大一初次接触编程的时候，我使用编译器Visual Studio 2010，后来为了与时俱进，改用编译器Visual Studio 2019，新版本的优点是自动地对代码进行高亮与补全，但是很多以前正常的代码在Visual Studio 2019中运行报错，比如scanf()函数需要用scanf\_s()函数代替。同样的，此次使用老师提供的模板，编译器报错：

*不能将 "const wchar\_t \*" 类型的值分配到 "LPTSTR" 类型的实体*

新版本编译器不支持这样的类型转换，第一种方法是将所有涉及到这种类型转换的语句都修改，该方法可行性不强，因为用的是老师提供的模板，代码量大，是个黑盒，且不说能不能全部改过来，修改之后会不会影响整体结构产生新的问题，不得而知。第二种方法是去思考，这种报错是否是因为新版本编译器语法规则改变或者语法检查加强导致，如果真是如此，那么应该可以通过修编译器的全局属性解决。因此我联想到：MSVC在每个版本会进行了符合性改进和bug修复。因此，我进入调试->%当前项目名%属性->C/C++->语言，将“符合模式”改为“否”，再次运行，即不报错。

同时需要注意，Visual Studio 2019无法直接创建Win32应用程序，而是选择Windows桌面向导，修改项目名称和项目位置后，创建桌面项目，选择应用程序类型为桌面应用程序，空项目打勾，点击确定即可，创建完毕后再如图所示修改为Win32.

