



桂林电子科技大学
GUILIN UNIVERSITY OF ELECTRONIC TECHNOLOGY



人工智能学院
SCHOOL OF ARTIFICIAL INTELLIGENCE

第一讲 面向对象编程与C++

赵彬

zhaobin@guet.edu.cn

1.1 关于课程形式

理论（40课时）+实验（16课时-孙绍华老师）

目的

掌握基于C++语言的面向对象开发方法，能独立开发应用软件。

作用

毕业/课程设计、竞赛活动、毕业工作（大厂就业热门语言之一）、深造（初试/复试常见语言之一，平时科研）

考核

平时成绩+实验成绩+大作业+考试成绩（理论+机试）=总成绩 比例待定

要求

100% 出勤率（随机点名两次不到，重修，有重要事项可以请假）

1.2 面向对象编程之概念

面向对象编程 (Object Oriented Programming, OOP) 是一种计算机编程架构。OOP的一条基本原则是计算机程序由单个能够起到子程序作用的单元或对象组合而成。OOP=对象+类+继承+多态+消息, 其中核心概念是**类**和**对象**。

什么是类? 什么是对象?

类是对现实世界的抽象, 包括表示静态属性的数据和对数据的操作, **对象**是**类**的实例化。对象间通过消息传递相互通信, 来模拟现实世界中不同实体间的联系。在面向对象编程中, **对象**是组成程序的基本模块。

1.2 面向对象编程之概念

类(Class)和对象(Object)示例:



类 (class) 定义了一件事物的抽象特点。类可以为程序提供模版和结构。一个类的方法和属性被称为“成员”。

对象是类的具体实例(instance), 一个对象有状态、行为和标识三种属性。

1.3 面向对象与面向过程

面向对象(Object Oriented)是把构成问题事务分解成各个对象，建立对象的目的不是为了完成一个步骤，而是为了描述某个事物在整个解决问题的步骤中的行为。

面向过程(Procedure Oriented)就是分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候一个一个依次调用。

例：洗衣机里面放有脏衣服，怎么洗干净？

1.3 面向对象与面向过程

例：洗衣机里面放有脏衣服，怎么洗干净？（续）

1. 执行加洗衣粉方法
2. 执行加水方法
3. 执行洗衣服方法
4. 执行清洗方法
5. 执行烘干方法

- 1、确定两个对象：“洗衣机”对象和“人”对象
- 2、针对对象“洗衣机”加入一些属性和方法：“洗衣服方法”“清洗方法”、“烘干方法”
- 3、针对对象“人”加入属性和方法：“加洗衣粉方法”、“加水方法”
- 4、然后执行
人.加洗衣粉
人.加水
洗衣机.洗衣服
洗衣机.清洗
洗衣机.烘干

1.4 面向对象编程的目的

- 利用计算机解决现实世界中的问题，就是要将现实世界的问题经过抽象转换为计算机程序或软件，而**现实世界本身是由对象所组成**；
- 用对象来描述问题比用功能来描述问题更自然、更完整、更准确；
- 软件所要解决的**各种现实问题本身就是由各种对象所组成**，而且相对于功能的变化，对象更稳定；

1.5 面向对象的编程语言

Java是一种广泛使用的计算机编程语言，拥有跨平台、面向对象、泛型编程的特性，广泛应用于企业级Web应用开发和移动应用开发。

C#是微软推出的一种基于.NET框架的、面向对象的高级编程语言。C#以.NET框架类库作为基础，拥有类似Visual Basic的快速开发能力。

Python是一种面向对象的动态类型语言，它是解释型、高级编程、通用型编程语言，由吉多·范罗苏姆创造，第一版发布于1991年。

Go（又称Golang）是Google开发的一种面向对象的静态强类型、编译型、并发型，并具有垃圾回收功能的编程语言。

C++是一种使用广泛的计算机程序设计语言。它是一种通用程序设计语言，支持多重编程模式，例如过程化程序设计、数据抽象、面向对象程序设计、泛型程序设计和设计模式等。

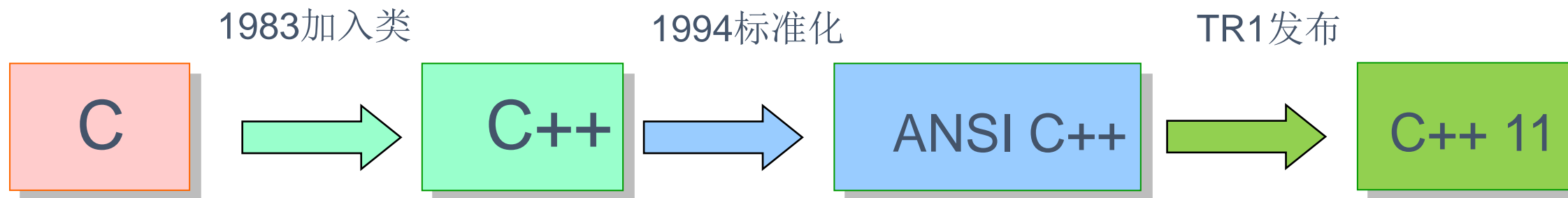
1.6 C++的发展

伴随着C的推广及暴露的缺点！

- C语言对数据类型检查的机制比较弱，缺少支持代码重用的结构；
- 随着计算机性能的飞速提高，软件规模的扩大，很多软件的体积都超过 1G，用C语言开发这些软件就显得非常吃力；
- C语言是一种面向过程的编程语言，已经不能满足运用面向对象的方法开发软件的需要；
- 为克服C语言本身存在的缺点，同时为支持面向对象的程序设计，在C语言基础上创建、开发出C++。

1.6 C++的发展

C++语言的产生



1.6 C++的发展

- 1983年8月，C with class首次投入使用，开天辟地；
- 1983年12月，C with class更名为C++，加入了很多C语言所没有的特性，包括虚函数，操作符重载，常数、用户可控制的自由空间存储区，新的类型检查和注释风格；
- 1985年2月，C++ Release 1.0发布
- 1985年10月，巨著《The C++ Programming Language》第一版出版
- 1989年，C++ Release 2.0发布
- 1990年7月，C++加入模板
- 1991年6月，C++加入异常处理
- 1993年3月，加入运行时类型识别
- 1993年7月，加入名字空间
- 1998年11月，C++ ISO标准得到批准
- 2003年，C++标准委员会发布处理缺陷报告，发布C++03
- 2005年，C++标准委员会发布TR1（Technical Report 1），目前，几乎所有流行的C++编译器都已经支持TR1
- 2008年，C++标准委员会发布N2800
- 2011年，C++的最新标准C++0x完成，加入了许多新特性，如Lambda表达式、智能指针share_ptr、右值引用、多任务内存模型等

1.7 C++与C

C++与C项目组织形式不同:

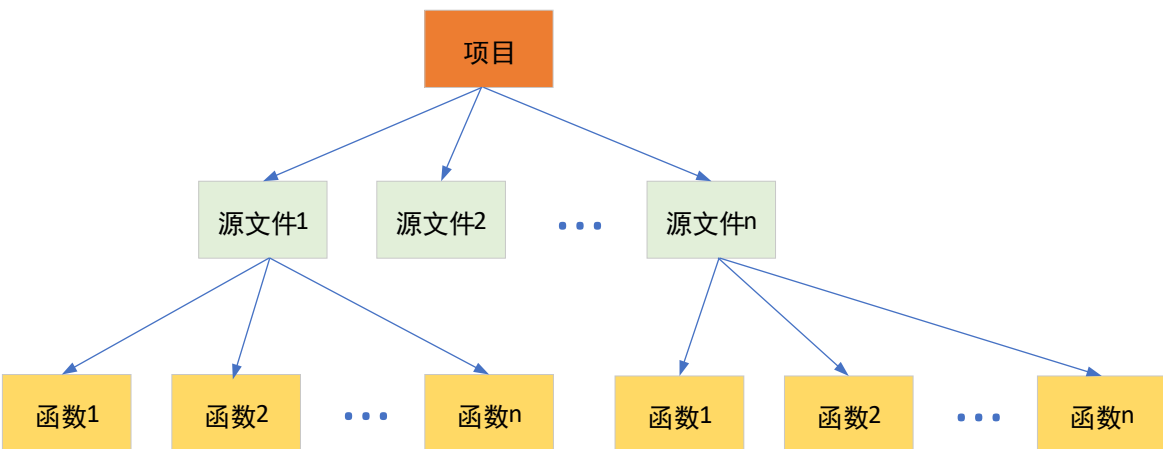


图1 基于C的项目组织形式

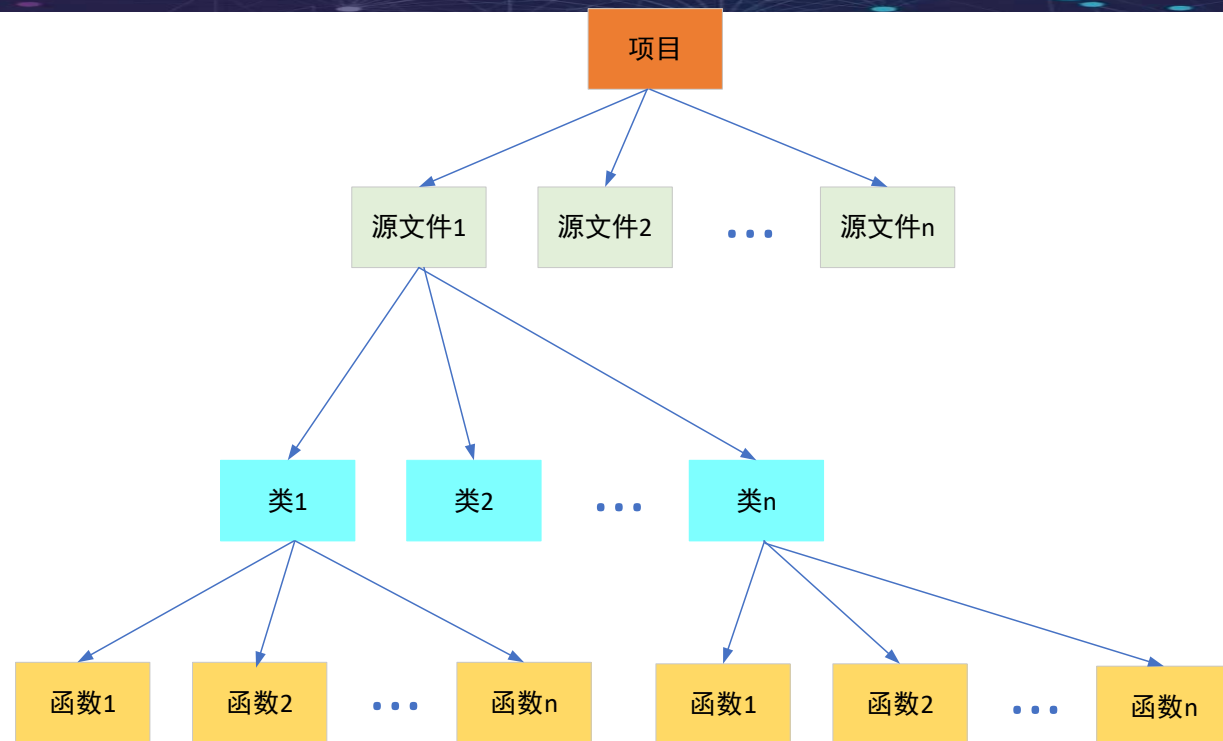


图2 基于C++的项目组织形式

C把重复使用或具有某项功能的代码封装成一个函数，将具有相似功能的函数放在一个源文件，调用函数时，引入对应的头文件就可以。

C++中多一层封装，就是类（Class）。类是由一组相关联的函数、变量组成。可以将一个类或多个类放在一个源文件，使用时引入对应的类。

1.7 C++与C

C++与C语言的代码格式不同：

- C语言：C语言源文件后缀为**.c**。C/C++ 使用相同的编译器，Windows 下默认有微软的 TC.exe，Linux 下一般是 GCC。**.c** 源文件会按照C语言的方式编译。很多C语言初学者创建源文件时使用的后缀为 **.cpp**，也完全没有问题，编译、链接、运行都顺利通过；
- C++：C++源文件后缀为**.cpp**，**.cpp** 源文件会按照C++的方式编译。C++几乎完全兼容C语言，它们类似子集（C语言）和超集（C++）的概念

1.8 C++的特点

- 保持与C兼容

C++集成环境约定：当源程序文件的扩展名为.c时，则为C程序；而当源程序文件的扩展名为.cpp时，则为C++程序。

- C++语法灵活，功能强大

C++设计无需复杂的程序设计环境，对语法限制比较宽松；缺点是由于对语法限制比较宽松，编译时许多逻辑上的错误不容易发现。C++的很多特性都是以库（如STL）的形式提供的。

- 面向对象的机制

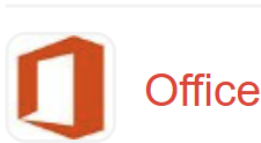
C++是一种面向对象的，使得开发人机交互类型的应用程序更为简单、快捷。程序的各个模块独立性更强，可读性和可移植性更强，代码结构更加合理，扩充性更强，这对于设计、编制和调试一些大型的软件尤为重要。

- 适合大型系统的开发

- 多样化风格

支持多种程序设计风格（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）。

1.9 C++的应用



- 系统编程

C++凭借其C的兼容，可以方便嵌入汇编语言，实现底层的调用，适合开发系统级软件，编写驱动程序等。

- 网络编程

在多线程、网络通信、分布应用、服务器端、客户端程序方面，C++有不可比拟的优势。C++拥有很多成熟的用于网络通信的库服务端开发，很多互联网公司的后台服务器程序都是基于C++开发的，而且大部分是linux，unix等类似操作系统。

- 服务器端开发：很多互联网公司的后台服务器程序都是基于C++开发的。

- 嵌入式开发

低端嵌入式开发主要是基于汇编语言和C语言，中端嵌入式开发主要是使用C和C++。

- 游戏工具开发

目前很多游戏客户端都是基于C++开发的，除了一些网页游戏，C++凭借先进的数值计算库、泛型编程等优势，在游戏领域应用非常广泛。

1.10 C++的开发环境

● Turbo C

使用集成的开发环境，采用一系列下拉式菜单，将文本编辑、程序编译、连接以及程序运行一体化。

● Visual Studio

Visual Studio 是一套完整的开发工具集，用于生成 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Basic、Visual C++、Visual C# 和 Visual J# 全都使用相同的集成开发环境 (IDE)。

● C++ Builder:

具有快速的可视化开发环境，专业C++开发环境所能提供的全部功能。

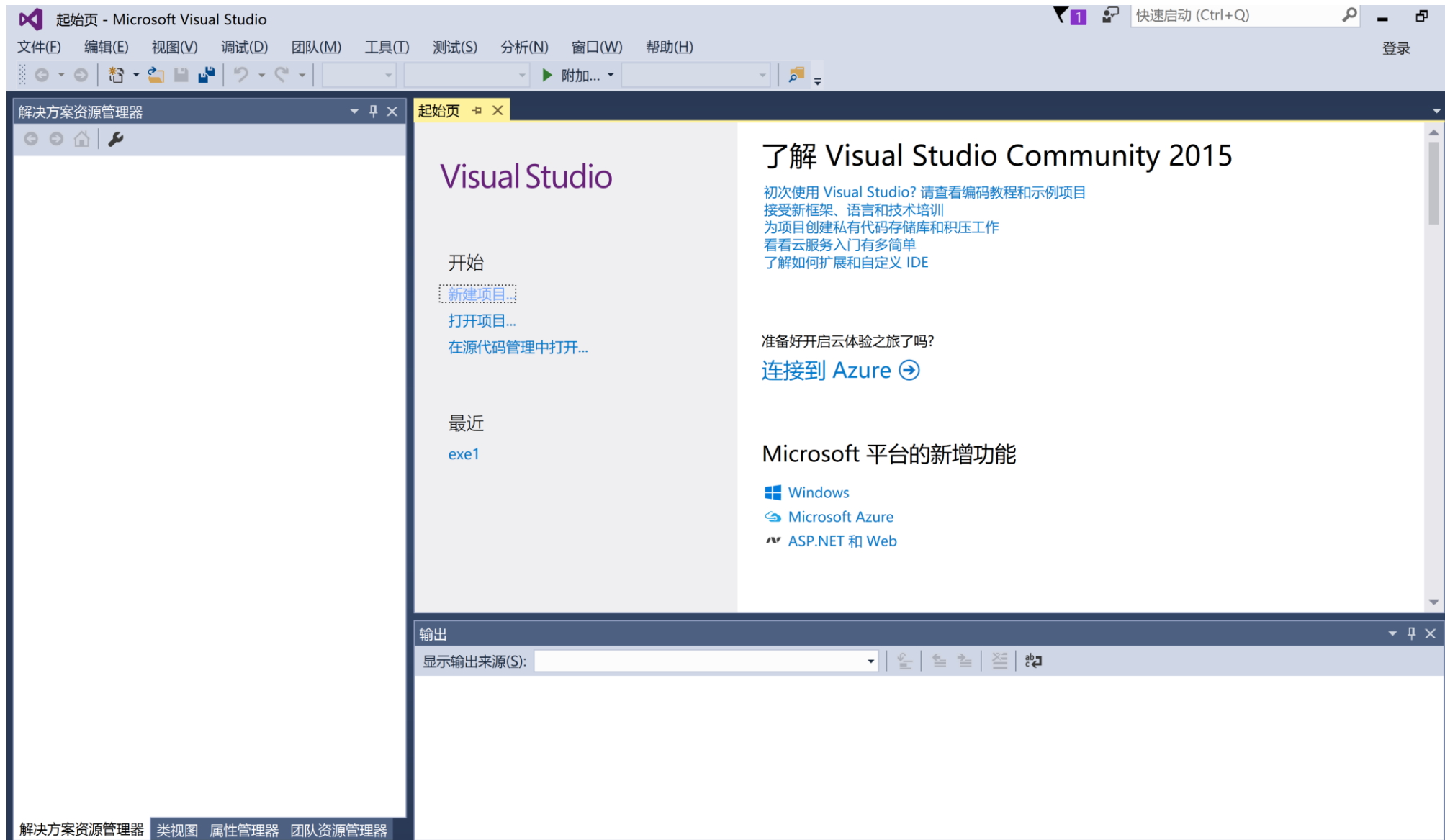
● VC++

交互式、可视化集成开发软件，不仅支持C++语言，也支持C。集程序的编辑、编译、连接、调试等功能于一体。

● Eclipse

开放源代码的、基于 Java 的可扩展开发平台。允许他们构建与 Eclipse 环境无缝集成的工具。Eclipse 中的每样东西都是插件，对于给 Eclipse 提供插件，以及给用户提供一致和统一的集成开发环境而言，所有工具开发人员都具有同等的发挥场所。

1.11 Visual Studio 2015



1.11 Visual Studio 2015

● 文件菜单

所有跟文件有关的操作，如新建，添加文件，项目，解决方案等都在文件菜单里面进行等。

● 视图菜单

当窗口关闭后，可以通过这里重新打开视图窗口，C++语言学习常用的视图有：

- ① 代码：编写代码；
- ② 解决方案资源管理器：管理项目文件；
- ③ 错误列表：查看警告，错误和系统消息；
- ④ 输出：编译，链接过程，以及这个过程中产生的一些消息；

1.11 Visual Studio 2015

- **生成菜单：执行跟编译相关的操作；**

① **解决方案：**对解决方案的操作，里面的所有的项目都会跟着执行。

清理解决方案：删除掉编译过程产生的所有的临时文件，包括最后生成的文件

生成解决方案：对修改了的内容进行重新的编译，链接操作；

重新生成解决方案：相当于先执行-清理解决方案，后在执行生成解决方案；

② **项目：**项目里面的所有的文件都会跟着执行；

“生成exe1”“重新生成exe1”，“清理exe1”的范围只针对当前项目；

③ **编译：**主要针对于单个文件而言

可以对单个文件进行编译，速度很快，当项目很大的时候，检查单个文件的语法错误时，经常会用到；

- **调试菜单：该功能使用非常多，是项目开发用的最重要的功能之一**

① **断点窗口：**查看当前项目的添加的断点的情况，方便统一管理；

② **自动窗口：**自动的显示当前正在使用的一些变量的值；

③ **监视窗口：**可以查看指定的变量的值；

④ **内存窗口：**进行大数据开发时，如图形图像处理用的非常多；

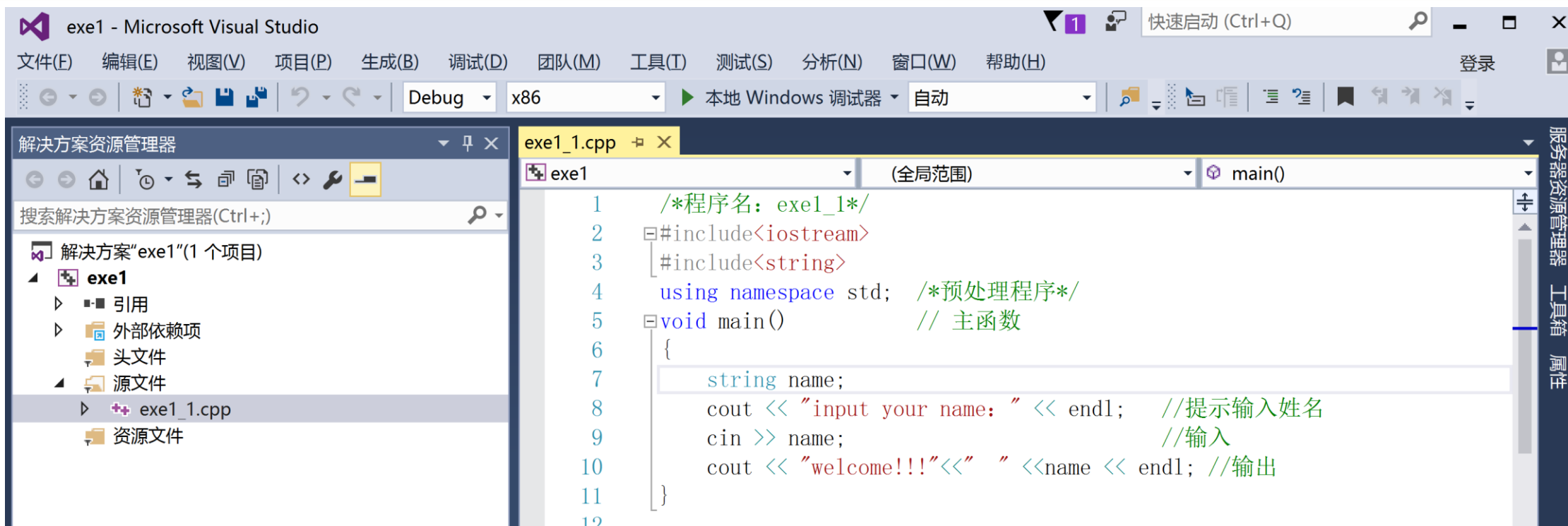
1.11 Visual Studio 2015

- 选项窗口：

该窗口位于工具菜单->选项里面，这里面的参数的设置针对的对象是Visual Studio 这个软件，一旦修改后将所有的解决方案都起作用，如下图是增加c/c++语言的文件的行号。

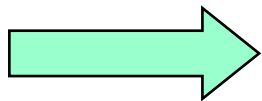
- 项目属性页：

exe1属性页通常用来设置，该项目相关的编译，链接等项目属性，该属性的设置只针对该项目有效。



1.12 编写过程与示例代码

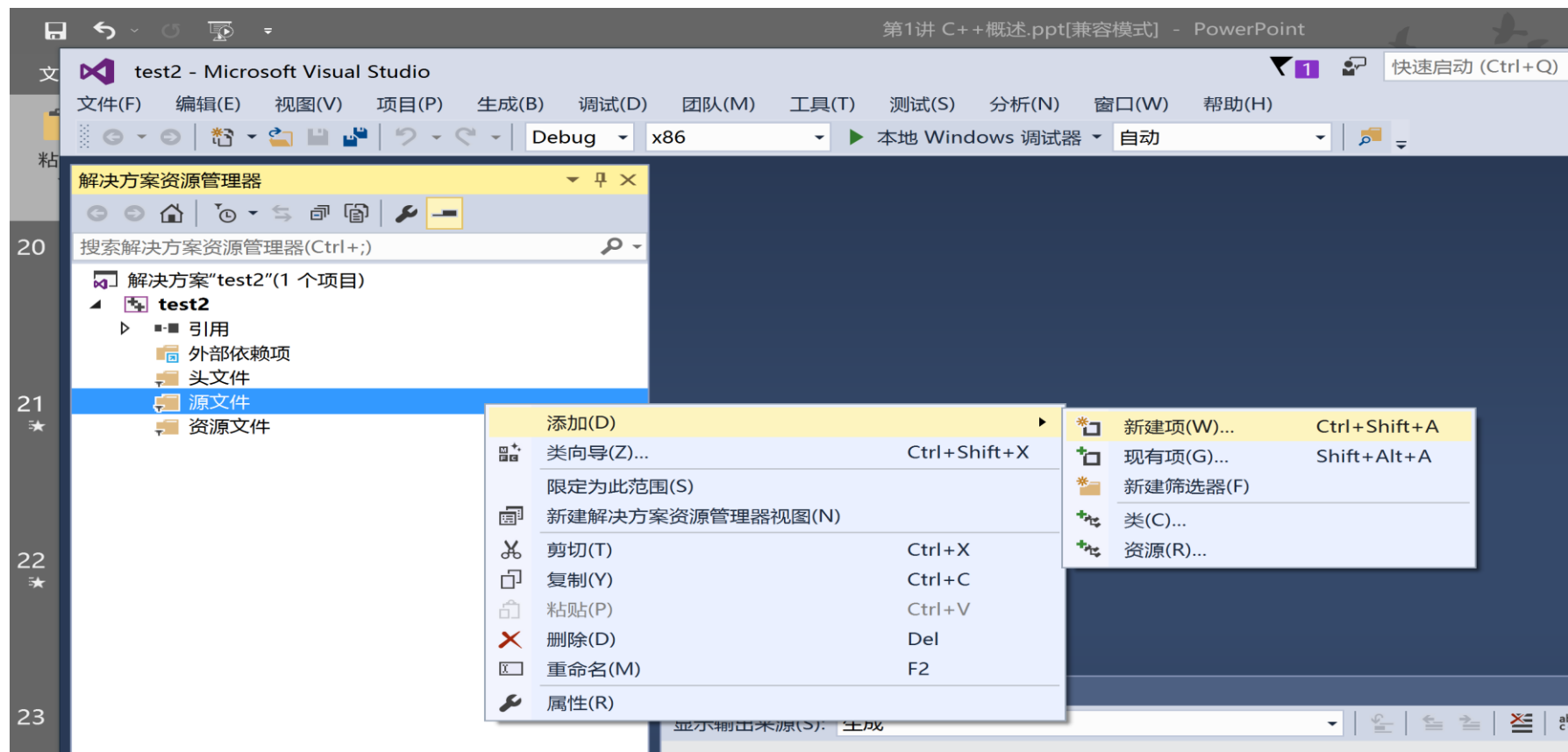
打开“解决方案
资源管理器”



建立“win32控
制台应用程序”




添加“源文件”
建立.cpp文件



1.12 编写过程与示例代码

```
1.      /* 程序名: exe1_1 */           ← 注释块
2.      #include <iostream>             /*预处理文件*/
3.      #include<string>                ← 预处理命令
4.      using namespace std;
5.      void main()                    ← 主函数 主函数
6.      {                               ← 函数开始
7.          string name;
8.          cout<<"input your name:" <<endl;           //提示输入姓名 ← 注释行
9.          cin>>name;                                   //输入
10.         cout<<" welcome!!! " <<name<<endl;          //输出
11.     } ← 函数结束                                ← 语句结束标志
```

运行结果



```
C:\Windows\system32\cmd.exe
input your name:
GUET
welcome!!! GUET
请按任意键继续. . .
```

1.13 std: 标准命名空间

Namespace（命名空间）：标准c++中的一种机制，使用它可以在不同的空间内使用同名字类或者函数，用来控制不同类库相同名字的冲突问题。

```
namespace 命名空间名  
{  
  
    // 命名空间成员  
  
}
```

使用标准命名空间std有3种方式：

- ✓ 整体声明 `using namespace std;`
- ✓ 使用时单个对象临时声明 `std::cout<<"Hello C++"<<std::endl;`
- ✓ 单个对象使用前声明 `using std::cout;`

1.13 std: 标准命名空间

- 整体声明

```
#include <iostream>
using namespace std;
int main( )
{  cout << "Hello
    C++!"<<endl;
    return 0;
}
```

- 单个对象临时声明

```
# include <iostream>
int main ( )
{  std::cout<<"Hello
    C++!"<<std::endl;
    return 0;
}
```

- 单个对象使用前声明

```
# include <iostream>
using std::cout;
using std::endl;
int main ( ) {
    cout<<"Hello
    C++!"<<endl;
    return 0;
}
```

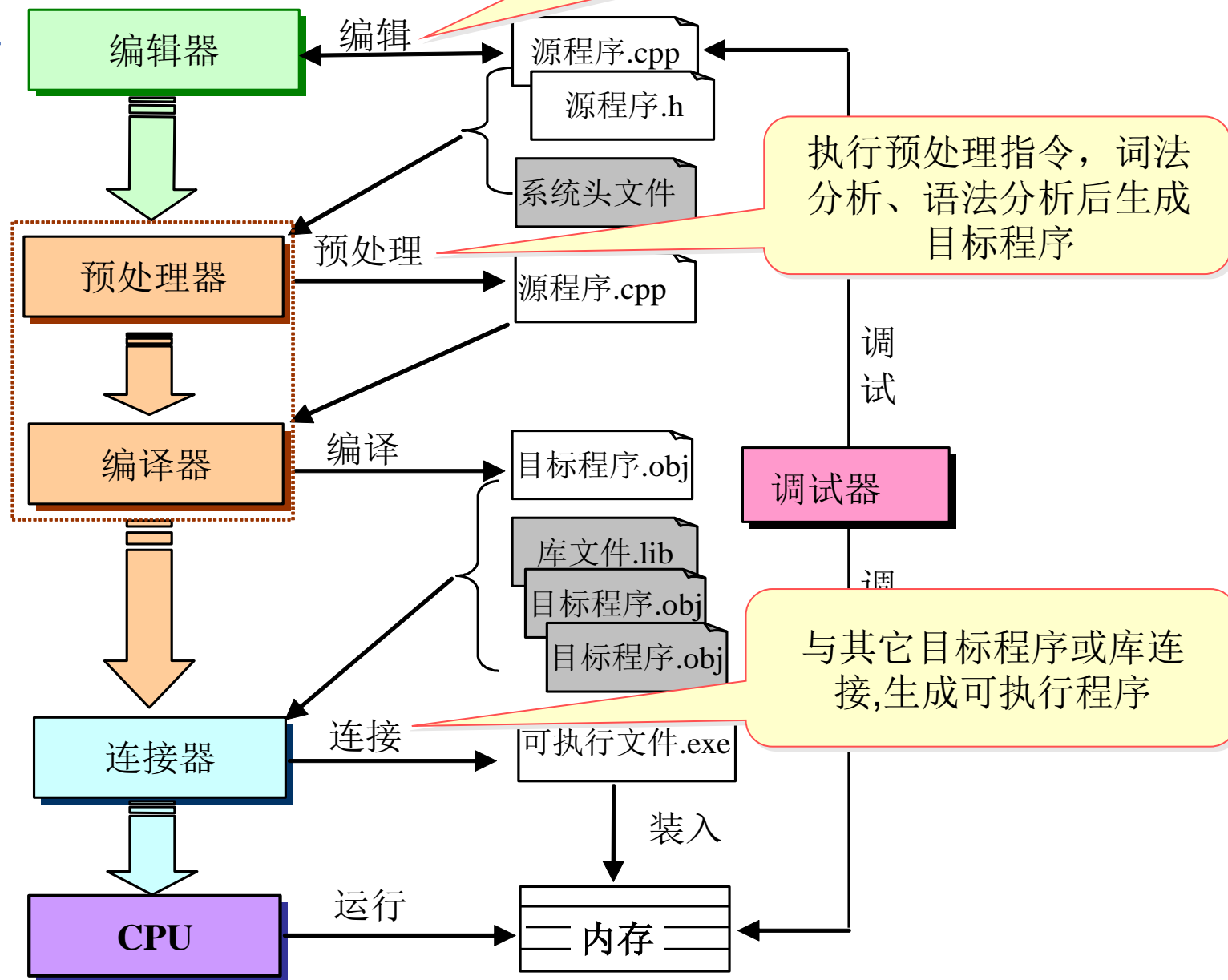
程序代码的录入、修改

执行预处理指令，词法
分析、语法分析后生成
目标程序

调试器

与其它目标程序或库连
接,生成可执行程序

1.14 C++程序开发步骤



1.14 C++程序开发步骤

编译预处理

C++提供的预处理功能主要有宏定义命令、文件包含命令、条件编译命令3种，主要处理#开始的预编译指令，如宏定义(#define)、文件包含(#include)、条件编译(#ifdef)等。这些命令以符号“#”开头，而且末尾不包含分号。

(1) 宏定义

宏定义的作用是实现文本替换。有两种格式：不带参数的宏定义和带参数的宏定义。

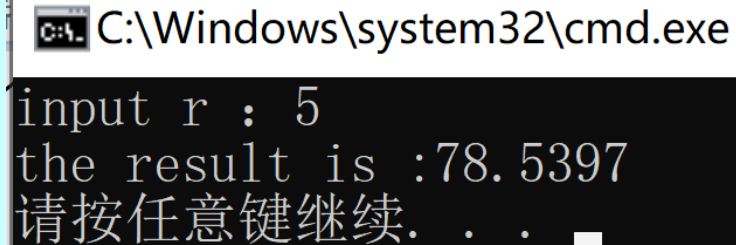
不带参数的宏定义

格式: #define 宏名 字符串

【例1-2】 利用宏表示圆周率，求圆面积。

```
/* 程序名：exe1_2 */  
#include <iostream>  
#define PI 3.14159 //宏定义  
using namespace std;  
void main( )  
{ double s,r;  
  cout<<"input r : ";  
  cin>>r;  
  s=PI*r*r;  
  cout<<"the result is :"<<s<<endl;  
}
```

运行结果:



```
C:\Windows\system32\cmd.exe  
input r : 5  
the result is :78.5397  
请按任意键继续. . .
```

带参数的宏定义

格式: **#define** 宏名(参数列表) 字符串

【例1-2】 利用宏表示圆周率, 求圆面积。

```
/* 程序名: exe1_3 */  
#include <iostream>  
#define PI 3.14159  
#define S(r) PI*r*r  
using namespace std;  
void main( )  
{ double s,r;  
  r=3.0;  
  s=S(r);  
  cout<<"the result is :"<<s<<endl;  
}
```

运行结果:

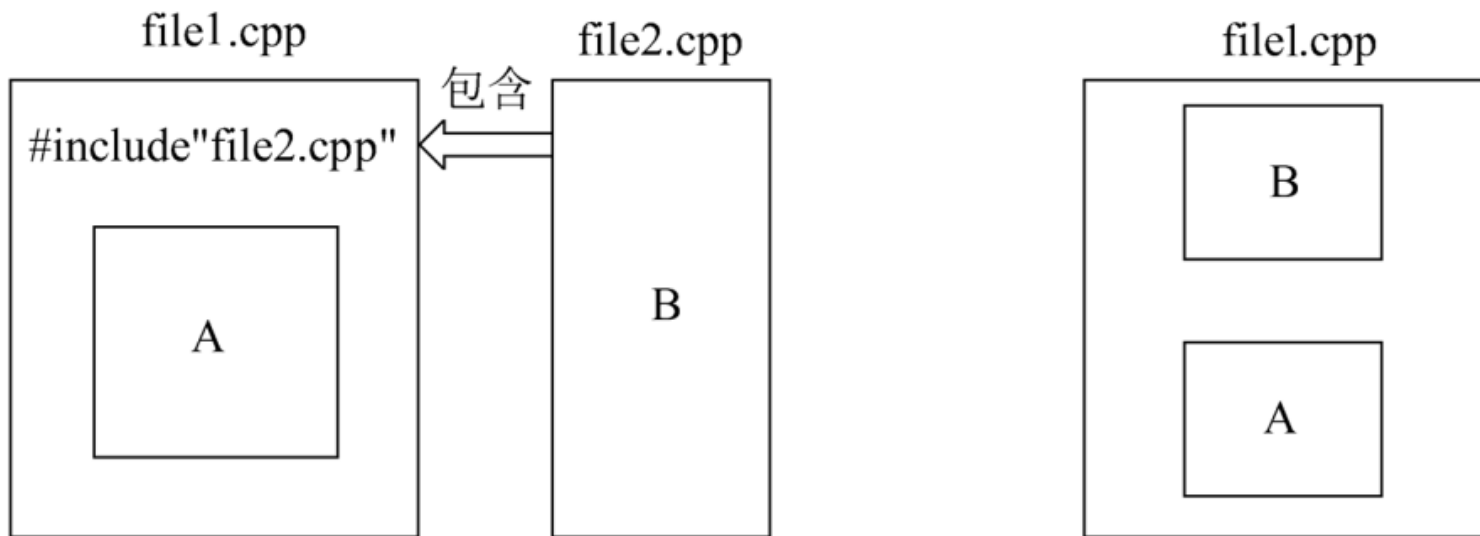
C:\Windows\system32\cmd.exe

the result is :28.2743
请按任意键继续. . .

(2) 文件包含

文件包含include命令的作用是将一个文件的全部内容包含到本文件中，有如下两种形式：

```
#include <文件名>  
#include "[路径\]文件名"
```



(3) 条件编译

一般情况下，源程序中所有的行都应参加**正式编译**。但有时候希望在**一定条件下编译某些行**，在其它情况不编译这些行。这时候就需要在预处理阶段作**条件编译处理**。

ifdef、else和endif

```
#ifdef 标识符  
    程序段1  
#else  
    程序段2  
#endif
```

ifndef、else和endif

```
#ifndef 标识符  
    程序段1  
#else  
    程序段2  
#endif
```

if、elif和else

```
#if 常量表达式1  
    程序段1  
#elif 常量表达式2  
    程序段2  
#elif 常量表达式3  
    程序段3  
.....  
#else  
    程序段n+1  
#endif
```

1.15 Visual Studio 2015常见问题

- 问题1：调试过程中Debug未输出结果就闪退。

解决思路：项目——当前项目属性——链接器——系统——子系统（右侧）——将其设置为控制台——单击确定，然后按Ctrl+F5进行调试，解决闪退。

- 问题2：无法打开头文件。

解决思路：去掉头文件iostream.h的“.h”，原因在于VS2015库文件中只有iostream没有isotream.h。

- 问题3：编译程序不能识别cin， cout。

解决思路：在iostream的头文件下加入using namespace std;程序语句

- 在源码中遗失“; ” 错误信息: missing ‘;’
- 缺少命名空间定义 错误信息: ‘cout’: undeclared identifier
- 程序中使用中文标识符
错误信息: unknown character ‘0xa3’
- 变量在赋值之前使用
错误信息: local variable ‘a’ used without having been initialized
- 在一个工程中出现多个 main 函数
- 不能打开指定头文件
错误信息:
Cannot open include file: ‘R.....h’ No such file or directory

思考：

将上例中`#include <iostream>`改为
`#include <iostream.h>`是否可以运行？有什么不同？

`<iostream>`和`<iostream.h>`格式不一样，两者是两个不同的文件。`<iostream>`为了支持新的标准，`<iostream.h>`为了兼容以前的C++代码。

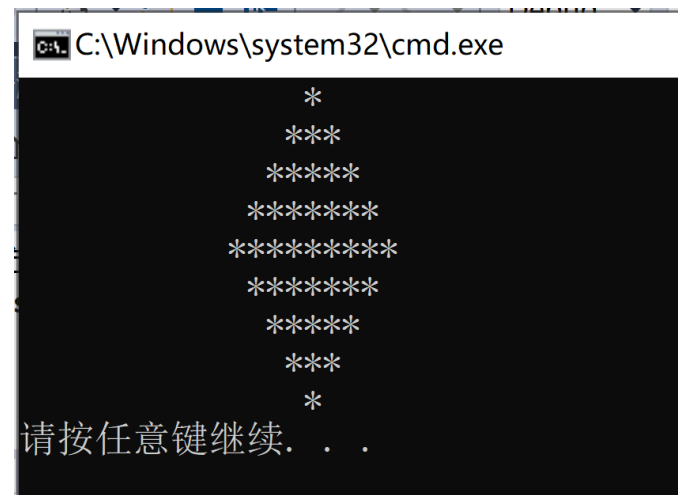
- C语言与早期的C++头文件后缀名为.h，如`<iostream.h>`。新的C++标准为了对C语言以及对早期c++文件的支持，附带了这些头文件。因此，当使用`<iostream.h>`时，相当于在c中调用库函数。
- `<iostream>`没有后缀，使用的时候，由于该头文件没有定义全局命名空间，必须使用`using namespace std`语句，这样才能正确使用cout

命名空间std封装的是标准程序库的名称，标准程序库为了和以前的头文件区别，一般不加".h"。

案例1 简单的图形输出程序

```
/* 程序名：lingxing */  
#include <iostream >  
#include <iomanip >  
using namespace std;  
int main()                //主函数  
{ cout<<setw(16)<<"*"<<endl;  
  cout<<setw(17)<<"***"<<endl;  
  cout<<setw(18)<<"*****"<<endl;  
  cout<<setw(19)<<"*****"<<endl;  
  cout<<setw(20)<<"*****"<<endl;  
  cout<<setw(19)<<"*****"<<endl;  
  cout<<setw(18)<<"*****"<<endl;  
  cout<<setw(17)<<"***"<<endl;  
  cout<<setw(16)<<"*"<<endl;  
  return 0;  
}
```

运行结果:

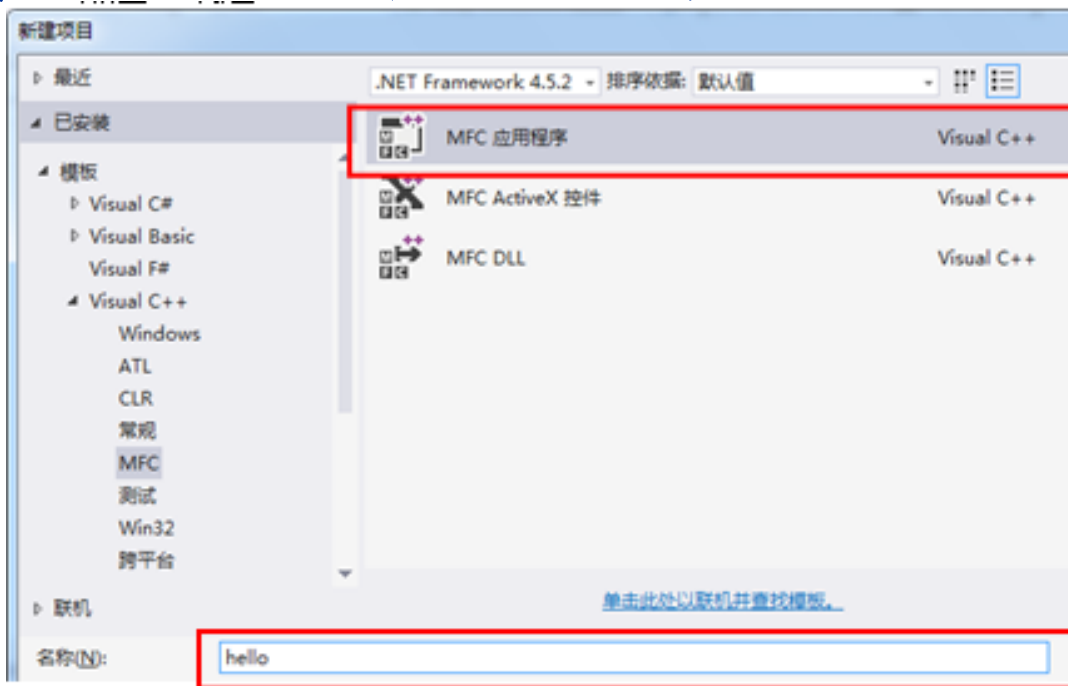


```
C:\Windows\system32\cmd.exe  
      *  
     ***  
    *****  
   *****  
  *****  
 *****  
*****  
 *****  
  *****  
   *****  
    *****  
     ***  
      *  
请按任意键继续...
```


程序说明：

- C++ 程序中的关键字（非自定义），如本程序中的main，float，cin，cout等**一律为小写英文字母**。
- C++ 程序中用到的特殊符号，如<、>、"、/、*、{、}等一律为英文符号，**不是全角中文符号**。
- C++ 由于面向对象的概念，输入输出将不再使用scanf和printf，而是使用输入输出流，其中是**cin输入流，负责接受键盘的输入，cout是输出流，负责输出到屏幕或外设等**。
- **setw(int n)用来控制输出间隔**。setw(16)表示其后面紧跟的输出占16个位置，不足的用空格填充。
- **iomanip: I/O流控制头文件**。io代表输入输出，manip是manipulator（操纵器）的缩写，类似于C里面的格式化输出，主要是对cin,cout之类的一些操纵运算，如setw，setbase，setprecision等。

案例2 MFC应用程序



应用程序类型:

☐ 单个文档(S)

☐ 多个文档(M)

☐ 选项卡式文档(T)

☒ 基于对话框(D)

☐ 使用 HTML 对话框(U)

☐ 无增强的 MFC 控件(U)

☐ 多个顶级文档(T)

☒ 文档/视图结构支持(V)

☒ 安全开发生命周期(SDL)检查(C)

资源语言(L):

中文(简体, 中国)

项目类型:

☒ MFC 标准(A)

☐ Windows 资源管理器(X)

☐ Visual Studio(U)

☐ Office(F)

视觉样式和颜色(C):

Windows 本机/默认

☐ 启用视觉样式切换(C)

MFC 的使用:

☒ 在共享 DLL 中使用 MFC(U)

☐ 在静态库中使用 MFC(C)

```
void ChelloDlg::OnBnClickedOk()
{
    // TODO: 在此添加控件通知处理程序代码
    MessageBox(_T("欢迎开始学习——MFC对话框程序"), _T("欢迎!"), MB_OK);
    CDialogEx::OnOK();
}
```

运行结果:



程序说明：

MFC是作为一个扩展轻量级的WindowsAPI，面向对象的C++封装库而引入的。应用程序类型主要有以下三种：

- 基于对话框视图架构：

主窗口是对话框类型，可以方便的使用控件，一般没有菜单栏、工具栏。

- 单文档视图架构：

只能进行一份文档或图片的操作，不能同时在一个程序打开两个文件。

- 多文档视图架构：

可以用多个窗口显示不同的信息，进行不同的任务，有多个视图环境，可以同时操作多个文件，多文档可产生子窗口。

1.16 第一讲小结:

1. C++特点: 与C兼容、语法灵活、功能强大、面向对象、适合大型系统开发、多样化风格。

2.C++编程的开发工具: Turbo C、Visual C++、Visual Studio、Eclipse等, 其中Visual studio集编辑、编译、连接、调试等功能于一体。

3.开发 C++程序的步骤为:

- ① 分析问题。分析需求, 确定解决方法, 并用适当的工具描述。
- ② 编辑程序。
- ③编译程序。编译源程序, 产生目标程序。文件的扩展名为.obj。
- ④连接程序。将一个或多个目标程序与库函数进行连接, 产生可执行文件。
- ⑤运行调试程序。

4.一个标准的C++源程序主要由三个部分组成: 注释部分、编译预处理部分和程序正文部分。

5.MFC应用程序的三种形式: 基于对话框视图架构、单文档视图架构、多文档视图架构。

本次作业

安装visual studio 2015, 并编写代码输出“My student number is 你的学号, I am 你的名字, I major in 你的专业”

Visual studio 2015 下载链接 <https://share.weiyun.com/SXZzpRuq>

安装使用说明: <https://blog.csdn.net/xucongyoushan/article/details/86028712>