实验一

第一部分：Visual Studio 2008开发环境应用入门（2学时）

* 1. 实验目的
     1. 了解Visual Studio 2008的特点。
     2. 熟悉Visual Studio 2008的开发环境。
     3. 学习用Visual Studio 2008编写标准的C++控制台程序。
  2. 实验任务
     1. 使用Visual Studio 2008建立一个非图形化的标准C++程序，编译、运行例2.1，即以下程序：

#include <iostream >

using namespace std ;

int main()

{

cout<<"Hello!\n";

cout<<"Welcome to c++!\n";

return 0;

}

* 1. 实验步骤
     1. 启动Visual Studio 2008开发环境
* 从“开始”菜单中选择“程序”| Microsoft Visual Studio 2008| Microsoft Visual Studio 2008，显示Visual Studio 2008开发环境主窗口。
  + 1. 创建一个项目
* 单击File菜单中的New选项中的Project，显示New Project（新建项目）对话框（如图1-1所示）。

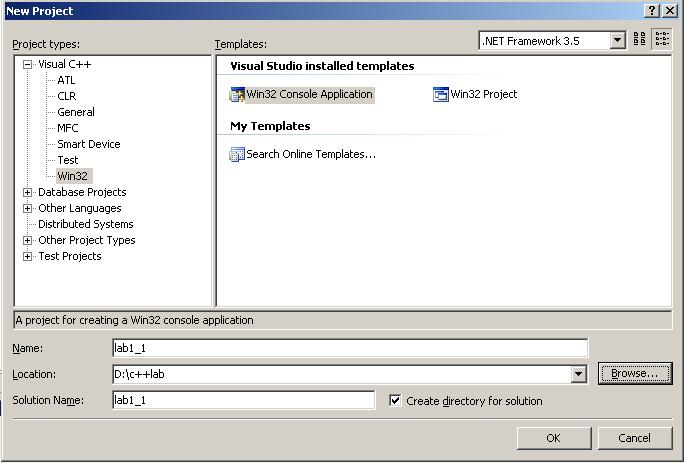


图1-1 New对话框

* 展开Visual C++树状节点单击Win32，选择Win32 Console Application（Win32控制台应用程序）。在Location（位置）文本框中指定一个路径d:\c++lab，在Name（名称）文本框中为项目输入一个名字“lab1\_1”，单击OK（确定）按钮。
* 在弹出的Win32 Application Wizard – lab1\_1 对话框中，单击左侧Application Settings，并选择Console application单选项和Empty Project复选框，然后单击Finish（完成）按钮（如图1-2所示）。
* 最后在New Project Information对话框中单击OK按钮，完成项目的建立。

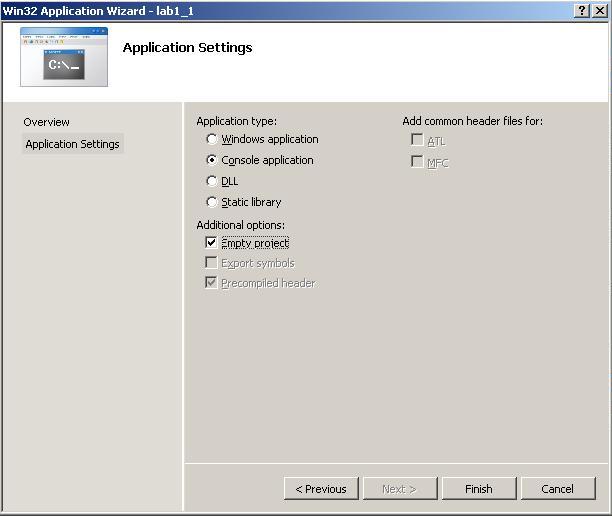


图1-2 创建控制台应用程序第一步

* + 1. 建立C++源程序文件
* 选择菜单命令Project | Add New Item，弹出Add New Item对话框。
* 如图1-3所示，单击左侧树状结构中的Code节点，然后在右侧选择C++ File (.cpp)，并填入文件名称“lab1\_1”，单击OK按钮，完成新建C++源程序文件。

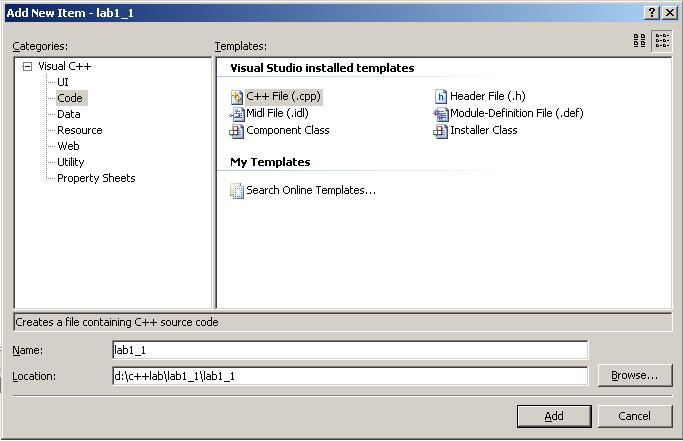


图1-3 添加新的源程序文件

* + 1. 编辑C++源程序文件内容
* 在文件编辑窗口中输入代码（如图1-4所示）。

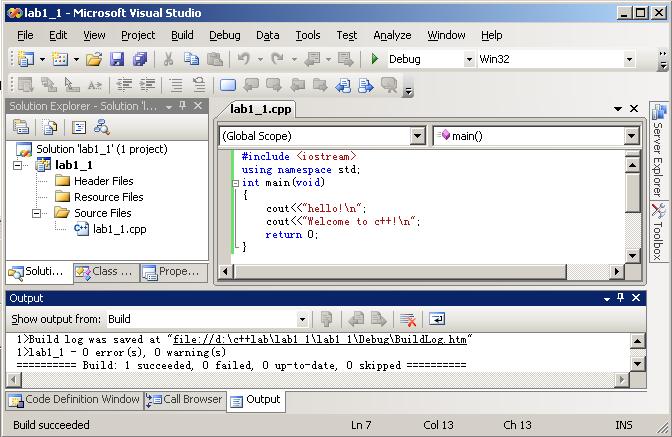


图1-4在文件编辑窗口中输入代码

* 选择菜单命令File | Save lab1\_1.cpp保存这个文件。
  + 1. 建立并运行可执行程序
* 选择菜单命令Build-> Build Solution建立可执行程序。  
  如果你正确输入了源程序，此时便成功地生成了可执行程序lab1\_1.exe。如果程序有语法错误，则屏幕下方的状态窗口中会显示错误信息。根据这些错误信息对源程序进行修改后，重新选择菜单命令Build | Build Solution，建立可执行程序。
* 选择菜单命令Debug | Start Without Debugging运行程序，观察屏幕的显示内容。
  + 1. 关闭工作空间
* 选择菜单命令File | Close关闭工作空间。

第二部分：Eclipse开发环境应用入门（2学时）

* 1. 实验目的

1. 熟悉Unix/Linux环境下开发C/C++的方法
2. 了解Eclipse的特点。
3. 熟悉Eclipse开发环境。
4. 学习用Eclipse编写标准的C++控制台程序。
   1. 实验任务
5. 安装Eclipse SDK和CDT插件。
6. 使用Eclipse建立一个非图形化的标准C++程序，编译、运行教材例2.1，即以下程序：

#include <iostream.h>

int main(int argc, char \* argv[])

{

cout<<"Hello!\n";

cout<<"Welcome to c++!\n";

return 0;

}

* 1. 实验步骤

1. 安装Eclipse IDE for C/C++ Developers。

Eclipse通常被人们用作开发Java程序的IDE。实际上，更准确地说Eclipse是一个平台，它的核心（Core）是很小的，其它绝大部分功能都是通过插件的形式组织上去的，包括Eclipse自带的JDT（Java Development Tools）开发环境，此外Eclipse是一个跨平台的开发环境，可以运行于Windows环境，也可以运行于Unix/Linux环境。

正是由于Eclipse的这种插件组织以及跨平台的特性，使得Eclipse的应用非常广泛，可以被用来作为多种计算机语言的开发环境，如C/C++，Fortran等。

本次实验以Linux环境下Eclipse的使用为例，开发环境的版本为Eclipse IDE for C/C++ Developers 1.2.2，其中包含了Eclipse Platform 3.5.2；Eclipse C/C++ Development Tools（CDT）6.0.2，由于Eclipse各个版本之间的变动比较大，因此请尽量使用与本书相同的版本。（如图1-5，1-6所示）

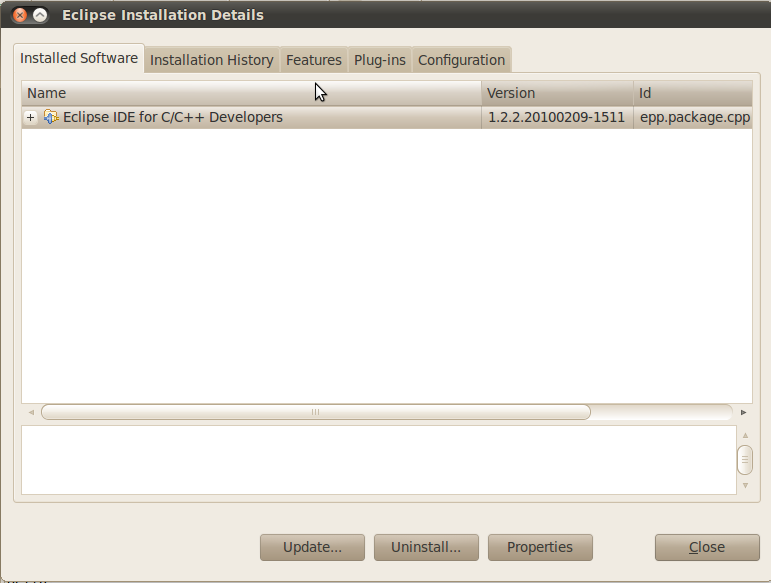


图1-5 Eclipse IDE for C/C++ Developers 的版本号

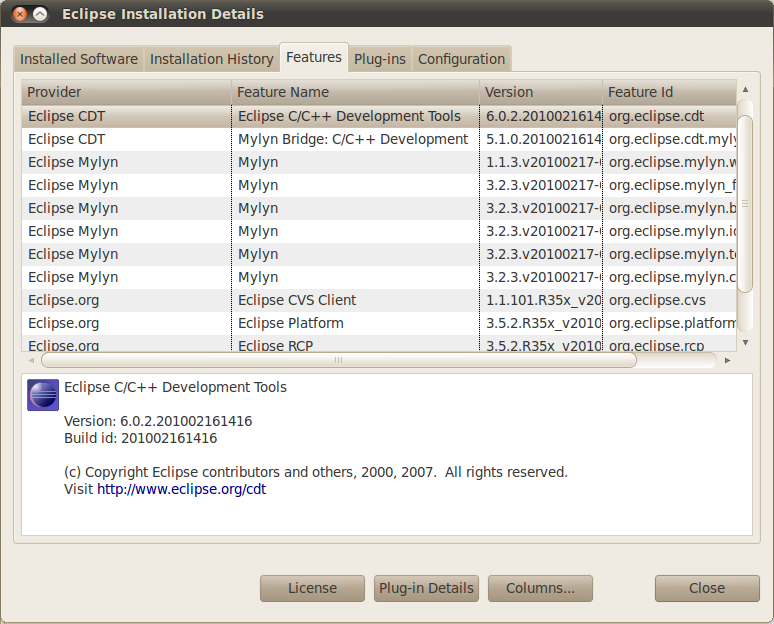


图1-6 CDT以及Eclipse Platform的版本号

下面介绍Eclipse IDE for C/C++ Developers 1.2.2的安装方法，如果您已经安装过Eclipse SDK但尚未安装CDT，可参考Eclipse官方网站的相关说明安装CDT插件，以下假设您尚未安装任何Eclipse相关的软件包。

Eclipse IDE for C/C++ Developers可以直接从Sun的官方网站上下载（http://www.eclipse.org/downloads）。Eclipse属于绿色软件，它不需要安装，下载成功后直接把文件eclipse-cpp-galileo-SR2-linux-gtk.tar.gz解压缩到一个合适的目录就可以了，如/root/installed。然后运行/root/installed/eclipse/目录下的eclipse二进制文件就可以了。

这时Eclipse会提示你选择Workspace，随便输入一个，如/root/Workspace。接着你会看到如下所示的欢迎界面。(如图1-7所示)

如果你在运行Eclipse的时候遇到一些莫名其妙的错误，那么可能是Eclipse使用了你的Linux系统使用了开源组织GNU开发的Java虚拟机gij，它与Sun开发的JRE有一些不兼容的地方，解决方法安装JRE，并把系统的JVM切换到SUN JVM。

以Ubuntu为例，打开终端运行“sudo apt-get install sun-java6-jre sun-java6-jdk sun-java6-plugin”根据提示便可以安装，如果运行这条命令时出现了类似“现在没有可用的软件包”的问题，那么可能是Ubuntu版本较新，新版本的Ubuntu默认去掉了partner软件源，在system（系统）中的Administration（系统管理）单击Software Source（软件源），选择Other Software（其他软件）选项卡，勾选<http://archive.canonical.com/> lucid partner，关闭窗口时系统会提示更新软件源，更新完毕后，再次在终端运行上述命令，根据提示安装SUN JRE，安装JRE成功后在终端运行“sudo update-alternatives --config java”，根据提示选择SUN提供的JVM，将JVM切换到SUN JVM。

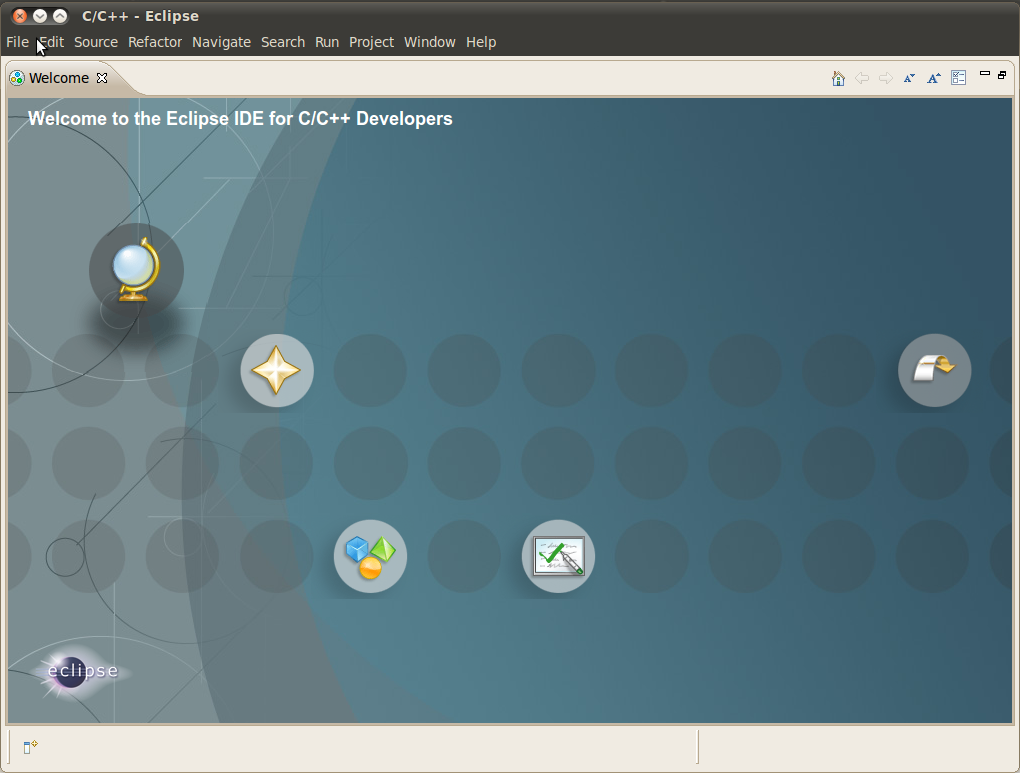


图1-7 Eclipse的欢迎界面

2．创建一个C++项目

单击File菜单中的New子菜单，选择Project，打开如下所示的对话框（如图1-8所示），选择C++ Project：

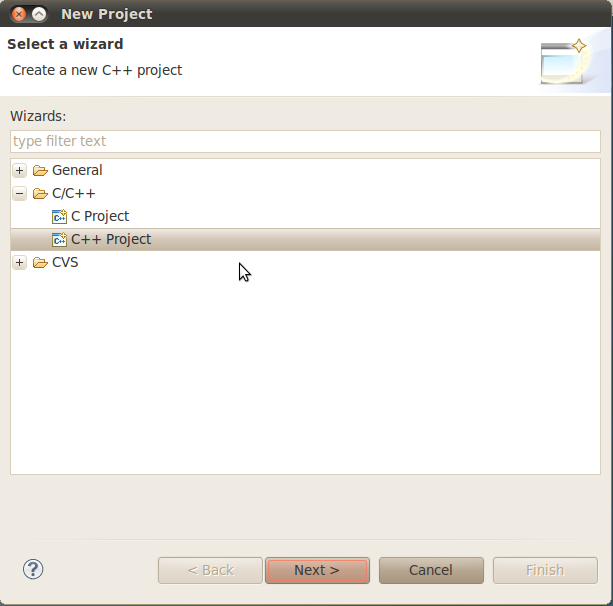


图1-8新建Standard Make C++ Project

接着输入一个项目名称，如“Hello”。(如图1-9所示)。然后直接按“Finish”就可以。这时Eclipse会问你是否打开C/C++视图（Perspective），选择是，并让它“Remember my decision”。

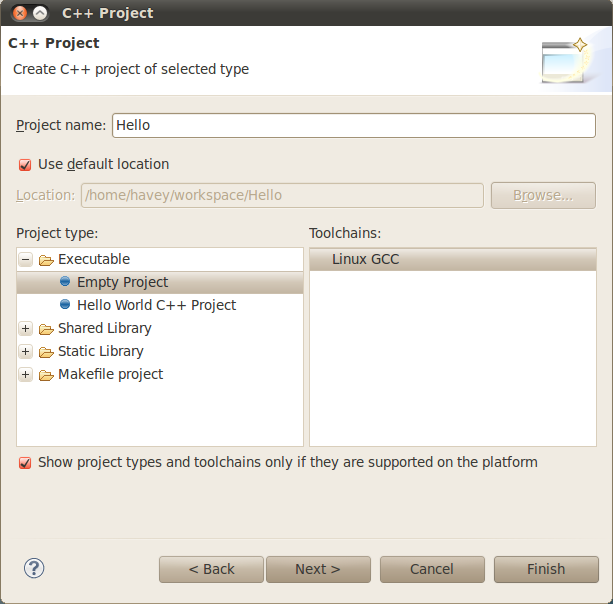


图1-9新建一个Hello项目

3．建立C++源程序文件

选择菜单命令：File->New->Source File，弹出New Source File对话框。

在New Source File对话框的Source File一栏中填入C++源文件名，将文件名为main.cpp，单击Finish按钮，即可完成源程序文件的创建。(如图1-10所示)

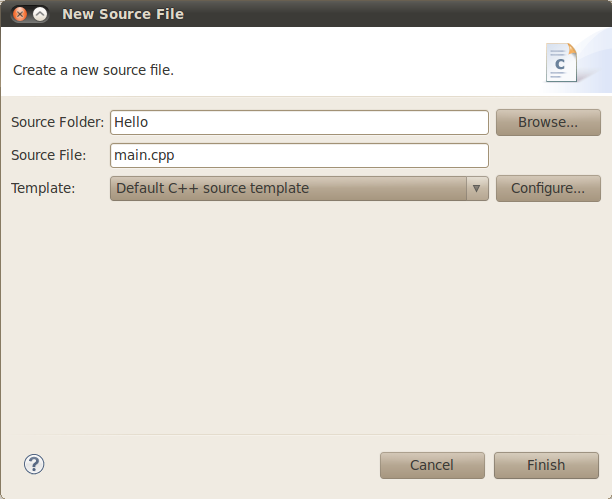


图1-10建立C++源程序文件

4．编辑C++源程序文件内容

在文件编辑窗口中输入代码：(如图1-11所示)

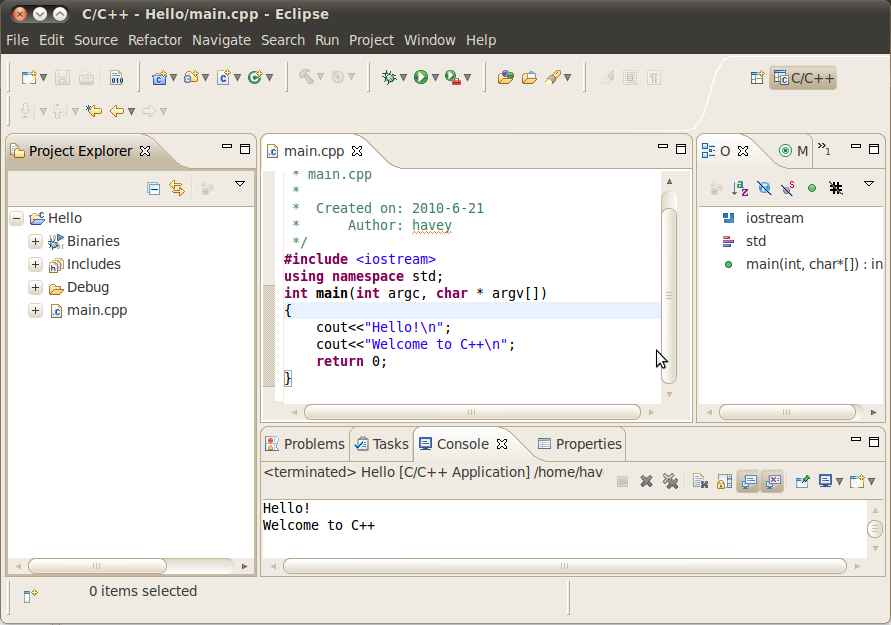


图1-11编辑C++源程序文件

5．建立Makefile文件

由于Eclipse编译C++源文件的时候使用了GNU的开发工具make，而make的实现需要依赖一个叫做Makefile的文件，如果工程中没有Makefile，默认情况下Eclipse会自动为您生成一个Makefile，为了了解make的原理这里我们手工编写一个Makefile。有关Makefile的编写方法及各参数的含义请参见附录B。

新建Makefile的步骤如下：在C/C++视图的项目名处单击鼠标右键，选择New菜单下的File，在弹出的New File对话框的File name输入框中输入Makefile (如图1-12所示)。然后在编辑区域输入Makefile的内容。

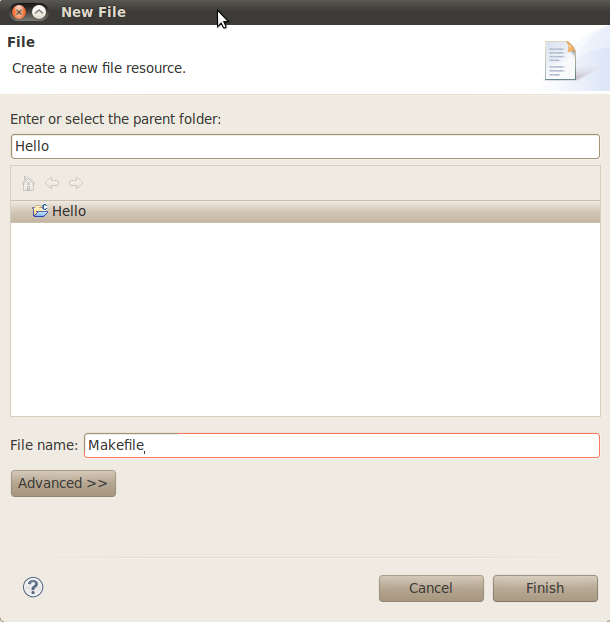


图1-12建立Makefile文件

本例中使用的Makefile文件内容如下所示：

PROG = Hello

DEP = $(PROG).cpp

CC = g++

CFLAGS = -Wno-deprecated

all : $(PROG)

$(PROG) : $(DEP)

$(CC) $(CFLAGS) -o $@ $^

clean:

rm -f \*.o \*~ $(PROG)

保存所有改动过的文件之后，就可以创建二进制可执行程序了。单击Project->Build All就可以了。（为方便程序的编译和修改，最好把Project菜单下的Build Automatically选项取消。）

当程序编译完毕后就可以运行了。程序的运行方法为：单击C/C++ Projects视图下的项目名称，在本例中就是Hello项目，然后再单击菜单Run->Run As->Run Local C/C++ Application（如果该菜单不可执行，则说明没有C/C++ Projects视图下的项目名称，请单击后再选择执行该菜单）。这时就可以在控制台视图（Console）中看到标准输出的输出结果（如图1-13所示）。

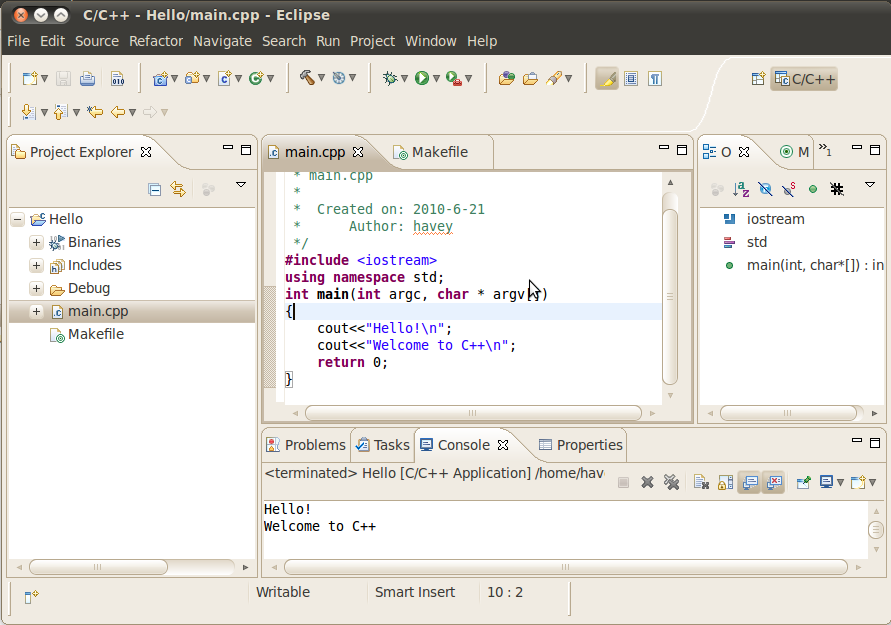


图1-13运行程序后的输出结果在Console中

1. C++简单程序设计（4学时）
   1. 实验目的
      1. 学会编写简单的C++程序。
      2. 复习基本数据类型变量和常量的应用。
      3. 复习运算符与表达式的应用。
      4. 复习结构化程序设计基本控制结构的运用。
      5. 复习自定义数据类型的应用。
      6. 练习使用简单的输入输出。
      7. 观察头文件的作用。
      8. 学会使用Visual Studio 2008开发环境中的debug调试功能：单步执行、设置断点、观察变量值。
      9. 学会使用Eclipse IDE for C/C++ Developers 1.2.2开发环境中的debug调试功能：单步执行、设置断点、观察变量值。
   2. 实验任务
      1. 输入并运行例题2.7，即：用do-while语句编程，求自然数1~10之和。程序正确运行之后，去掉源程序中#include语句，重新编译，观察会有什么问题。
      2. 将do-while语句用for语句代替，完成相同的功能。
      3. 编程计算图形的面积。程序可计算圆形、长方形、正方形的面积，运行时先提示用户选择图形的类型，然后，对圆形要求用户输入半径值，对长方形要求用户输入长和宽的值，对正方形要求用户输入边长的值，计算出面积的值后将其显示出来。
      4. 使用debug调试功能观察任务3程序运行中变量值的变化情况。
      5. 声明一个表示时间的结构体，可以精确表示年、月、日、小时、分、秒；提示用户输入年、月、日、小时、分、秒的值，然后完整地显示出来。
   3. 实验步骤
      1. 建立一个控制台应用程序项目lab2\_1，向其中添加一个C++源文件lab2\_1.cpp（方法见实验一），输入例2.7的代码，检查一下确认没有输入错误，选择菜单命令Build | Build Solution（Eclipse下使用Project | Build All）编译源程序，再选择Debug | Start Without Debugging（Eclipse下使用Run | Run As | Local C/C++ Applocation） 运行程序，观察输出是否与书上的答案一致。
      2. 程序正确运行之后，在源程序第一行“#include<iostream >”前面加注释标记“//”，使之成为注释行，重新编译，此时，编译器会输出类似于下面内容的提示：

表格 1：Visual Studio 2008 环境下输出结果

|  |
| --- |
| 1>------ Build started: Project: lab2\_1, Configuration: Debug Win32 ------  1>Compiling...  1>lab2\_1.cpp  lab2\_1.cpp(3) : error C2871: 'std' : a namespace with this name does not exist  lab2\_1.cpp(11) : error C2065: 'cout' : undeclared identifier  lab2\_1.cpp(11) : error C2065: 'endl' : undeclared identifier  1>lab2\_1 - 3 error(s), 0 warning(s)  ========== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped ========== |

表格 2：Eclipse IDE for C/C++ Developers 1.2.2环境下输出结果

|  |
| --- |
| \*\*\*\* Build of configuration Debug for project Hello \*\*\*\*  make all  Building file: ../ lab2\_1.cpp  Invoking: GCC C++ Compiler  g++ -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"main.d" -MT"main.d" -o"main.o" "../ lab2\_1.cpp"  ../ lab2\_1.cpp: In function ‘int main()’:  ../ lab2\_1.cpp:11: error: ‘cout’ was not declared in this scope  ../ lab2\_1.cpp:11: error: ‘endl’ was not declared in this scope  make: \*\*\* [main.o] Error 1 |

*这是因为C++语言本身没有输入/输出语句，只是C++编译系统带有一个面向对象的I/O软件包，即I/O流类库。cout和cin都是这个类库预定义的流对象，#include <iostream.h>指示编译器在对程序进行预处理时，将头文件iostream.h中的代码嵌入到该程序中该指令所在的地方。文件iostream.h中声明了程序所需要的输入和输出操作的有关信息，在C++程序中如果使用了系统中提供的一些功能，就必须嵌入相关的头文件，否则，系统无法找到实现这些功能的代码。*

现在，删除注释标记，将程序恢复正确。

* + 1. 另建立一个项目lab2\_2，包含一个c++源程序lab2\_2.cpp，将do-while语句用for语句代替，完成与实验任务1相同的功能。
    2. 建立项目lab2\_3，计算图形的面积。圆形的面积计算公式为S=PI\*r\*r，长方形的面积计算公式为S=a\*b，正方形的面积计算公式为S=a\*a；程序中声明一个整型变量iType表示图形的类型，用cout语句输出提示信息让用户选择图形的类型，用cin读入iType的值，然后，使用switch语句判断图形的类型，分别提示用户输入需要的参数值，计算出面积的值后用cout语句显示出来；最后，编译运行程序。
    3. 学习简单的debug调试功能，参考程序如下：

//lab2\_3.cpp

#include <iostream>

using namespace std ;

const float PI = 3.1416;

int main()

{

int iType;

float radius, a, b, area;

cout << "图形的类型为?(1-圆形 2-长方形 3-正方形):";

cin >> iType;

switch(iType)

{

case 1:

cout << "圆的半径为：";

cin >> radius;

area = PI \* radius \* radius;

cout<<"面积为："<<area<<endl;

break;

case 2:

cout << "矩形的长为：";

cin >> a;

cout << "矩形的宽为：";

cin >> b;

area = a \* b;

cout<<"面积为："<<area<<endl;

break;

case 3:

cout << "正方形的边长为：";

cin >> a;

area = a \* a;

cout<<"面积为："<<area<<endl;

break;

default:

cout << "不是合法的输入值!"<<endl;

}

}

*一个程序，特别是大型程序，编写完成后往往会存在这样或那样的错误。有些错误在编译连接阶段可以由编译系统发现并指出（如步骤2所示），称为语法错误。当修改完语法错误生成了可执行程序后，并不意味着程序已经正确。我们常常会发现程序运行的结果与我们预期的结果相去甚远，有时甚至在运行过程中程序中止或发生死机，这种错误称为运行错误，是因为算法设计不当或编程实现时的疏忽造成的。所谓调试就是指在发现了程序存在运行错误以后，寻找错误的原因和位置并排除错误。这一工作是非常困难的，对于初学者而言尤其如此。*

*虽然编译系统不能像对待语法错误那样，明确指出运行错误的原因和位置，但大多数开发环境都为我们提供了辅助调试工具，可以实现单步运行、设置断点、观察变量和表达式的值等功能，使我们可以跟踪程序的执行流程、观察不同时刻变量值的变化状况。*

1. **Visual Studio 2008开发环境下的调试方法**

* 首先在第12行处设置调试断点。用鼠标右键单击源程序第12行左边的空白处，出现一个菜单，如图2-1所示；

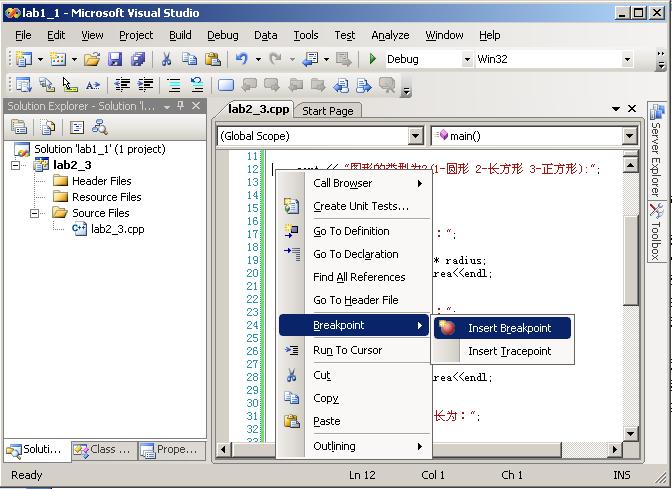


图2-1 添加断点

* 选择Insert Breakpoint一项，可看到左边的边框上出现了一个褐色的圆点，这代表已经在这里设置了一个断点；

*所谓断点就是程序运行时的暂停点，程序运行到断点处便暂停，这样我们就可以观察程序的执行流程，以及执行到断点处时有关变量的值。*

* 然后选择菜单命令Debug | Start Debuging，或按下快捷键F5，系统进入Debug（调试）状态，程序开始运行，一个DOS窗口出现，此时，Visual Studio的外观如图2-2所示，程序暂停在断点处。

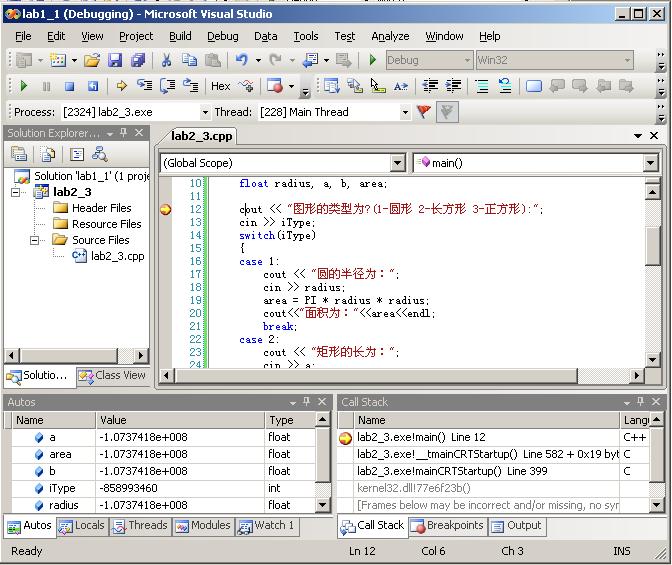


图2-2 调试状态下的Visual Studio

* 单步执行：从Debug菜单或Debug工具栏中单击Step Over两次（或者连续按两次F10）。在程序运行的DOS窗口中输入选择的图形类型，例如，输入3，代表正方形，这时，回到Visual Studio中，把鼠标放在变量名iType上片刻，可看到出现了一个提示：iType = 3；此时，在Variables窗口中也可看到iType以及其他变量的值。

*单步执行时每次执行一行语句，便于跟踪程序的执行流程。因此为了调试方便，需要单步执行的语句不要与其它语句写在一行中。*

* 在Watch窗口中，在Name栏中输入iType，按回车键，可看到Value栏中出现3，这是变量iType现在的值（如果没看到Watch窗口，可通过Debug菜单的Windows | Watch | Watch 1（共有四个Watch 可选）。图2-3是此时Variables窗口和Watch窗口的状态。
* 继续执行程序，参照上述的方法，再试试Debug菜单栏中别的菜单项，熟悉Debug的各种方法。

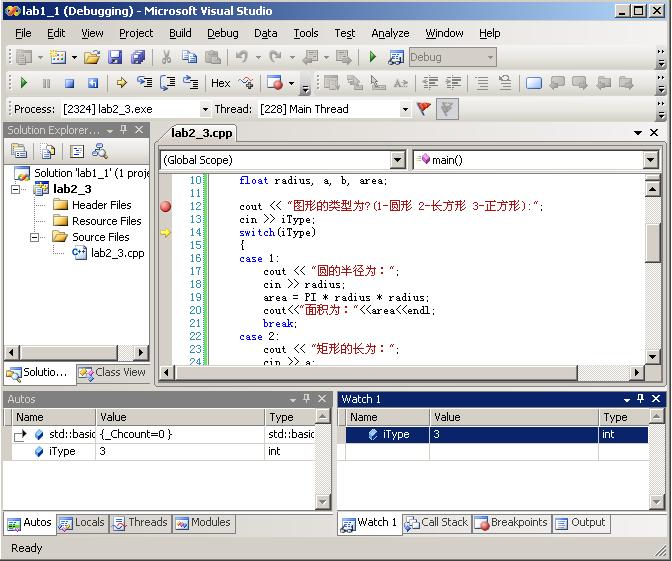


图2-3 Variables窗口和Watch窗口的状态

1. Eclipse IDE for C/C++ Developers 1.2.2**开发环境下的调试方法**

* 首先在第12行处设置调试断点。将光标放到12行，并点击菜单栏中Run | Toggle Breakpoint（或者直接按下Shift+Ctrl+B），可看到左边的边框上出现了一个蓝色的圆点，这代表已经在这里设置了一个断点，如图2-4所示；

*所谓断点就是程序运行时的暂停点，程序运行到断点处便暂停，这样我们就可以观察程序的执行流程，以及执行到断点处时有关变量的值。*

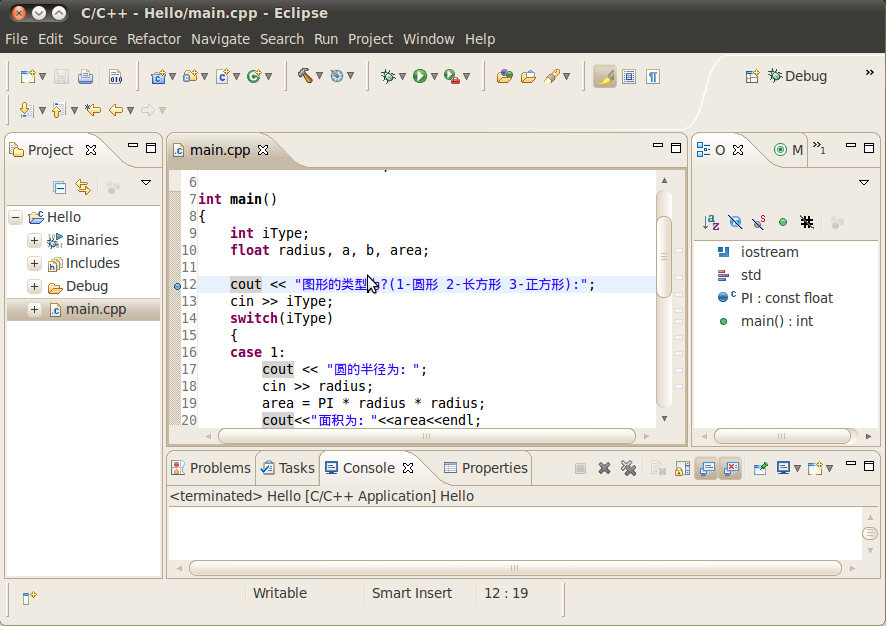


图2-4 添加断点

* 然后选择菜单命令Run | Debug As | Local C/C++ Application，或按下快捷键F11，系统进入Debug（调试）状态，程序开始运行，外观如图2-5所示，程序暂停在断点处。

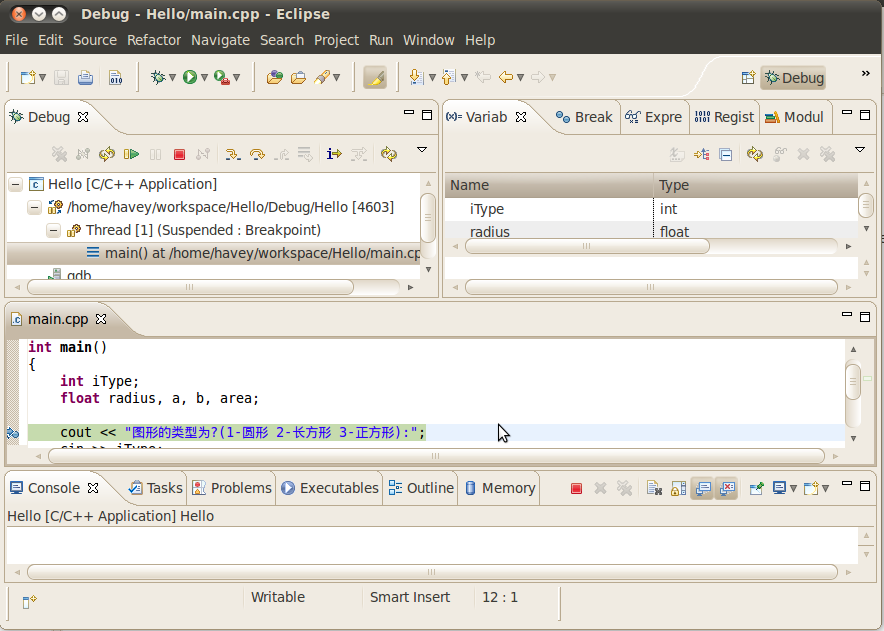


图2-5 调试状态下的Eclipse

* 单步执行：从Run菜单中单击Step Over两次（或者连续按两次F6）。在下方的Console窗口中输入选择的图形类型，例如，输入3，代表正方形，这时，回到代码中，在Variables窗口中也可看到iType以及其他变量的值。

*单步执行时每次执行一行语句，便于跟踪程序的执行流程。因此为了调试方便，需要单步执行的语句不要与其它语句写在一行中。*

* 在Expressions窗口中，在Expression栏中输入iType，按回车键，可看到Value栏中出现3，这是变量iType现在的值（如果没看到Variables窗口或Expressions窗口，可在菜单的Windows | Show View中找到他们并单击打开。图2-6是此时Expressions窗口和Expressions窗口的状态。
* 继续执行程序，参照上述的方法，再试试Debug菜单栏中别的菜单项，熟悉Debug的各种方法。

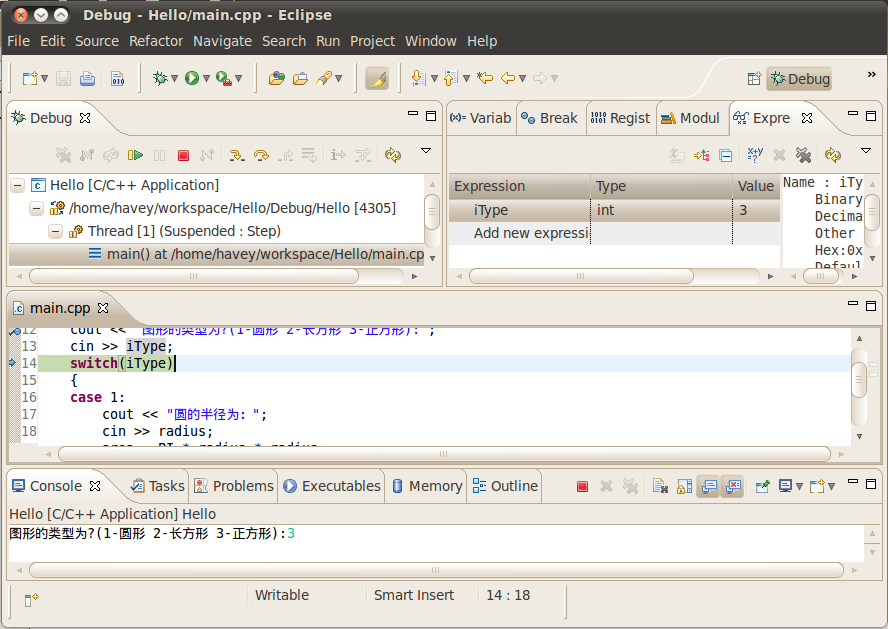


图2-6 Variables窗口和Watch窗口的状态

* + 1. 建立一个项目lab2\_4，包含一个c++源程序lab2\_4.cpp。声明一个表示时间的结构体，有表示年、月、日、小时、分、秒的成员，可以使用short类型；在主程序中实现输入输出。

实验三 函数的应用（2学时）

实验目的

掌握函数的定义和调用方法。

练习重载函数的使用。

练习使用系统函数。

学习使用VS 2008 以及Eclipse的debug调试功能，使用step into追踪到函数内部。

实验任务

编写一个函数把华氏温度转换为摄氏温度，转换公式为：C = (F - 32) \* 5/9。

编写重载函数Max1可分别求取两个整数，三个整数，两个双精度数，三个双精度数的最大值。

使用系统函数pow(x,y)计算xy的值，注意包含头文件math.h。

用递归的方法编写函数求Fibonacci 级数，观察递归调用的过程。

实验四 类与对象（4学时）

* 1. 实验目的
     1. 掌握类的声明和使用。
     2. 掌握类的声明和对象的声明。
     3. 复习具有不同访问属性的成员的访问方式。
     4. 观察构造函数和析构函数的执行过程。
     5. 学习类的组合使用方法。
     6. 使用VS2008以及Eclipse的debug调试功能观察程序流程，跟踪观察类的构造函数、析构函数、成员函数的执行顺序。
  2. 实验任务
     1. 声明一个CPU类，包含等级（rank）、频率（frequency）、电压（voltage）等属性，有两个公有成员函数run、stop。其中，rank为枚举类型CPU\_Rank，声明为enum CPU\_Rank {P1=1,P2,P3,P4,P5,P6,P7}，frequency为单位是MHz的整型数，voltage为浮点型的电压值。观察构造函数和析构函数的调用顺序。
     2. 声明一个简单的Computer类，有数据成员芯片（cpu）、内存（ram）、光驱（cdrom）等等，有两个公有成员函数run、stop。cpu为CPU类的一个对象，ram为RAM类的一个对象，cdrom为CDROM类的一个对象，声明并实现这个类。
     3. （选做）设计一个用于人事管理的People（人员）类。考虑到通用性，这里只抽象出所有类型人员都具有的属性：number（编号）、sex（性别）、birthday（出生日期）、id（身份证号）等等。其中“出生日期”声明为一个“日期”类内嵌子对象。用成员函数实现对人员信息的录入和显示。要求包括：构造函数和析构函数、拷贝构造函数、内联成员函数、组合。

实验五 C++程序的结构（2学时）

* 1. 实验目的
     1. 观察程序运行中变量的作用域、生存期和可见性；
     2. 学习类的静态成员的使用；
     3. 学习多文件结构在C++程序中的使用；
  2. 实验任务
     1. 运行下面的程序，观察变量x、y的值。

//lab5\_1.cpp

#include <iostream>

using namespace std;

void fn1();

int x = 1, y = 2;

int main()

{

cout << "Begin..." << endl;

cout << "x = " << x << endl;

cout << "y = " << y << endl;

cout << "Evaluate x and y in main()..." << endl;

int x = 10, y = 20;

cout << "x = " << x << endl;

cout << "y = " << y << endl;

cout << "Step into fn1()..." << endl;

fn1();

cout << "Back in main" << endl;

cout << "x = " << x << endl;

cout << "y = " << y << endl;

return 0;

}

void fn1()

{

int y = 200;

cout << "x = " << x << endl;

cout << "y = " << y << endl;

}

* + 1. 实现客户机（CLIENT）类。声明字符型静态数据成员ServerName，保存其服务器名称；整型静态数据成员ClientNum，记录已定义的客户数量；定义静态函数ChangeServerName()改变服务器名称。在头文件client.h中声明类，在文件client.cpp中实现，在文件test.cpp中测试这个类，观察相应的成员变量取值的变化情况。

实验六 数组、指针与字符串（4学时）

* 1. 实验目的
     1. 学习使用数组数据对象；
     2. 学习字符串数据的组织和处理；
     3. 学习标准C++库的使用；
     4. 掌握指针的使用方法；
     5. 练习通过debug观察指针的内容及其所指的对象的内容；
     6. 练习通过动态内存分配实现动态数组，并体会指针在其中的作用；
     7. 分别使用字符数组和标准C++库练习处理字符串的方法。
  2. 实验任务
     1. 编写并测试3×3矩阵转置函数，使用数组保存3×3矩阵。
     2. 使用动态内存分配生成动态数组来重新完成上题，使用指针实现函数的功能。
     3. 编程实现两字符串的连接。要求使用字符数组保存字符串，不要使用系统函数。
     4. 使用string类声明字符串对象，重新实现上一小题。
     5. 声明一个Employee类，其中包括姓名、街道地址、城市和邮编等属性，以及change\_name()和display()等函数。display()显示姓名、街道地址、城市和邮编等属性，change\_name()改变对象的姓名属性，实现并测试这个类。
     6. 声明包含5个元素的对象数组，每个元素都是Employee类型的对象。
     7. （选做）修改实验四中的选做实验中的people（人员）类。具有的属性如下：姓名char name[11]、编号char number[7]、性别char sex[3]、生日birthday、身份证号char id[16]。其中“出生日期”声明为一个“日期”类内嵌子对象。用成员函数实现对人员信息的录入和显示。要求包括：构造函数和析构函数、拷贝构造函数、内联成员函数、聚集。在测试程序中声明people类的对象数组，录入数据并显示。

实验七 继承与派生（4学时）

* 1. 实验目的
     1. 学习声明和使用类的继承关系，声明派生类；
     2. 熟悉不同继承方式下对基类成员的访问控制；
     3. 学习利用虚基类解决二义性问题。
  2. 实验任务
     1. 声明一个基类Animal，有私有整型成员变量age，构造其派生类dog，在其成员函数SetAge(int n)中直接给age赋值，看看会有什么问题，把age改为公有成员变量，还会有问题吗？编程试试看。
     2. 声明一个基类BaseClass，有整型成员变量Number，构造其派生类DerivedClass，观察构造函数和析构函数的执行情况。
     3. 声明一个车（vehicle）基类，具有MaxSpeed、Weight等成员变量，Run、Stop等成员函数，由此派生出自行车（bicycle）类、汽车（motorcar）类。自行车（bicycle）类有高度（Height）等属性，汽车（motorcar）类有座位数（SeatNum）等属性。从bicycle和motorcar派生出摩托车（motorcycle）类，在继承过程中，注意把vehicle设置为虚基类。如果不把vehicle设置为虚基类，会有什么问题？编程试试看。
     4. （选做）从实验6中的people（人员）类派生出student（学生）类，添加属性：班号char classNO[7]；从people类派生出teacher（教师）类，添加属性：职务char principalship[11]、部门char department[21]。从student类中派生出graduate（研究生）类，添加属性：专业char subject[21]、导师teacher adviser；从graduate类和teacher类派生出TA（助教生）类，注意虚基类的使用。重载相应的成员函数，测试这些类。类之间的关系如图7-1所示。

实验八 多态性（2学时）

* 1. 实验目的
     1. 掌握运算符重载的方法；
     2. 学习使用虚函数实现动态多态性。
  2. 实验任务
     1. 声明Point类，有坐标\_x，\_y两个成员变量；对Point类重载“++”（自增）、“--”（自减）运算符，实现对坐标值的改变。
     2. 声明一个车（vehicle）基类，有Run、Stop等成员函数，由此派生出自行车（bicycle）类、汽车（motorcar）类，从bicycle和motorcar派生出摩托车（motorcycle）类，它们都有Run、Stop等成员函数。观察虚函数的作用。

（选做）对实验六中的people类重载“==”运算符和“=”运算符，“==”运算符判断两个people类对象的id属性的大小；“=”运算符实现people类对象的