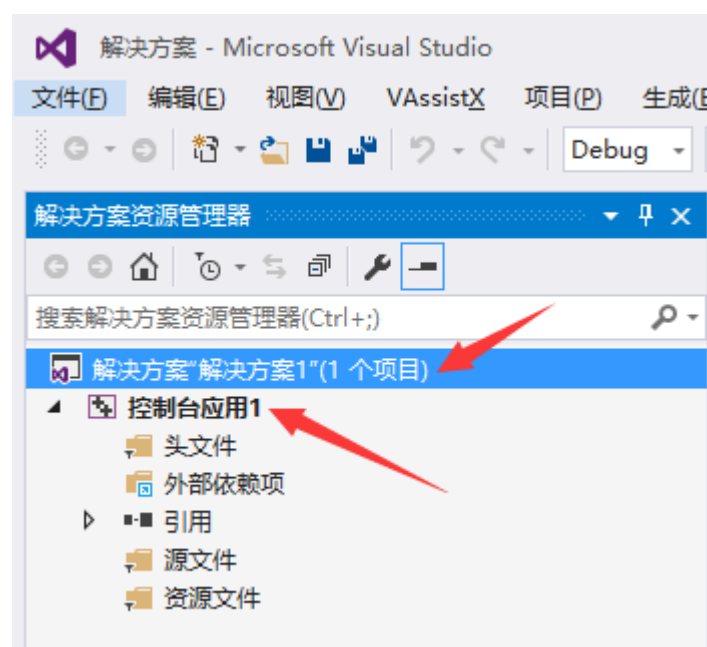
然后在对话框中勾选“空项目”，完成创建。



创建完成后，打开刚刚设置的“位置”路径，检查是否自动创建了解决方案目录“你的学号_s1”，解决方案目录中是否创建了工程目录“你的学号_c”，类似下图所示：

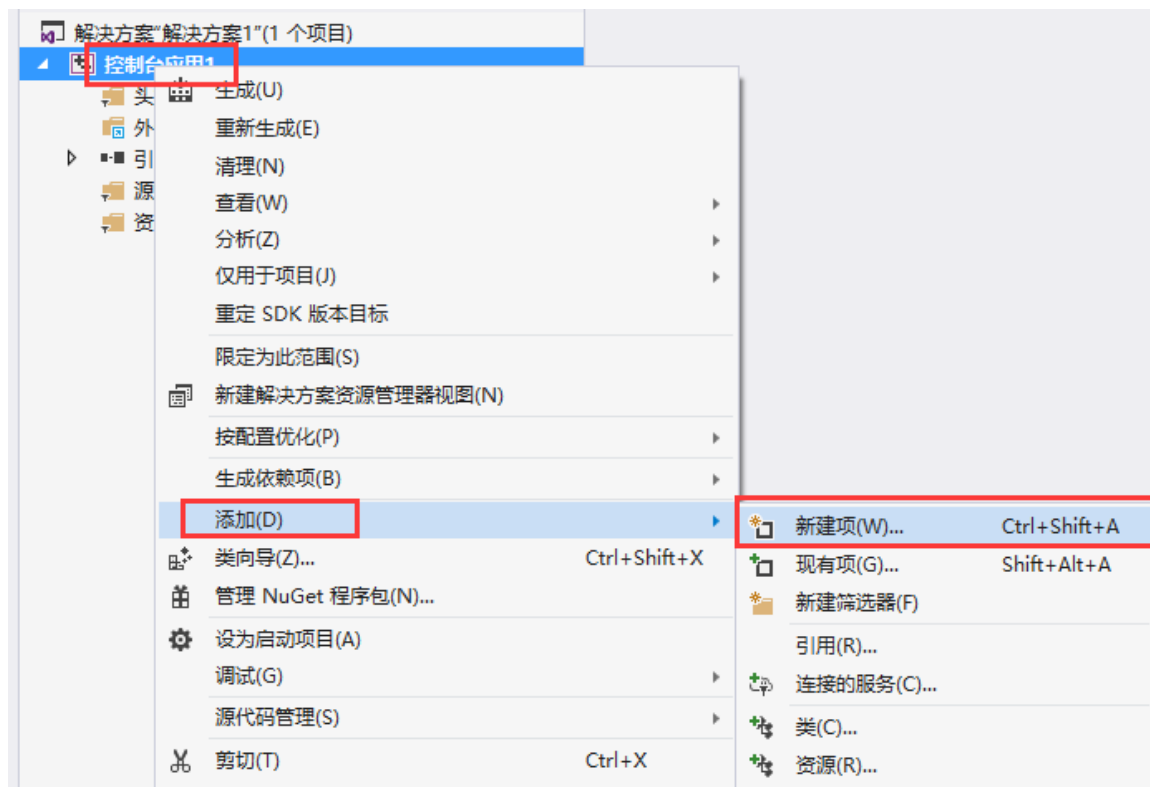


至此，我们完成了解决方案和工程的创建。其**逻辑关系**是：解决方案代表一个整体方案（如一个软件系统），该方案下可以包含若干个工程，每个工程代表一个程序或者库文件等（如软件系统保护的各个模块）。在 VS 中编写一个程序，实际是完成一个工程。因此，我们需要事先创建好解决方案，才能在里面开发一个工程。解决方案的文件为*.sln（solution），工程文件为*.vcxproj。**注意：这个目录结构是 VS 自动创建的，最好不要随意移动、重命名里面的文件或文件夹位置，避免破坏工程结构。**

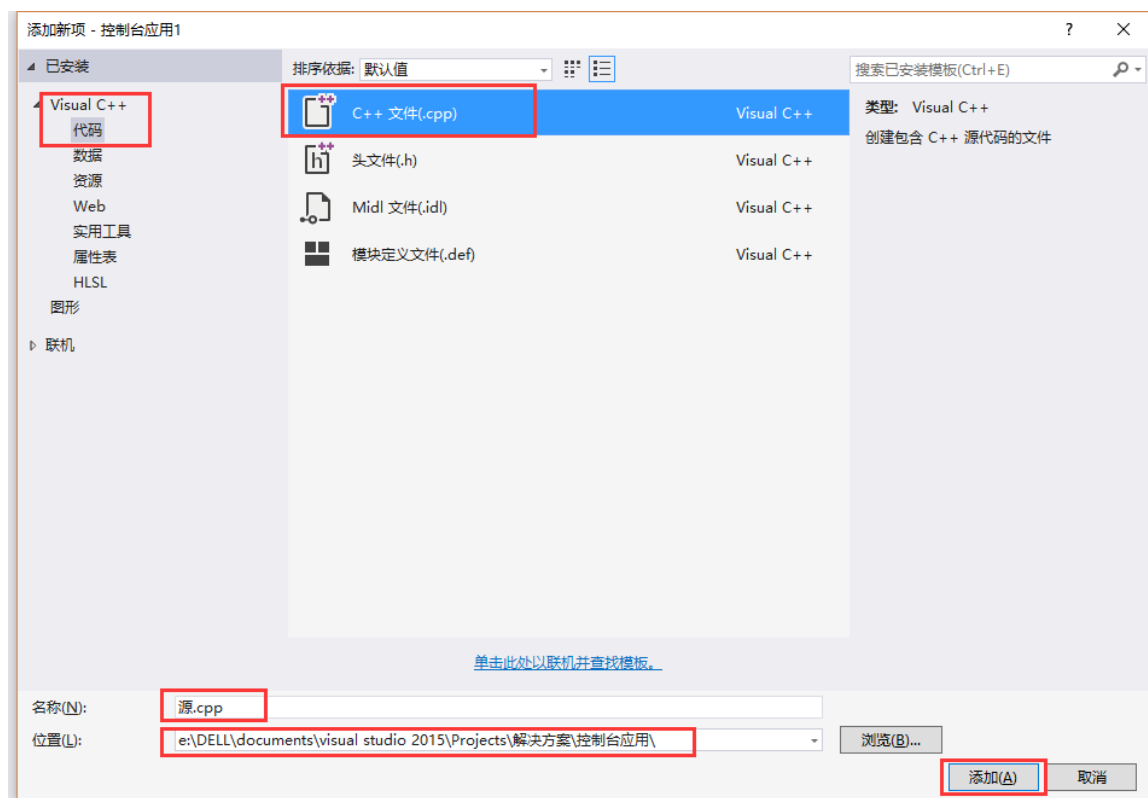


练习 2：添加文件

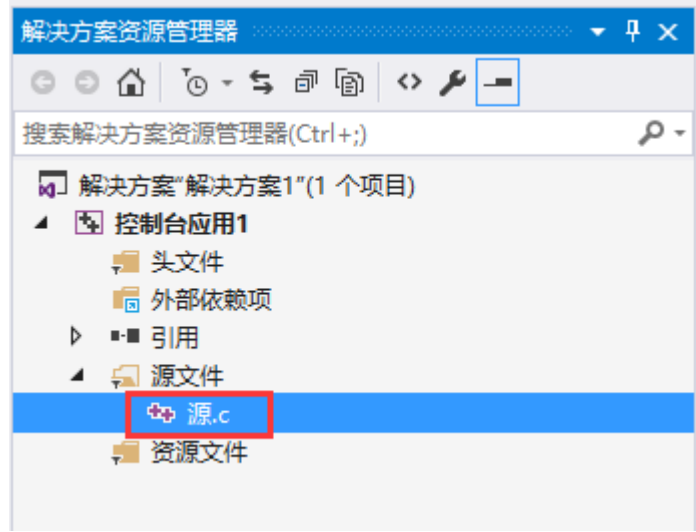
在练习 1 创建好的工程“你的学号_c”上右键单击，添加，新建项。



在弹出对话框中输入名称“你的学号_c.c”，注意后缀名 c 告诉 VS 这是一个 C 语言源程序，而非 C++ (*.cpp) 源程序。点击添加，完成文件创建。



在左侧“解决方案资源管理器”中可以看到添加的 c 文件，双击打开编辑窗口，即可开始编写代码。



练习 3：编写 C 程序

我们知道，图像处理是一种基本的计算机技术。对于各种来源的图像，计算机需将图像读入内存，才可以进行相关处理，试用 C 语言完成该题。

已知：位图（Bitmap）是一种基本的图像格式，它以二进制形式存储，图像文件后缀名为*.bmp。一个位图文件的基本结构包括 4 个部分：

Windows 位图格式

1. 文件头
2. 信息头
3. 颜色表（调色板）
4. 数据（像素值）

或下图表示：

数据段名称	对应的Windows结构体定义	大小(Byte)
bmp文件头	BITMAPFILEHEADER	14
位图信息头	BITMAPINFOHEADER	40
调色板		由颜色索引数决定
位图数据		由图像尺寸决定

其中：文件头和信息头都由若干个不同的字段组成，在 C 语言中可以用两个结构体来定义，如下图所示：

文件头 (BITMAPFILEHEADER):

```
typedef struct tagBITMAPFILEHEADER {  
    WORD    bfType;  
    DWORD   bfSize;  
    WORD    bfReserved1;  
    WORD    bfReserved2;  
    DWORD   bfOffBits;  
} BITMAPFILEHEADER;
```

信息头 (BITMAPINFOHEADER)

```
typedef struct tagBITMAPINFOHEADER{  
    DWORD    biSize;  
    LONG     biWidth;  
    LONG     biHeight;  
    WORD     biPlanes;  
    WORD     biBitCount;  
    DWORD    biCompression;  
    DWORD    biSizeImage;  
    LONG     biXPelsPerMeter;  
    LONG     biYPelsPerMeter;  
    DWORD    biClrUsed;  
    DWORD    biClrImportant;  
} BITMAPINFOHEADER;
```

关于一个图像的重要元数据，如红框标识的图像宽度 (*biWidth*)、高度 (*biHeight*) 等，都存储在这两个结构体当中。读取了这两个结构体，就可以知道图像的基本属性。

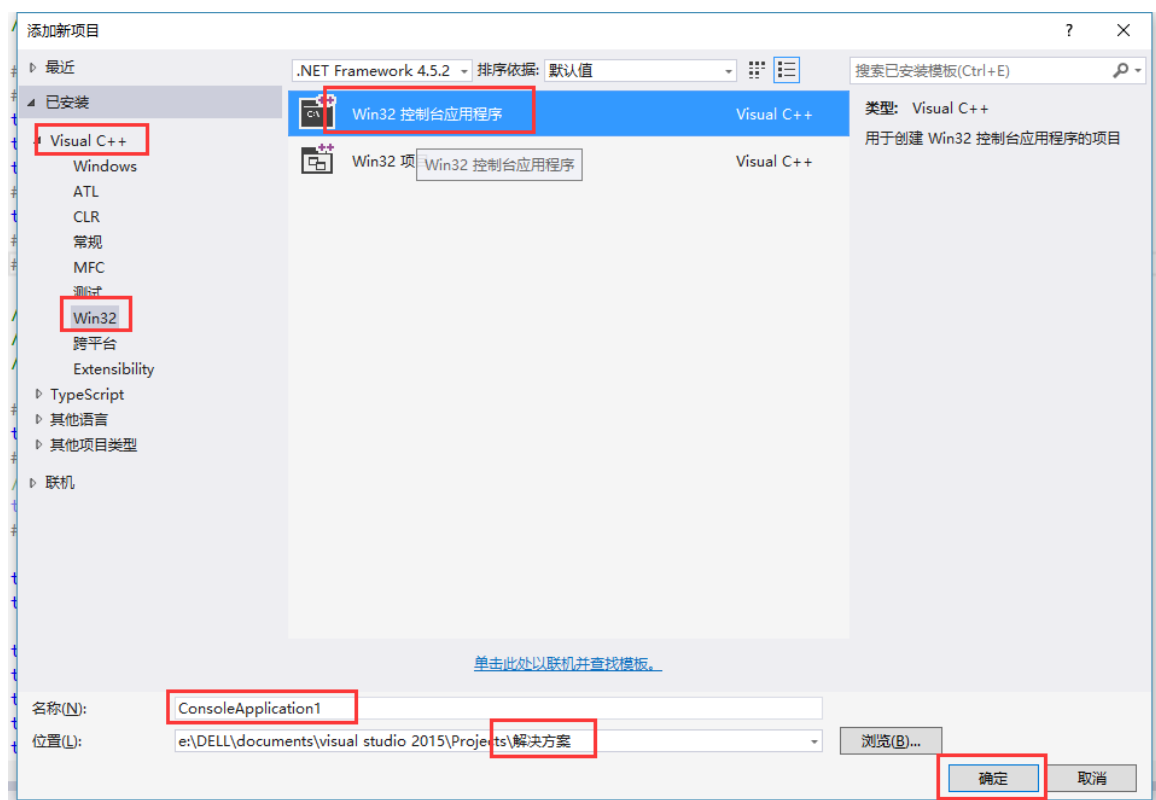
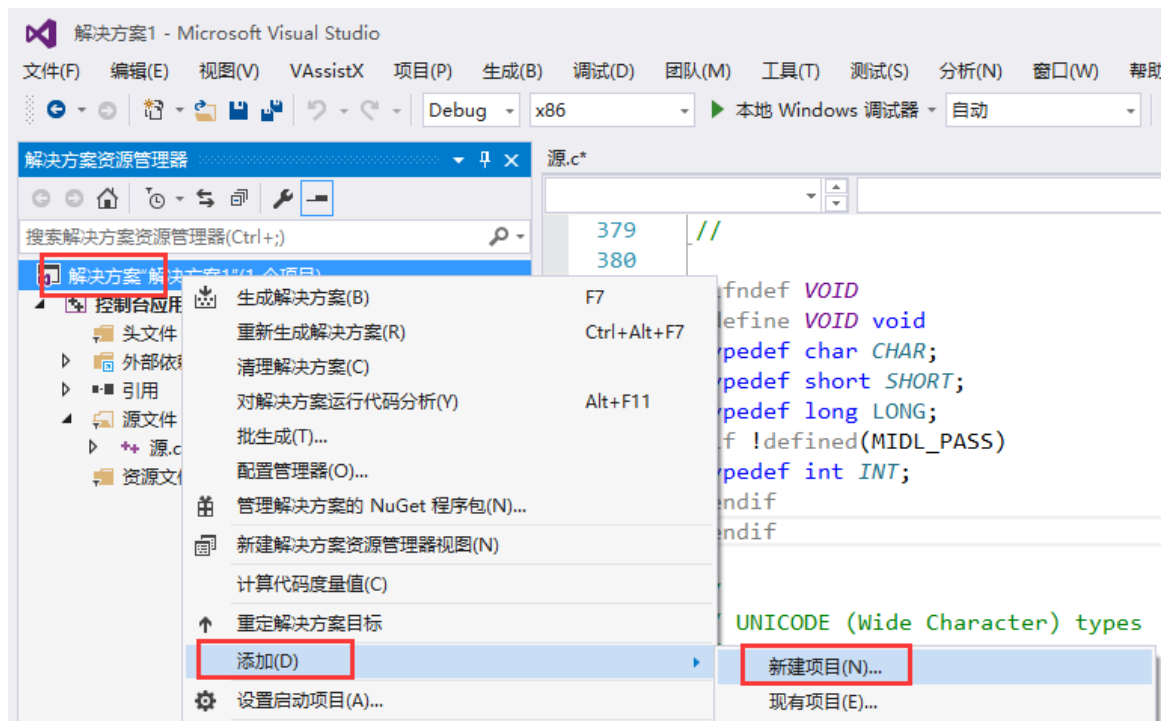
要求: 按照上述说明，补充练习 2 的“你的学号_c.c”文件，实现“bjfu_gray.bmp”图像宽度、高度和总像素数的屏幕输出。

参考: “image_c.c” 文件给出了参考示范但不完整，可按它的写法实现要求的功能。

练习 4：编写 C++ 程序

参考第一章 PPT，将练习 3 的 C 程序改写为 C++ 程序。步骤：

首先，在解决方案（注意不是工程）中添加一个新的工程。



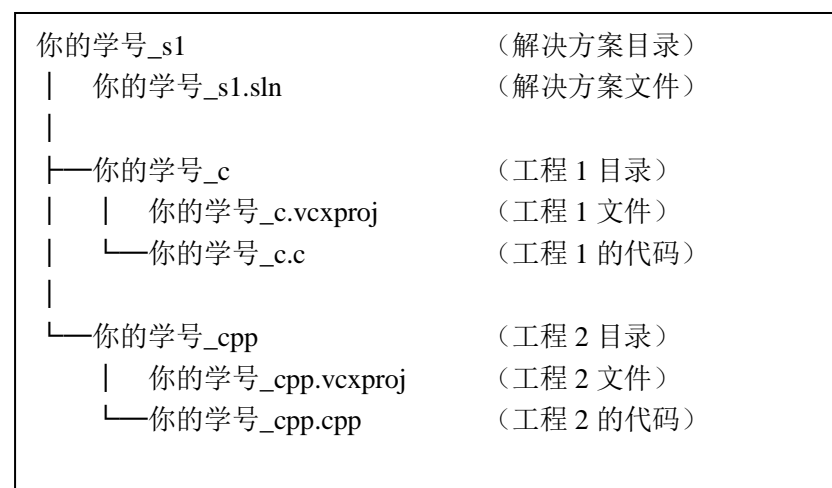
类型为“Win32 控制台应用程序”，将名称修改为“你的学号_cpp”，注意检

查位置是否自动选定为当前解决方案的目录。点击确定、下一步、勾选“空项目”、完成。此时，解决方案“你的学号_s1”中出现了两个并列的工程“你的学号_c”和“你的学号_cpp”。

按照练习 2 的方式在“你的学号_cpp”中添加一个源程序文件“你的学号_cpp.cpp”（注意后缀名为 cpp 了）。

最后，在“你的学号_cpp.cpp”中重写“你的学号_c.c”的程序，实现相同功能。要求使用 C++ 的输入输出流和 string 字符串。string 字符串如何转 C 字符串？请通过网络搜索。

此时，你得到一个类似这样的目录结构，清晰展示了解决方案和工程的结构逻辑，其他系统自动生成的文件和目录不包括在内：



练习 5：调试程序

在练习 3 或 4 的程序代码中，使用 F9 或者菜单“调试-切换断点”插入断点，然后 F5 或菜单“调试-开始调试”启动调试运行。

程序运行到断点处被中断时，使用下方“自动窗口”或“局部变量”窗口查看变量的值。在程序读入图像文件信息后，查看 BITMAPFILEHEADER 和 BITMAPINFOHEADER 结构体中各个元素的值，认识图像的相关的属性。

使用 F10、F11 或菜单“调试-逐过程/逐语句”，一步一步跟踪程序运行，观察程序执行情况是否与期望一致。

更多调试方法和技巧可参考网络资源或其他参考资料：

<https://blog.csdn.net/shu15121856/article/details/65494048/>

实验考核：

1、实验完毕后上交实验报告，实验报告模板从 FTP 中下载，实验报告的内容包括实验目的、实验内容和结果分析，实验报告一律写成 Word 格式文档（**.doc/.docx**）。

2、将调试好的程序源代码和程序的输入输出数据的情况附到实验报告中，并对程序的数据结果进行分析，说明基本运行机理。

3、实验报告重点总结你的心得体会，遇到了什么问题，怎么解决的，对以后有什么帮助。

4、上传电子版实验报告和源程序（删除 Debug、Release 目录和其他无用中间文件，仅保留解决方案、工程等文件和程序代码文件）到 FTP 服务器 homework 目录下对应课程、班级目录中。多个文件请压缩打包，请注意文件命名格式。

5、文件打包命名为“你的学号_姓名_s1”。

6、鼓励提问、交流、讨论，严禁相互复制、抄袭，违者按 0 分处理。