

算法语言与程序设计（C++）

实验指导

信息科学与技术学院
（网络安全学与、牛津布鲁克斯学院）
网络空间安全系

2020 年 2 月

目录

实验 1 认识 Visual C++6.0 开发环境	3
实验 2 数据类型、运算符和表达式	11
实验 3 语句和预处理(分支语句)	14
实验 4 语句和预处理(循环语句)	16
实验 5 函数 (1)	18
实验 6 指针和引用	20
实验 7 函数 (2)	21
实验 8 类和对象 (1)	22
实验 9 类和对象 (2)	24
实验 10 类和对象 (3)	25
实验 11 继承和派生	26
实验 12 多态	28
实验 13 输入/输出流	31
实验 14 模板	33
实验 15 异常处理和命名空间	34
部分参考答案	1
实验 1 认识 Visual C++6.0 开发环境	1
实验 2 数据类型、运算符和表达式	3
实验 3 语句和预处理(分支语句)	6
实验 4 语句和预处理(循环语句)	10
实验 5 函数 (1)	15
实验 6 指针和引用	21
实验 7 函数 (2)	27
实验 8 类和对象 (1)	33
实验 9 类和对象 (2)	41
实验 10 类和对象 (3)	46

实验 1 认识 Visual C++6.0 开发环境

实验目的

- 了解和使用 Visual C++ 6.0（或 CodeBlocks）集成开发环境。
- 熟悉 Visual C++ 6.0（或 CodeBlocks）集成开发环境的基本编辑命令及功能键，学会常规窗口操作，熟悉常用的菜单命令。
- 学习完整的 C++程序开发过程（编辑、编译、连接、调试、运行及查看结果）。
- 理解简单的 C++程序结构。

实验要求

- 在学完教材第 1 章内容后进行本次实验。
- 熟悉 Windows 操作系统的环境和基本操作。
- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

- (1) 熟悉 Visual C++ 6.0（或 CodeBlocks）的开发环境。
- (2) 用应用程序向导创建一个控制台应用项目。
- (3) 输入并执行一个新的 C++程序（代码为教材第 5 页例 1.1）。

具体实验步骤：

(1) 启动 Visual C++ 6.0 开发环境

从“开始”菜单中选择“程序” | Microsoft Visual Studio 6.0 | Microsoft Visual C++6.0，显示 Visual C++ 6.0 开发环境主窗口，如图 1-1 所示。

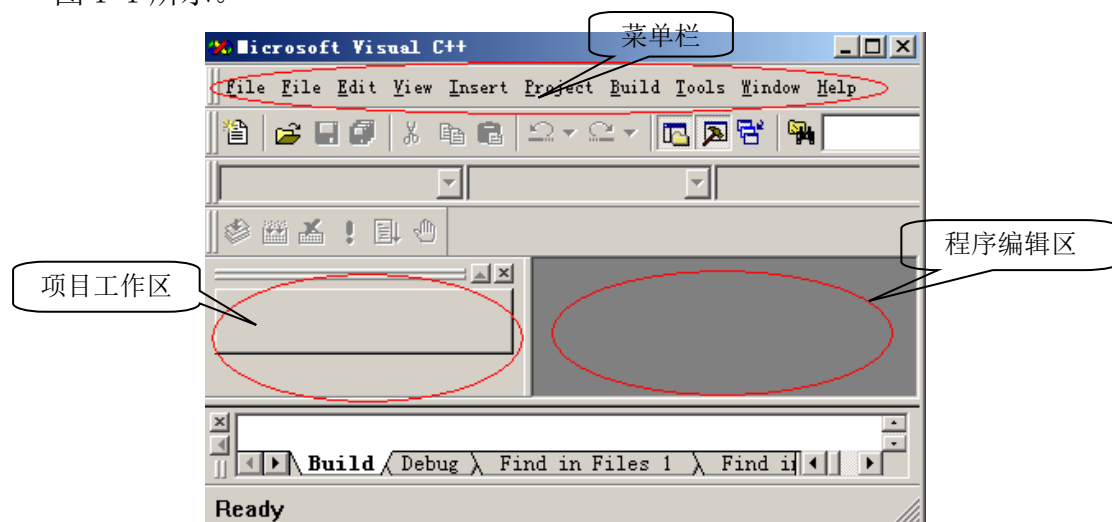


图 1-1 Visual C++6.0 的主窗口

在 Visual C++ 主窗口的顶部是菜单栏，左侧为项目工作区窗口，右侧是程序编辑窗口。工作区窗口用来显示所设定的工作区的信息，程序编辑窗口用来输入和编辑源程序。

(2) 建立和运行只包含一个 C++ 源程序的方法

① 建立只包含一个 C++ 源程序的方法

一个最简单的 C++ 程序只包含一个源程序的文件，建立这样的一个 C++ 源程序的方法是在 Microsoft Visual C++ 主窗口的菜单栏中选择“File → New”命令，弹出“New”对话框，单击“Files”选项卡，在其列表框中选择“C++ Source File”选项，表示要建立一个新的 C++ 源程序文件，然后在对话框右半部分的 Location 文本框中输入准备编辑的源程序文件的存储路径（假设为 F:\C++），表示准备编辑的源程序文件将存放在 F 盘 C++ 子目录下，在其上方的 File 文本框中输入准备编辑的源程序文件的名字（假设为 example1.cpp），如图 1-2 所示。

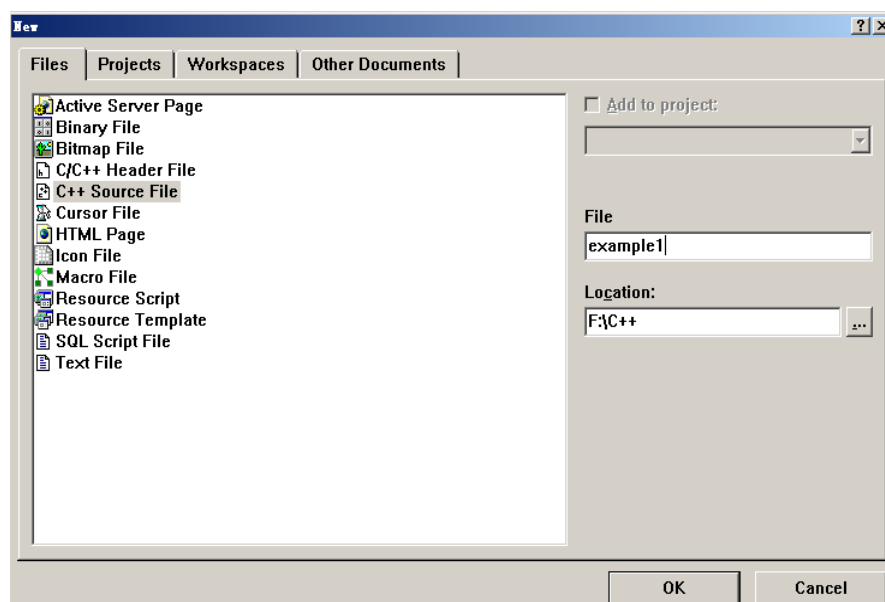


图 1-2 Files 中 New 选项卡

单击“OK”按钮回到主窗口，在程序编辑窗口编辑源程序文件，如图 1-3 所示。

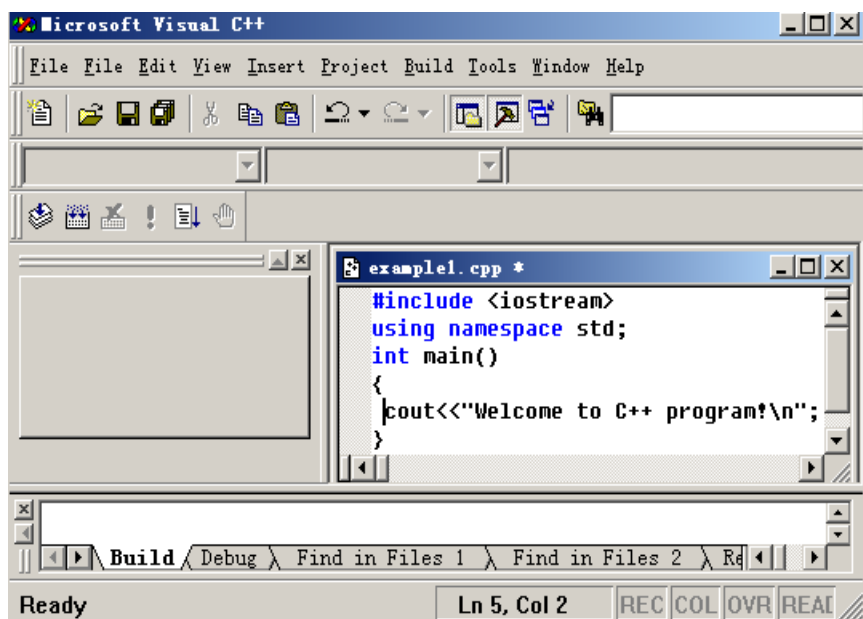


图 1-3 编辑源文件

选择“File → Save”命令或快捷键Ctrl+S保存源程序文件。也可以选择“File → Save As”命令指定新的文件名和路径，这时输入和编辑的源程序就以example1.cpp为文件名存放在F:\C++子目录下。

②运行只包含一个C++源程序的方法

若要运行当前的源文件或一个已有的源文件，首先需要对该源文件进行编译，只要选择“Build → Compile example1.cpp”命令进行编译即可。

在选择编译命令后，屏幕上出现一个对话框，内容是“This build command requires an active project workspace. Would you like to create a default project workspace?”（此编译命令要求有一个有效的项目工作区。你是否同意建立一个默认的项目工作区？），如图1-4所示。

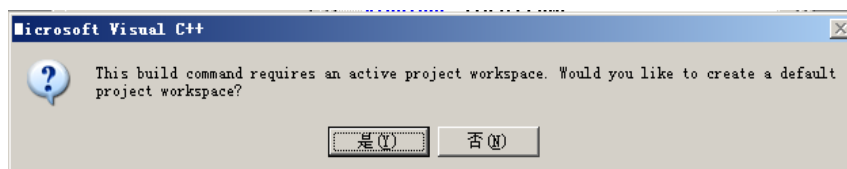


图 1-4 对话框提示信息

单击“是（Y）”按钮，表示同意由系统建立一个默认的项目工作区，然后开始编译。在进行编译时，编译系统检查源程序文件有无语法错误，然后在主窗口下部的调试信息窗口输出编译的信息。如果有错误，就会指出错误的位置和性质，如图1-5所示。

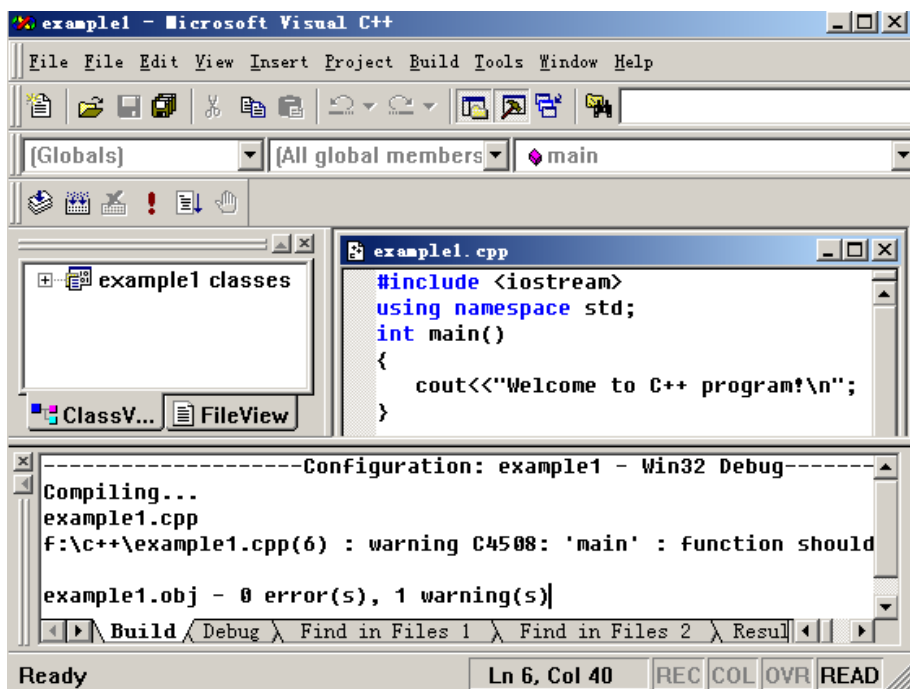


图 1-5 编译提示信息

如果存在致命错误（error）和警告性错误（warning），则需要修改并重新编译；没有错误，则编译成功，生成一个 example1.obj 文件。

编译成功后，就可以对程序文件进行连接了。选择主菜单中的“Build → Build example1.exe”命令，表示要求连接并建立一个可执行文件 example1.exe。运行该文件，在屏幕就可以看到输出结果。

**（下面的多文件的程序编写部分内容选作，可以在学习完函数部分后
再做）**

2. 建立一个**多文件的程序**，求最大值。主函数放在一个文件中，Max 函数放在另一个文件中。

如果一个程序中包含多个文件，则需要建立一个 project file（项目文件），project file 是放在 Workspace（项目工作区）中，并在项目工作区的管理下工作的，因此，如果有多个项目文件存在时，还需要建立一个项目工作区。在编译时，先分别对每个文件进行编译，然后将项目文件中的文件连接成一个整体，再与系统的有关资源进行连接，生成一个可执行文件，最后执行这个文件。

在实际操作时有两种方法：一种是由用户建立项目工作区和项目文件；另一种是用户只建立项目和文件，而不建立项目工作区，由系统自动建立项目工作区。后一种方法比前一种方法操作更简单。

由用户建立项目工作区和项目文件的具体操作为：

(1) 分别编辑好同一个程序的多个源程序文件，并存放在指定的目录下。如一个程序包含两个源程序文件 file1.cpp 和 file2.cpp，如图 1-6、1-7 所示，并已经把它们保存在 F:\C++子目录下。

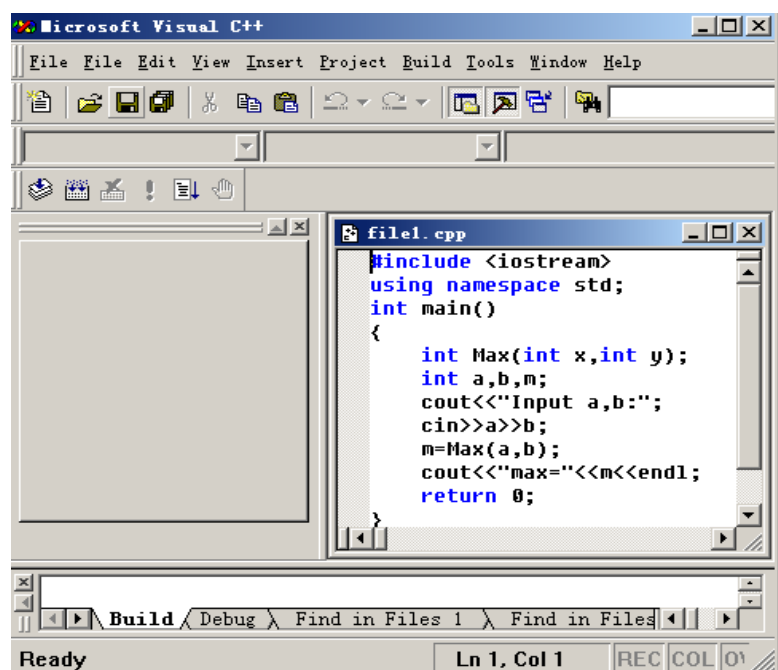


图 1-6 file1.cpp 源文件

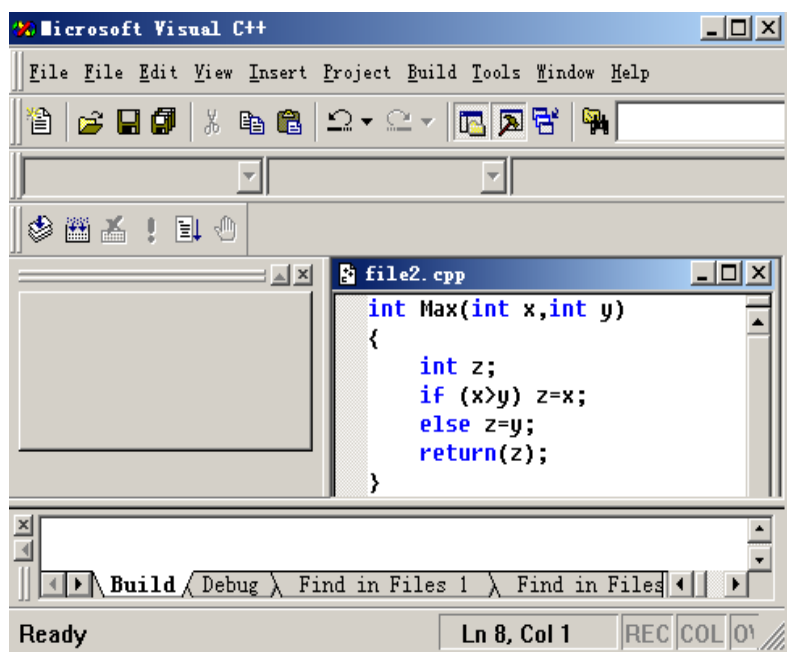


图 1-7 file2.cpp 源文件

(2) 建立一个项目工作区

建立一个项目工作区需要在 Microsoft Visual C++6.0 的主窗口中选择“File → New”命令，然后单击此对话框上方的 Workspace 选项卡，在对话框右部 Workspace name 和 Location 中分别输入工作区名 ws1 和路径名 F:\C++\ws1，如图 1-8 所示。

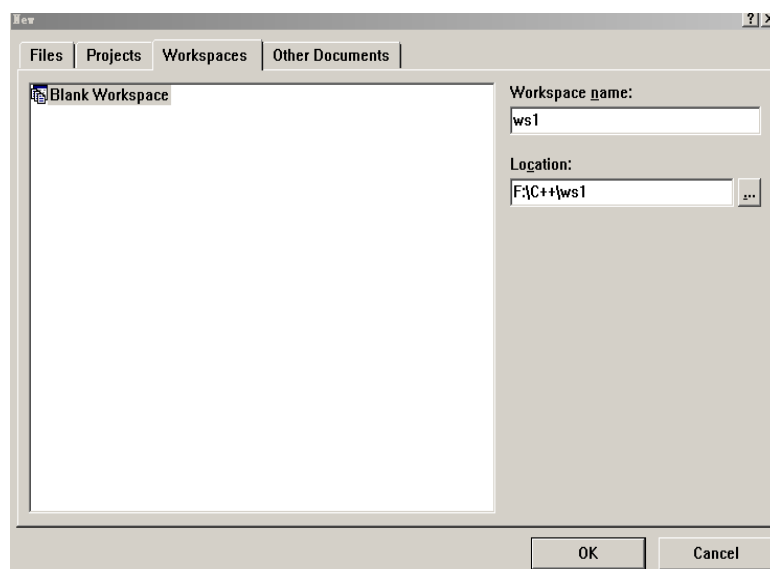


图 1-8 选择 File→New 命令 Workspace 选项卡

单击“OK”按钮，返回主窗口。此时在屏幕左部的工作区窗口中显示了“Workspace ‘ws1’: 0 project(s)”，说明当前的工作区名是 ws1，其中没有放入项目文件，如图 1-9 所示。

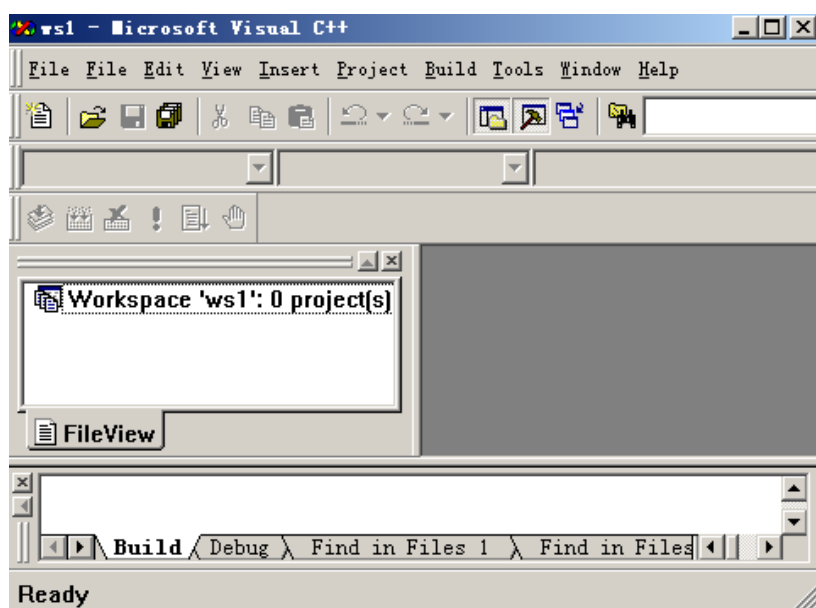


图 1-9 新建工作区 ws1

(3) 建立项目文件

选择“File →New”命令，然后单击此对话框上方的 Project 选项卡，在对话框左部的列表中选择“Win32 Console Application”项，在右部 Project name 和 Location 中分别输入项目文件名 Project 和路径名 F:\C++\ws1\project1，并需要选中是否加入当前工作区，还是创建一个新的工作区单选按钮（假设选择加入当前已有工作区），如图 1-10 所示。

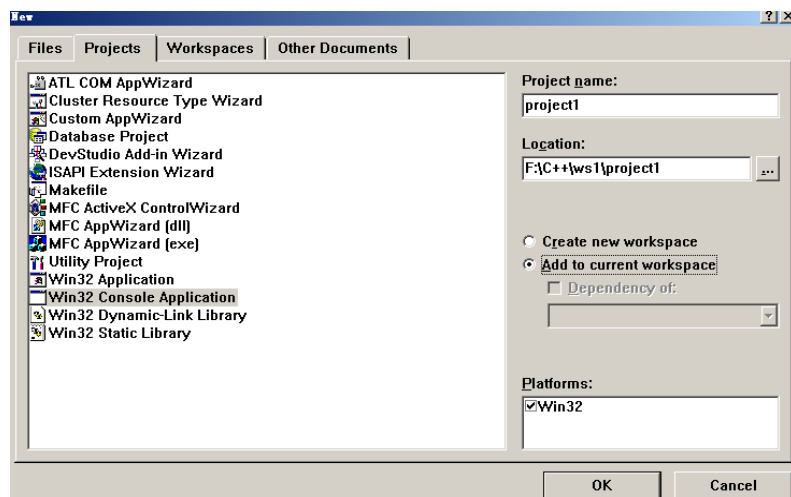


图 1-10 选择 File→New 命令 Project 选项卡

单击“OK”按钮后，在弹出的对话框上选中“An empty project”单选按钮，单击“Finish”按钮，则系统弹出一个工程信息对话框，单击“OK”按钮，出现图 1-11 所示对话框。

可以看到在屏幕左部的工作区窗口中显示了“Workspace ‘ws1’: 1 project(s)”，说明当前的工作区名是 ws1，其中已经放入了项目文件 project1。

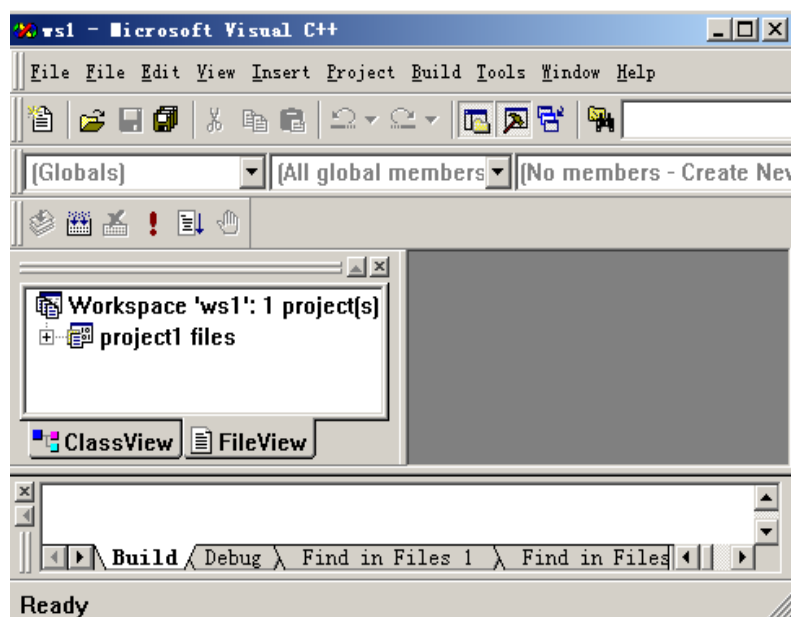


图 1-11 新建的项目文件 project1

项目文件建好后，选择将源程序文件加入到该项目文件中。选择“Project → Add To Project → Files”命令，找到要加入的源程序文件 file1.cpp 和 file2.cpp，把他们加入到项目文件 project1 中，其显示结果如图 1-12 所示。

此时可以对加入多个源程序文件的项目文件进行编译和连接了。方法是：单击主菜单中的“Build → Build Project1.exe”命令，系统将对整个项目文件进行编译和连接，在窗口下部会显示编译和连接的信息。如果程序有错，会显示出错信息，这时需要进行修改；如果没错，则生成可执行文件 Project1.exe。

选择主菜单中的“Build → Execute Project1.exe”命令就可以执行 project1.exe。

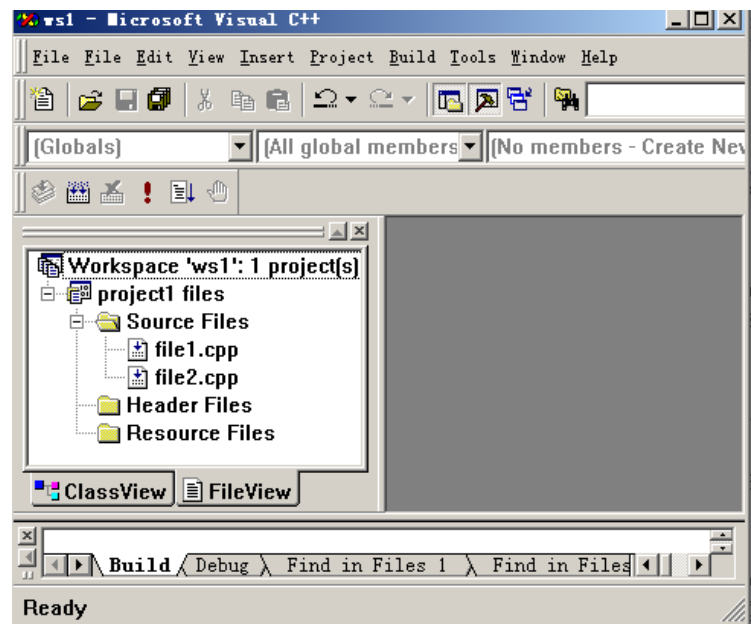


图 1-12 加入 file1.cpp 和 file2.cpp 的 project1
在程序运行时，根据要求输入所需的数据，回车后即得到程序的输出结果。

2.进阶部分

(1) 参考教材第 6 页例 1.2 代码，编写一个 C++程序：输入圆的半径，输出其周长。

(2) 实现一个多行输出的程序，输出内容为：

```
*  
***  
*****  
*****
```

实验 2 数据类型、运算符和表达式

实验目的

- 了解基本数据类型的字节宽度和范围表示。
- 掌握表达式中各种运算符的功能和特点,掌握基本的输入输出方法。
- 理解表达式和语句的概念。
- 学习过程化程序设计的方法。
- 培养编写程序的艺术,明确程序可读性是程序质量的重要标准。
- 初步学习调试方法。

实验要求

- 根据实验内容的相应描述和要求,自行设计并调试代码。
- 认真填写实验报告。

实验内容

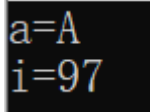
1.基础部分

(1) 测试基本数据类型 char 和 int 之间的相互转换。

设有部分代码如下:

```
a = 65;  
i='a';  
cout << "a=" << a << endl;  
cout << "i=" << i << endl;
```

如果希望得到如下的输出结果:



```
a=A  
i=97
```

请完善代码,(对变量 a 和 b 的定义)。

(2) 测试前缀和后缀自增、自减运算符。

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int myAge, yourAge;  
    myAge=18;
```

```

yourAge=20;
cout << "I am " << myAge << " years old." << endl;
cout << "You are " << yourAge << " years old." << endl;

myAge++;
++yourAge;
cout << "I am " << myAge << " years old." << endl;
cout << "You are " << yourAge << " years old." << endl;

cout << "I am " << myAge++ << " years old." << endl;
cout << "You are " << ++yourAge << " years old." << endl;

cout << "I am " << myAge << " years old." << endl;
cout << "You are " << yourAge << " years old." << endl;

return 0;
}

```

分析代码，说出每一个输出语句所输出的值的大小和原因，尤其要体会理解标红的语句的语法规则。

(3) 测试 `cin` 和 `cout` 的基本输入和输出用法。设有如下语句：

```

float x=100,y=3.14;
char a='t',b='s';
cout<<"x is"<<setw(10)<<x<<"\t";
cout<<setfill('*')<<setw(20)<<y<<endl;
cout<<a<<b<<endl;

```

则最终的输出结果是什么？体会理解 `setw()`、`setfill()` 的作用。

(4) 编写程序，求圆的周长、圆面积、圆球体积、圆柱体积。要求用 `const` 设定 `PI` 常量，定义适当数据类型的变量，并设圆、球半径和圆柱的高的初值分别为 2.5、4，依次计算上述结果并输出，输出时要求有相应的文字提示。

(5) 试设计程序，实现：从键盘输入一个三位数，从左到右用 `a`、`b`、`c` 表示各位数字，记为 `abc`，现要求依次输出从右到左的各位数字，即输出另一个三位数 `cba`。例如，输入 123，输出 321。

2.进阶部分

(1) 用 `sizeof` 运算符编写一个测试程序，测试本机中各基本数据类型或字符串所占的字节数，并将其填写到下表中，然后分析其结果。

基本数据类型	所占字节数	基本数据类型或字符串	所占字节数
<code>char</code>		<code>float</code>	
<code>short</code>		<code>double</code>	

int		long double	
long		“\nCh\t\v\0ina”	

(2) 已知 $a=b=100000$, $c=1000$, 编程求 $a*b/c$, 注意不要让中间结果溢出。

实验3 语句和预处理(分支语句)

实验目的

- 理解程序的分支结构，能实现简单的条件判断程序。
- 进一步学习编程技巧，条件表达式代替条件语句，逻辑短路表达式。
- 掌握 switch 多分支结构。

实验要求

- 基础部分实验必选，进阶部分实验至少选择一个。
- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

- (1) 编写一个程序，只要输入生日的年、月、日，就能计算出生日是星期几。
- (2) 利用肥胖程度计算公式 BMI 法自我诊断肥胖程度。BMI 法如下：
体重指数 = 体重（公斤）除 身高（米）的平方。即 kg/m^2 。
正常体重：体重指数 = 18 ~ 25
超重：体重指数 = 25 ~ 30
轻度肥胖：体重指数 > 30
中度肥胖：体重指数 > 35
重度肥胖：体重指数 > 40
编写程序，输入体重和身高，输出肥胖程度。
- (3) 程序 Ex_Switch 给出一个百分制成绩，要求输出成绩等级 A、B、C 和 D。其中，85 分以上为 A，75~84 分为 B，65~74 分为 C，65 分以下为 D。
- (4) 程序 Ex_Root 求解一元二次方程 $ax^2+bx+c=0$ 的根。当 $a=0$ 且 $b=0$ 时，方程无根。当 $a=0$ 且 $b \neq 0$ 时，方程有一个根。当 $a \neq 0$ 时，根据 $\Delta=b^2-4ac$ 确定方程的根，若 $\Delta>0$ ，有两个不同的实根；当 $\Delta=0$ ，有两个相同的实根；当 $\Delta<0$ ，有两个不同的复根。输入参数 a、b、c，输出相应的结果。

2.进阶部分

- (1) 实现一个简单的菜单程序，运行时显示“Menu: A(dd) D(delete) S(ort) Q(uit), Select one:”提示用户输入，A 表示增加，D 表示删除，S 表示排序，Q 表示退出，输入为 A，D，S 时分别提示“数据已经增加、删除、排序。”输入 Q 时程序结束。
要求：使用 switch 语句进行判断。
- (2) 编程实现输入一个整数，判断其能否被 3，5，7 整除，并输出以下信

息之一：

- ①能同时被 3，5，7 整除；
- ②能被其中两数（要指出哪两个）整除；
- ③能被其中一个数（要指出哪一个）整除；
- ④不能被 3，5，7 任一个整除。

要求：使用 if-else 语句。

实验 4 语句和预处理(循环语句)

实验目的

- 掌握表达式中逻辑运算符、关系运算符的功能和特点。
- 掌握 while 和 do-while 循环的区别。
- 掌握 for 循环的语法结构并用其解决问题。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

(1) 编写程序用于解 Fibonacci 数列问题。Fibonacci 数列的前两个数是 1 和 1，从第三个数开始，每个数等于前两个数之和。编程计算并输出此数列的前 30 个数，且每行输出 5 个数。

(2) 编写程序用泰勒级数求 e 的近似值，直到最后一项小于 10^{-6} 为止。

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

(3) 编写程序打印下列菱形图案：

```
      *
     * * *
    * * * * *
   * * * * * *
  * * * * *
 * * * *
* * *
 *
```

2.进阶部分

(1) 对运行中输入的 x，计算级数：

$$1 + x - x^2/2! + x^3/3! - \dots (-1)^{n+1} * x^n/n!。$$

要求输出精度为 10^{-8} 。

(2) 编程求所有的 3 位素数，且该素数是对称的，所谓“对称”是指一个数倒过来还是该数。例如，375 不是对称数，因为倒过来变成了 573。

(3) 编程打印九九乘法表：

实验5 函数（1）

实验目的

- 掌握函数声明、定义和使用的方法。
- 掌握函数递归调用的方法。
- 掌握重载函数及函数参数默认的使用方法。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

（1）编写程序求三角形的面积。已知三角形的三边 a 、 b 、 c ，则三角形的面积为

$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

其中， $s=(a+b+c)/2$ 。需要说明的是，三角形的三边的边长由 `cin` 输入，需要判断这三边是否构成一个三角形（要判断输入的三条边能否够成三角形，只需满足任意两边之和都大于第三边即可）。若是，则计算其面积并输出，否则输出“错误：不能构成三角形！”。编写一个完整的程序，其中需要两个函数，一个函数判断是否构成三角形，另一个函数计算三角形的面积。

（2）编程求下式值，其中 n^i 用函数来实现，且设参数 n 的默认值为 2：

$$n^1 + n^2 + n^3 + n^4 + \cdots + n^{10} \quad n = 1, 2, 3$$

（3）用递归法将一个整数 n 转换成字符。如输入 1234，应输出字符 ‘1’ ‘2’ ‘3’ ‘4’。 n 的位数不确定，可以是任意位数的整数。

2.进阶部分

（1）设计一个程序，通过重载求两个数中最大数的函数 `max()`，分别实现求两个实数和两个整数及两个字符的最大者。

（2）设计一个函数，要求能将一个正整数 n 按反序输出， n 的位数不定。例如，123 输出 321。

要求：用非递归函数来实现，编写完整的程序并测试。

（3）用递归方法求解母牛问题。若一头母牛，从出生起第四个年头开始每年生一头母牛，按此规律，第 n 年时有多少头母牛？

要求：用递归函数求解；先测试递归条件，后递归调用；运行时 n 由键盘输入。

实验 6 指针和引用

实验目的

- 学习指针的定义、初始化、赋值、使用的方法。
- 声明和操作指针和引用。
- 学习调试方法。
- 学习数组的定义与使用

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

(1) 程序中分别定义不同的变量,定义指针分别指向不同变量,实现变量值的交换。

(2) 程序中分别定义不同的变量,定义引用分别指向不同变量,实现变量值的交换。

2.进阶部分

(1) 输入 10 个数到一维数组中，用冒泡排序法将其按从大到小的次序排序后输出。

要求：分别用三个函数实现数据的输入、排序及输出。

(2) 输入一个 5*5 的整数矩阵，求出两条对角线元素值之和。

(3) 编写一个函数，统计一条英文句子中字母的个数，在主程序中实现输入、输出。

实验 7 函数 (2)

实验目的

- 掌握指针做函数参数的用法。
- 掌握重载函数的定义与使用方法。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础实验

(1) 编程实现下述功能：定义 `add` 函数实现 2 个同数据类型数据的加（可以是实数+实数，也可能是整数+整数），并定义主函数调用 `add` 函数。

(2) 编程实现数据交换。分别用指针和引用作为函数 `mySwap` 的形参，并通过重载 `int`、`double` 型变量值的交换。

2.进阶部分

(1) 字符串交换

编写程序，使用指针作为参数通过函数调用的方式交换两个不同长度的字符串。

要求：主函数中定义两个字符串，然后调用交换函数，打印交换前和交换后字符串。

(2) 设计一个函数，形参是引用类型，要求输入三个整数，按从小到大排序后输出。编程完整的程序并测试。

实验 8 类和对象（1）

实验目的

- 学习类与对象的定义和声明；
- 掌握类与对象的使用方法；
- 学习数据成员与成员函数的访问方式。学习定义类的方法。
- 学习构造函数、析构函数的定义使用方法。
- 学习对象的使用方法。
- 领会面向对象程序设计的基本方法。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

（1）应用 VC++6.0 的编辑环境构造一个类 `Triangle`，该类主要实现三角形的基本操作，例如初始化（构造函数及构造函数的重载）、求三角形的周长、面积、判断该三角形是什么三角形（等边、等腰、直角、等边直角、一般三角形、非三角形）、输出等，具体说明如下：

```
class Triangle{
public:
    Triangle(double,double,double); //构造函数及构造函数的重载，三条边为
给定值
    Triangle(double,double); //两条边为给定值，第三条边为固定值
    Triangle(double); //一条边为给定值，另外两条边为固定值
    Triangle(); //三条边均为固定值
    double GetArea(); //返回三角形面积
    double GetPerimeter(); //返回三角形周长
    void Tri_Type();//输出三角形类型
```

```
void Show();//输出三角形面积、周长及三角形类型  
private:  
    double a,b,c; //三角形三条边，规定三条边长度范围为[1,200]  
};
```

(2) 定义一个 Dog 类。

该类包含 name、age、sex、weight 等属性，以及对这些属性操作的方法，并测试这个类。

2.进阶部分

定义日期类型 Date。

要求：

(1) 包含年月日数据成员；提供默认构造函数、带年月日参数的构造函数；定义 set 成员函数可以设置日期；定义 input 成员函数，输入日期信息，输入格式为“YYYY-MM-DD”；定义 print 成员函数输出对象的日期。

(2) 编写一个可运行的完整程序，构造几个日期类型的对象，并输出相应的日期信息以对程序进行测试。

实验 9 类和对象（2）

实验目的

- 巩固构造函数、析构函数的定义使用方法。
- 学习拷贝构造函数方法。
- 对象数组的使用。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

定义日期类型 `Date`。

要求：

（1）包含年月日数据成员；提供默认构造函数、带年月日参数的构造函数、和拷贝构造函数；定义 `set` 成员函数可以设置日期；日期加一天操作；定义 `input` 成员函数，输入日期信息，输入格式为“YYYY-MM-DD”；定义 `print` 成员函数输出对象的日期。

2.进阶部分

（1）在完成基础部分的基础上去实现（2）。

（2）分别定义不同的日期对象，定义对象数组分别用不同的对象初始化，使用不同的数组元素。

实验 10 类和对象 (3)

实验目的

- 学习友元函数及友元类，领会友元函数的作用。
- 学习运算符重载的概念和方法，掌握基本运算符重载方法。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

设计实现一个分数类 Fraction。要求：

- (1) 包含分子 num，分母 den 数据成员；提供默认构造函数。
- (2) 可以调用 standardize() 方法，对分数类对象进行标准化。一个标准的 Fraction 表示的分数必须满足如下条件：1. 分母永远为正，分数的符号由分子表示；2. 分子分母互质，即总表示为最简分数。
- (3) 对基本的运算符进行重载，能够完成分数的加减乘除等运算，能使用输入输出流输出。

实验 11 继承和派生

实验目的

- 学习根据现有类建立新类的方法。
- 理解继承的含义，掌握派生类的定义方法和实现；
- 理解公有继承下基类成员对派生类成员和派生类对象的可见性，能正确地访问继承层次中的各种类成员；
- 理解保护成员在继承中的作用，能够在适当的时候选择使用保护成员以便派生类成员可以访问基类的部分非公开的成员；
- 了解虚基类的用法。

实验要求

- 基础部分实验必选，进阶部分实验（1）必选，其它任选。
- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

编写一个程序 `Ex_Student`，以实现一个班级学生的学期成绩管理。个人信息类 `Cperson` 的数据成员有姓名、年龄、性别；课程类 `CCourse` 的数据成员有课程名、学分、成绩；学生类 `CSutudent` 从 `CPerson` 类派生，并增加数据成员学号、`CCourse` 对象成员数组（大小为 8）、总成绩、总学分和平均成绩；班级类 `CClass` 类的数据成员有 `CStudent` 类对象数组（大小由构造函数决定）、班号、班级人数等。

要求：

- （1）本班级类 `CClass` 中添加成员函数 `Seek`，通过学号查找并输出学生的全部信息。
- （2）构造三个学生的数据，每个学生都有三门课程成绩。输入并显示这些数据。
- （3）根据类的需要添加适当的其他成员，编写完整的程序并测试。

2.进阶部分

(1) 对程序 `Ex_Student` 进行扩展，定义导师类 `CTutor`，继承 `CPerson` 类，并增加数据成员职称；定义研究生类 `CGraduateStudent` 继承学生类，并且增加数据成员导师类对象，根据需要添加适当的其他成员，编写完整程序并测试。

(2) 定义时间类型 `Time`

要求：

继承 `Date` 类，并增加数据成员时分秒；增加构造函数、拷贝构造函数；增加设置时间的成员函数；编写完整的程序，创建时间对象，并输出时间。

实验 12 多态

实验目的

- 理解虚函数在类的继承层次中的作用，虚函数的引入对程序运行时的影响，能够对使用虚函数的简单程序写出程序结果。
- 理解抽象类的作用和使用方法。
- 理解重载运算符的作用，学会对典型的运算符进行重载。

实验要求

- 基础部分实验必选，进阶部分实验任选。
- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

(1) 编写程序 Ex_Shape: 定义一个类 CShape, 包含两个虚函数 Area (用来计算面积) 和 SetData (用来设计形状大小)。然后派生出矩形 CRect 类、圆 CCircle 类, 分别求其面积。最后定义一个 CArea 类, 计算这几个形状的面积之和, 各形状的数据通过 CArea 类的构造函数或成员函数来设置。

(2) 编写程序 Ex_String: 重载字符串运算符=、+、<分别用于字符串的赋值、拼接、比较运算, 实现字符串直接操作。其中=和<运算符重载函数为友元函数, 而+运算符重载为成员函数。

2.进阶部分

(1) 仿真农场

哺乳动物类如下所示:

```
#include <iostream.h>

class Mammal
{
public:
    Mammal():itsAge(2), itsWeight(5)
    {
```

```

        cout<<"Mammal constructor...\n";
    }
    ~Mammal () {cout<<"Mammal destructor...\n";}
    int GetAge () const { return itsAge;}
    void SetAge (int age) {itsAge = age;}
    int GetWeight () const {return itsWeight; }
    void SetWeight (int weight) {itsWeight = weight;}
    void Move () const {cout << "Mammal move one step\n";}
    virtual void Speak () const {}
    void sleep () const { cout << "shhh, I'm sleeping.\n";}
protected:
    int itsAge;
    int itsWeight;
};

```

狗属哺乳动物，且它的属性有品种之分（在哺乳类的基础上增加品种资料成员），叫声区别于其它动物（**Speak（）**实现为虚函数，输出”Woof!”），还会要尾巴（增加成员函数，输出”Tail wagging...”），乞讨食物（增加成员函数，输出”begging for food...”）。

猫、马、猪也属于哺乳动物，其叫声分别为：”Meow!”，“Winnie!”，“Oink”。

编程分别使各个动物表现为不一样的行为。要求：

① 狗的品种：

enum BREED{YORKIE, CAIRN, DANDIE, SHETLAND, DORERMAN, LAB};分别用一个操作实现设置狗的品种，另一个操作实现读出狗的品种。

② 从基类 **Mammal** 类分别派生出各种动物，通过虚函数实现 **Mammal** 类动物对象指针被赋予不同的动物种类时，表现为不同的行为。

③ 今有狗 **CAIRN**：3 岁，30kg；**DORERMAN**：4 岁，28kg；**LAB**：6 岁，40kg；猫：2 岁，15kg；马：5 岁，60kg；猪：2 岁，45kg。

④ 设置一个 **Mammal** 数组，设置一个屏幕菜单，选择不同的动物或不同的品种，则实现对应的动作，直到选择结束。

⑤ 对应的动作中首先显示动物名称，然后年龄，重量，品种，叫声，以及其它特点。

思考问题：

① 继承的资料成员为私有的或保护的有什么不同，继承的成员函数总是公有的

或保护的，为什么？

② 如果上述程序的实现中，**Speak** 函数不是虚函数，那么各种动物的叫声会怎样，为什么？

③ 什么时候要用虚析构函数？

④ 面向对象程序设计中，继承机制使程序员可以在现有的类库代码基础上，稍加改造就能开发出准确性和可靠性都很高的应用程序代码来。熟悉现有的类，将现实世界中的事物分解并描述成跟现成类库相近的对象，试思考之。

实验 13 输入/输出流

实验目的

- 学习 I/O 流类的操作。
- 学习对 I/O 流进行格式控制。
- 学习文件流的使用方法。

实验要求

- 基础部分实验必选，进阶部分实验任选。
- 根据实验内容的相应描述和要求，自行设计并调试代码。

实验内容

1.基础部分

1.编写一个程序将下面的信息表写入文件 stock.txt。

Shen	fa	zhan	0000001
Shang	hai	qiche	0000006
Guang	ju	neng	0000096

2.编写程序，要求定义 in 为 fstream 的对象，与输入文件 file1.txt 建立关联，文件 file1.txt 的内容如下：

abcdef
Ghijklmn

定义 out 为 fstream 的对象，与输出文件 file2.txt 建立关联。当文件打开成功后键 file1.txt 文件的内容转换为大写字母，输出到 file2.txt 文件中。

2.进阶部分

(1) 将资料存入文件要求：

将下列格式化的资料输出到文件 xyz.txt 中。

名字	年龄	编号	工资
张明丽	29	111-22-333	425.00
李楷楷	27	333-22-1111	350.00
钟静	41	11-11-0000	650.00

定义一个 Employee 类，包含名字、年龄、编号、工资数据成员，创建三个对象保

存上述员工信息，并员工信息按照上述格式输出到文件 xyz.txt 中。

实验 14 模板

实验目的

- 熟悉函数模板的定义和使用方法。
- 熟悉类模板的定义格式、使用方法及应用。

实验要求

- 基础部分实验必选，进阶部分实验任选。
- 根据实验内容的相应描述和要求，自行设计并调试代码。

实验内容

1.基础部分

1.编写一个函数模板，求数组中的最大元素，试写出调用此函数模板的完整程序，使得函数调用时，数组的类型可以使整型也可以使双精度类型。

2.建立一个用来实现求 3 个数和的类模板（将成员函数定义在类的内部）并写出调用此类模板的完整程序。

2.进阶部分

将基础部分第二题，改写为在类模板外定义各成员函数。

实验 15 异常处理和命名空间

实验目的

- 熟悉异常处理机制的一般定义和使用方法，重抛异常的方法，多种异常类型的处理。
- 熟悉命名空间的定义及外部空间访问其成员的方法，熟悉使用命名空间成员的简化机制。

实验要求

- 基础部分实验必选，进阶部分实验任选。
- 根据实验内容的相应描述和要求，自行设计并调试代码。

实验内容

1.基础部分

1. 1 调试教材 13.4

定义一个异常类 A,其成员函数 `report()` 显示异常类型。定义一个函数 `fun()` 抛出异常，在主函数 `try` 语句块中调用 `fun()`，在 `catch` 语句块中捕获异常，观察程序执行流程。。

2.进阶部分

编写一个栈的类模板，其成员函数 `push()`完成入栈操作，成员函数 `pop()`完成出栈操作。出栈时检查栈是否为空，栈空要给出警告信息；入栈时检查栈是否满，栈满要给出警告信息，使用异常类的继承关系处理在栈操作中可能遇到的异常。

部分参考答案

实验 1 认识 Visual C++6.0 开发环境

实验内容

2.进阶部分

- (1) 参考教材第 6 页例 1.2 代码，编写一个 C++ 程序：输入圆的半径，输出其周长。

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

int main()
{
    double radius;

    cout << "请输入圆的半径： ";
    cin >> radius;

    cout << "圆的周长为： " << 2*3.14*radius << endl;

    return 0;
}
```

- (2) 实现一个多行输出的程序，输出内容为：

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

int main()
{
```

```

    cout << "*" << endl;
    cout << "****" << endl;
    cout << "*****" << endl;
    cout << "*****" << endl;
    return 0;
}

```

或者：（使用了循环的语法，后面会详细讲）

```

#include <iostream>
using namespace std;

int main()
{
    for(int i=1;i<=4;i++)
    {
        for(int j=1;j<=2*i-1;j++)
        {
            cout << "*";
        }
        cout<<endl;
    }
    return 0;
}

```

实验 2 数据类型、运算符和表达式

实验内容

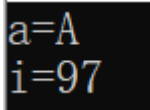
1.基础部分

(1) 测试基本数据类型 char 和 int 之间的相互转换。

设有部分代码如下：

```
a = 65;  
i='a';  
cout << "a=" << a <<endl;  
cout << "i=" << i <<endl;
```

如果希望得到如下的输出结果：



```
a=A  
i=97
```

请完善代码，（对变量 a 和 b 的定义）。

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
    char a;  
    int i;  
    a = 65;  
    i='a';  
    cout << "a=" << a <<endl;  
    cout << "i=" << i <<endl;  
  
    return 0;  
}
```

(4) 编写程序，求圆的周长、圆面积、圆球体积、圆柱体积。要求用 const 设定 PI 常量，定义适当数据类型的变量，并设圆、球半径和圆柱的高的初值分别为 2.5、4，依次计算上述结果并输出，输出时要求有相应的文字提示。

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

```
int main()  
{
```

```

const double PI=3.14;
double radius=2.5, height=4;

cout << "圆的周长为: " << 2*PI*radius << endl;
cout << "圆的面积为: " << PI*radius*radius << endl;
cout << "圆球的体积为: " << 3*PI*radius*radius*radius/4 << endl;
cout << "圆柱的体积为: " << PI*radius*radius*height << endl;

return 0;
}

```

(5) 试设计程序，实现：从键盘输入一个三位数，从左到右用 a、b、c 表示各位数字，记为 abc，现要求依次输出从右到左的各位数字，即输出另一个三位数 cba。例如，输入 123，输出 321。

```

#include <iostream>
using namespace std;

int main()
{
    int i,a,b,c;
    cout << "请输入一个三位整数: ";
    cin >> i;

    a=i/100;
    b=i%100/10;
    c=i%10;

    cout <<c<<b<<a;

    return 0;
}

```

2.进阶部分

(1) 用 sizeof 运算符编写一个测试程序，测试本机中各基本数据类型或字符串所占的字节数，并将其填写到下表中，然后分析其结果。

基本数据类型	所占字节数	基本数据类型或字符串	所占字节数
char	1	float	4
short	2	double	8
int	4	long double	12
long	4	“\nCh\t\v\0ina”	10
long long	8	unsigned short	2

(2) 已知 a=b=100000，c=1000，编程求 a*b/c，注意不要让中间结果溢出。

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

int main()
{
    int a,b;
    a=b=100000;
    int c=1000;

    cout<<(long long)a*b/c<<endl;
    //先将 a 强制类型转换为更大的类型 longlong，之后的计算都是在 longlong
    类型下进行。

    return 0;
}
```


实验3 语句和预处理(分支语句)

实验内容

1.基础部分

- (1) 利用肥胖程度计算公式 BMI 法自我诊断肥胖程度。BMI 法如下：

体重指数 = 体重（公斤） 除 身高（米）的平方。即 kg/m^2 。

正常体重：体重指数 = 18 ~ 25

超重：体重指数 = 25 ~ 30

轻度肥胖：体重指数 > 30

中度肥胖：体重指数 > 35

重度肥胖：体重指数 > 40

编写程序，输入体重和身高，输出肥胖程度。

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

int main()
{
    double weight,height;
    cout<<"请输入体重（单位：千克）："<<endl;
    cin >> weight;
    cout<<"请输入身高（单位：米）："<<endl;
    cin >> height;

    double bmi;
    bmi = weight/(height*height);

    if(bmi<18)
        cout<<"体重偏轻"<<endl;
    else if(bmi<=25)
        cout<<"体重正常"<<endl;
    else if(bmi<=30)
        cout<<"轻度肥胖"<<endl;
    else if(bmi<=35)
        cout<<"中度肥胖"<<endl;
    else if(bmi<=40)
        cout<<"重度肥胖"<<endl;
    else
        cout<<"胖出天际……"<<endl;
```

```
return 0;
```

```
}
```

(2) 给出一个百分制成绩，要求输出成绩等级 A、B、C 和 D。其中，85 分以上为 A，75～84 分为 B，65～74 分为 C，65 分以下为 D。

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int score;
```

```
    cout << "请输入成绩: ";
```

```
    cin >> score;
```

```
    cout << "该成绩对应的等级为: ";
```

```
    if(score>=85)
```

```
        cout<<"A"<<endl;
```

```
    else if(score>=75)
```

```
        cout<<"B"<<endl;
```

```
    else if(score>=65)
```

```
        cout<<"C"<<endl;
```

```
    else
```

```
        cout<<"D"<<endl;
```

```
    return 0;
```

```
}
```

(3) 求解一元二次方程 $ax^2+bx+c=0$ 的根。当 $a=0$ 且 $b=0$ 时，方程无根。当 $a=0$ 且 $b \neq 0$ 时，方程有一个根。当 $a \neq 0$ 时，根据 $\Delta=b^2-4ac$ 确定方程的根，若 $\Delta>0$ ，有两个不同的实根；当 $\Delta=0$ ，有两个相同的实根；当 $\Delta<0$ ，有两个不同的复根。输入参数 a、b、c，输出相应的结果。

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    double a,b,c;
```

```
    cout << "请分别输入方程的三个系数（以空格作为分隔符）: ";
```

```
    cin >> a>>b>>c;
```

```
    if(a==0)
```

```
    {
```

```
        if(b==0)
```

```
            cout<<"方程无根"<<endl;
```

```
        else
```

```
            cout<<"方程有一个根，为:"<<-c/b<<endl;
```

```
    }
```

```

else
{
    double delta;
    delta=b*b-4*a*c;
    if(delta>0)
        cout<<"方程有两个根，分别为："<<(-b+sqrt(delta))/(2*a)<<"和
"<<(-b-sqrt(delta))/(2*a)<<endl;
    else if(delta==0)
        cout<<"方程有一个根，为："<<-b/(2*a)<<endl;
    else
        cout<<"方程无根"<<endl;
}

return 0;
}

```

2.进阶部分

(1) 实现一个简单的菜单程序，运行时显示“Menu: A(dd) D(delete) S(ort) Q(uit), Select one:”提示用户输入，A表示增加，D表示删除，S表示排序，Q表示退出，输入为A，D，S时分别提示“数据已经增加、删除、排序。”输入Q时程序结束。

要求：使用 switch 语句进行判断。

```

#include <iostream>

using namespace std;

int main()
{
    char s;
    cout << "Menu: A(dd) D(delete) S(ort) Q(uit), Select one: ";
    cin >> s;

    switch(s)
    {
        case 'A': cout<<"数据已经增加。"<<endl;break;
        case 'D': cout<<"数据已经删除。"<<endl;break;
        case 'S': cout<<"数据已经排序。"<<endl;break;

        default: cout<<"输入错误。"<<endl;
    }

    return 0;
}

```

(2) 编程实现输入一个整数，判断其能否被 3，5，7 整除，并输出以下信息之一：

①能同时被 3，5，7 整除；

- ②能被其中两数（要指出哪两个）整除；
- ③能被其中一个数（要指出哪一个）整除；
- ④不能被 3，5，7 任一个整除。

要求：使用 if-else 语句。

```
#include <iostream>

using namespace std;

int main()
{
    cout<<"请输入待判断的整数: ";
    int a;
    cin>>a;

    if(a%3&&a%5&&a%7)
        cout<<"不能被 3，5，7 任一个整除"<<endl;
    else if(a%3&&a%5)
        cout<<"只能被 7 整除"<<endl;
    else if(a%5&&a%7)
        cout<<"只能被 3 整除"<<endl;
    else if(a%3&&a%7)
        cout<<"只能被 5 整除"<<endl;
    else if(a%3)
        cout<<"能被 5 和 7 整除"<<endl;
    else if(a%5)
        cout<<"能被 3 和 7 整除"<<endl;
    else if(a%7)
        cout<<"能被 3 和 5 整除"<<endl;
    else
        cout<<"能同时被 3，5，7 整除"<<endl;

    return 0;
}
```

实验 4 语句和预处理(循环语句)

实验目的

- 掌握表达式中逻辑运算符、关系运算符的功能和特点。
- 掌握 while 和 do-while 循环的区别。
- 掌握 for 循环的语法结构并用其解决问题。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

(1) 编写程序用于解 Fibonacci 数列问题。Fibonacci 数列的前两个数是 1 和 1，从第三个数开始，每个数等于前两个数之和。编程计算并输出此数列的前 30 个数，且每行输出 5 个数。

```
#include <iostream>

using namespace std;

int main()
{
    int a=1,b=1,c;
    cout << a << "\t" << b << "\t";
    for(int i=3;i<=30;i++)
    {
        c=a+b;
        cout<< c<< "\t";
        a=b;
        b=c;
        if(!(i%5))
            cout<<endl;
    }

    return 0;
}
```

(2) 编写程序用泰勒级数求 e 的近似值，直到最后一项小于 10^{-6} 为止。

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

```
#include <iostream>

using namespace std;

int main()
{
    double term=1,e=1,f=1,n=1;
    while(term>=1e-6)
    {
        for(int i=1;i<=n;i++)
        {
            f=f*i;
        }
        term=1/f;
        e=e+term;
        n++;
        f=1;
    }
    cout<<"e="<<e<<endl;

    return 0;
}
```

(3) 编写程序打印下列菱形图案:

```

      *
    * * *
  * * * * *
* * * * * * *
  * * * * *
    * * *
      *
```

```
#include <iostream>

using namespace std;

int main()
{
    for(int i=1;i<=7;i++)
    {
```

```

        if(i<=4)
        {
            for(int j=7;j>2*i-1;j--)
                cout<<" ";
            for(int j=1;j<=2*i-1;j++)
                cout<<"* ";
            cout <<endl;
        }
        else
        {
            for(int j=0;j<=2*(i-4)-1;j++)
                cout<<" ";
            for(int j=2*(8-i)-1;j>=1;j--)
                cout<<"* ";
            cout <<endl;
        }
    }

    return 0;
}

```

2.进阶部分

(1) 对运行中输入的 x ，计算级数：

$$1 + x - x^2/2! + x^3/3! - \dots (-1)^{n+1} * x^n/n!。$$

要求输出精度为 10^{-8} 。

```

#include <iostream>
#include <cmath>

```

```

using namespace std;

```

```

int main()
{
    double x;
    cout<<"请输入 x 的值: ";
    cin>>x;
    double term=1,sum=1+x,factorial=1,n=2,power=x;
    while(abs(term)>=1e-8)
    {
        factorial=factorial*n;

```

```

        power=power*x;
        term=-1*pow(-1,n)*power/factorial;
        sum=sum+term;
        cout<<term<<endl;
        n++;
    }
    cout<<"级数的值为: "<<sum<<endl;

    return 0;
}

```

(2) 编程求所有的 3 位素数，且该素数是对称的，所谓“对称”是指一个数倒过来还是该数。例如，375 不是对称数，因为倒过来变成了 573。

```

#include <iostream>
#include <cmath>

using namespace std;

int main()
{
    for(int i=100;i<=999;i++)
    {
        if(i/100==i%10)
        {
            int isPrime=1;
            for(int j=2;j<sqrt(i);j++)
            {
                if(i%j==0)
                {
                    isPrime=0;
                    break;
                }
            }
            if (isPrime)
                cout<<i<<" ";
        }
    }

    return 0;
}

```

(3) 编程打印九九乘法表：

*	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2		4	6	8	10	12	14	16	18
3			9	12	15	18	21	24	27

4		16	20	24	28	32	36
5			25	30	35	40	45
6				36	42	48	54
7					49	56	63
8						64	72
9							81

```
#include<iostream>
using namespace std;
int main()
{
    cout << '\t' << '*';
    for (int i = 1; i <= 9; i++)
        cout << '\t' << i;
    cout << endl;
    for (int i = 0; i < 89; i++)
        cout << '-';
    cout << endl;
    for (int a = 1; a <= 9; a++)
    {
        cout << '\t' << a;
        for (int b = 1; b <= 9; b++)
        {
            if(a<=b)
                cout << '\t' << a * b;
            else
                cout << '\t';
        }
        cout << endl << endl;
    }
    return 0;
}
```

实验 5 函数（1）

实验目的

- 掌握函数声明、定义和使用的方法。
- 掌握函数递归调用的方法。
- 掌握重载函数及函数参数默认的使用方法。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

（1）程序求三角形的面积。已知三角形的三边 a 、 b 、 c ，则三角形的面积为

$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

其中， $s=(a+b+c)/2$ 。需要说明的是，三角形的三边的边长由 `cin` 输入，需要判断这三边是否构成一个三角形（要判断输入的三条边能否够成三角形，只需满足任意两边之和都大于第三边即可）。若是，则计算其面积并输出，否则输出“错误：不能构成三角形！”。编写一个完整的程序，其中需要两个函数，一个函数判断是否构成三角形，另一个函数计算三角形的面积。

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

double area(double x,double y,double z);
bool isTriangle(double x,double y,double z);

int main()
{
    double a,b,c;
    cout<<"请输入三边长：（以空格分隔）";
    cin>>a>>b>>c;

    if(isTriangle(a,b,c))
        cout<<"三角形面积为： "<<area(a,b,c)<<endl;
    else
```

```

        cout<<"不构成三角形"<<endl;

        return 0;
    }
    bool isTriangle(double x,double y,double z)
    {
        if(x+y>z&& x+z>y&&y+z>x)
            return true;
        else
            return false;
    }

    double area(double x,double y,double z)
    {
        double s;
        s=(x+y+z)/2;
        return sqrt(s*(s-x)*(s-y)*(s-z));
    }
}

```

(2) 编程求下式值，其中 n^i 用函数来实现，且设参数 n 的默认值为 2:

$$n^1 + n^2 + n^3 + n^4 + \cdots + n^{10} \quad n = 1, 2, 3$$

```

#include <iostream>

using namespace std;

int fun(int n=2);

int main()
{
    cout<<fun(1)<<endl;
    cout<<fun()<<endl;
    cout<<fun(3)<<endl;

    return 0;
}

int myPow(int i,int n)
{
    int s=1;
    for(int j=1;j<=i;j++)
    {
        s=s*n;
    }
    return s;
}

```

```

    }

int fun(int n)
{
    int s=0;
    for(int i=1;i<=10;i++)
    {
        s=s+myPow(i,n);
    }
    return s;
}

```

（3）用递归法将一个整数 n 转换成字符。如输入 1234，应输出字符 ‘1’ ‘2’ ‘3’ ‘4’。
 n 的位数不确定，可以是任意位数的整数。

```

#include<iostream>
using namespace std;

int main()
{
    void convert(int);
    int m;
    cout<<"请输入一个整数:"<<endl;
    cin>>m;

    if(m<0)
    {
        cout<<"-";
        m=-m;
    }
    convert(m);
    cout<<endl;
    return 0;
}

void convert(int m)
{
    int i;
    char c;
    if((i=m/10)!=0)
        convert(i);
    c=m%10+'0';
    cout<<c;
}

```

2.进阶部分

(1) 设计一个程序，通过重载求两个数中最大数的函数 `max()`，分别实现求两个实数和两个整数及两个字符的最大者。

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

char max(char a,char b);
int max(int a,int b);
double max(double a,double b);
int main()
{
    cout<<"请输入两个整数： ";
    int a,b;
    cin>>a>>b;
    cout<<"大的那个是： "<<max(a,b)<<endl;
    cout<<"请输入两个实数： ";
    double c,d;
    cin>>c>>d;
    cout<<"大的那个是： "<<max(c,d)<<endl;
    cout<<"请输入两个字符： ";
    char e,f;
    cin>>e>>f;
    cout<<"大的那个是： "<<max(e,f)<<endl;
    return 0;
}

char max(char a,char b)
{
    return (a>b)?a:b;
}
int max(int a,int b)
{
    return (a>b)?a:b;
}
double max(double a,double b)
{
    return (a>b)?a:b;
}
```

(2) 设计一个函数，要求能将一个正整数 `n` 按反序输出，`n` 的位数不定。例如，123 输出 321。

要求：用非递归函数来实现，编写完整的程序并测试。

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

```

void reverse(int a);

int main()
{
    int a;
    cout<<"请输入一个整数: ";
    cin>>a;

    reverse(a);

    return 0;
}

void reverse(int a)
{
    int i=a;
    while(i!=0)
    {
        cout<<i%10;
        i=i/10;
    }
}

```

（3）用递归方法求解母牛问题。若一头母牛，从出生起第四个年头开始每年生一头母牛，按此规律，第 n 年时有多少头母牛？

要求：用递归函数求解；先测试递归条件，后递归调用；运行时 n 由键盘输入。

```

#include<iostream>
using namespace std;

int f(int N);

int main()
{
    int N;

    cin>>N;
    cout<<f(N);
    return 0;
}

```

```
}  
int f(int N)  
{  
    if (N < 4)  
        return 1;  
    else  
        return f(N - 1) + f(N - 3);  
}
```

注：递归的思路，今年的牛的数量等于去年的牛的数量加上去年的牛生的小牛的数量，又因为只有第四个年头的牛能生产，所以去年生的小牛的数量即为三年前的牛的数量。

实验 6 指针和引用

实验目的

- 学习指针的定义、初始化、赋值、使用的方法。
- 声明和操作指针和引用。
- 学习调试方法。
- 学习数组的定义与使用

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

- (1) 程序中分别定义不同的变量,定义指针分别指向不同变量,实现变量值的交换。

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

int main()
{
    int a=3,b=5;
    int *pA,*pB;
    pA=&a;
    pB=&b;

    int t;
    t=*pA;
    *pA=*pB;
    *pB=t;

    cout<<a<<" "<<b<<endl;
```



```

        return 0;
    }

```

- (2) 程序中分别定义不同的变量,定义引用分别指向不同变量,实现变量值的交换。

```

#include<iostream>
using namespace std;

int main()
{
    int a=3,b=5;
    int &rA=a,&rB=b;

    int t;
    t=rA;
    rA=rB;
    rB=t;

    cout<<a<<" "<<b<<endl;

    return 0;
}

```

2.进阶部分

- (1) 输入 10 个数到一维数组中,用冒泡排序法将其按从大到小的次序排序后输出。
要求: 分别用三个函数实现数据的输入、排序及输出。

```

#include<iostream>
using namespace std;

void input(int *a);
void output(int *a);
void sort(int *a);
int main()
{
    int a[10];

```

```

        input(a);
        sort(a);
        output(a);

        return 0;
    }

void input(int *a)
{
    cout<<"请输入 10 个整数: ";
    for(int i=0;i<10;i++)
    {
        cin>>a[i];
    }
}

void output(int *a)
{
    cout<<"排序后的结果为: ";
    for(int i=0;i<10;i++)
    {
        cout<<a[i]<<" ";
    }
    cout<<endl;
}

void sort(int *a)
{
    for(int i = 0; i < 10; i++)
    {
        for(int j = 0; j < 10 - i - 1; j++)
        {
            if(a[j] < a[j + 1])
            {
                int t;

```

```

        t=a[j];
        a[j]=a[j+1];
        a[j+1]=t;
    }
}

}

```

(2) 输入一个 5*5 的整数矩阵，求出两条对角线元素值之和。

```

#include<iostream>
using namespace std;

const int N=5;

void input(int (*a)[N]);
int sumOfDiagonal(int (*a)[N]);
int main()
{
    int a[N][N];
    input(a);

    cout<<"对角线元素和为: "<<sumOfDiagonal(a);

    return 0;
}

void input(int (*a)[N])
{
    cout<<"请输入矩阵元素: ";
    for(int i=0;i<N;i++)
        for(int j=0;j<N;j++)
        {
            cin>>a[i][j];
        }
}

```

```

int sumOfDiagonal(int (*a)[N])
{
    int sum=0;
    for(int i=0;i<N;i++)
        sum=sum+a[i][i];
    return sum;
}

```

(3) 编写一个函数，统计一条英文句子中字母的个数，在主程序中实现输入、输出。

```

#include <iostream>
#include <stdio.h>

using namespace std;

const int N=200;

int count(char *s);
int main()
{
    char sentence[N];

    int i=0;
    char c=getchar();
    while(c!='.')
    {
        sentence[i]=c;
        c=getchar();
        i++;
    }
    sentence[i]='\0';

    cout<<"句子中字母的个数为: "<<count(sentence)<<endl;

    return 0;
}

```

```
int count(char *s)
{
    int i=0;
    int num=0;
    while(s[i]!='.')
    {
        if(s[i]>=65&&s[i]<=90||s[i]>=97&&s[i]<=122)
            num++;
        i++;
    }
    return num;
}
```

实验 7 函数 (2)

实验目的

- 掌握指针做函数参数的用法。
- 掌握重载函数的定义与使用方法。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础实验

(1) 编程实现下述功能：定义 add 函数实现 2 个同数据类型数据的加（可以是实数+实数，也可能是整数+整数），并定义主函数调用 add 函数。

```
#include <iostream>

using namespace std;

int add(int a, int b);
double add(double a, double b);

int main()
{
    int a=10,b=20;
    double c=1.5, d=3.14;

    cout<<add(a,b)<<endl;
    cout<<add(c,d)<<endl;

    return 0;
}
int add(int a, int b)
{
```

```

        return a+b;
    }
double add(double a, double b)
{
    return a+b;
}

```

(2) 编程实现数据交换。分别用指针和引用作为函数 mySwap 的形参，并通过重载 int、double 型变量值的交换。

```

#include <iostream>

using namespace std;

void mySwap(int *p1, int *p2);
void mySwap(double *p1, double *p2);
void mySwap(int &r1, int &r2);
void mySwap(double &r1, double &r2);
int main()
{
    int a=10,b=20;
    double c=1.5, d=3.14;

    mySwap(&a,&b);
    mySwap(&c,&d);

    cout<<a <<" "<<b<<endl;
    cout<<c <<" "<<d<<endl;

    mySwap(a,b);
    mySwap(c,d);

    cout<<a <<" "<<b<<endl;
    cout<<c <<" "<<d<<endl;
    return 0;
}

```

```
void mySwap(int *p1, int *p2)
{
    int t;
    t=*p1;
    *p1=*p2;
    *p2=t;
}
```

```
void mySwap(double *p1, double *p2)
{
    double t;
    t=*p1;
    *p1=*p2;
    *p2=t;
}
```

```
void mySwap(int &r1, int &r2)
{
    int t;
    t=r1;
    r1=r2;
    r2=t;
}
```

```
void mySwap(double &r1, double &r2)
{
    double t;
    t=r1;
    r1=r2;
    r2=t;
}
```


2.进阶部分

(1) 字符串交换

编写程序，使用指针作为参数通过函数调用的方式交换两个不同长度的字符串。

要求：主函数中定义两个字符串，然后调用交换函数，打印交换前和交换后字符串。

```
#include <iostream>

using namespace std;

void mySwap(char *p1, char *p2);

int main()
{
    char s1[50]="Hello word";
    char s2[50]="I am a student";

    mySwap(s1,s2);
    cout<<s1<<endl;
    cout<<s2<<endl;

    return 0;
}

void mySwap(char *p1, char *p2)
{
    char s[50];
    int i=0;
    while(p1[i]!='\0')
    {
        s[i]=p1[i];
        i++;
    }
    s[i]='\0';
    i=0;
    while(p2[i]!='\0')
```

```

    {
        p1[i]=p2[i];
        i++;
    }
    p1[i]='\0';
    i=0;
    while(s[i]!='\0')
    {
        p2[i]=s[i];
        i++;
    }
    p2[i]='\0';

}

```

(2) 设计一个函数，形参是引用类型，要求输入三个整数，按从小到大排序后输出。编程完整的程序并测试。

```

#include <iostream>

using namespace std;

void mySort(int &a,int &b,int &c);

int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    mySort(a,b,c);
    cout<<a<<" "<<b<<" "<<c<<endl;

    return 0;
}

void mySort(int &a,int &b,int &c)
{
    int t;

```

```
    if(a>b)
    {
        t=a;a=b;b=t;
    }
    if(b>c)
    {
        t=b;b=c;c=t;
    }
    if(a>b)
    {
        t=a;a=b;b=t;
    }
}
```

实验 8 类和对象（1）

实验目的

- 学习类与对象的定义和声明；
- 掌握类与对象的使用方法；
- 学习数据成员与成员函数的访问方式。学习定义类的方法。
- 学习构造函数、析构函数的定义使用方法。
- 学习对象的使用方法。
- 领会面向对象程序设计的基本方法。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

（1）应用 VC++6.0 的编辑环境构造一个类 **Triangle**，该类主要实现三角形的基本操作，例如初始化（构造函数及构造函数的重载）、求三角形的周长、面积、判断该三角形是什么三角形（等边、等腰、直角、等边直角、一般三角形、非三角形）、输出等，具体说明如下：

```
class Triangle{
public:
    Triangle(double,double,double); //构造函数及构造函数的重载，三条边为给定值
    Triangle(double,double); //两条边为给定值，第三条边为固定值
    Triangle(double); //一条边为给定值，另外两条边为固定值
    Triangle(); //三条边均为固定值
    double GetArea(); //返回三角形面积
    double GetPerimeter(); //返回三角形周长
    void Tri_Type();//输出三角形类型
    void Show();//输出三角形面积、周长及三角形类型
private:
```

```

double a,b,c; //三角形三条边，规定三条边长度范围为[1,200]
};

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

class Triangle{
public:
    Triangle(double,double,double); //构造函数及构造函数的重载，三条边为给定值
    Triangle(double,double); //两条边为给定值，第三条边为固定值
    Triangle(double); //一条边为给定值，另外两条边为固定值
    Triangle(); //三条边均为固定值
    double GetArea(); //返回三角形面积
    double GetPerimeter(); //返回三角形周长
    void Tri_Type(); //输出三角形类型
    void Show(); //输出三角形面积、周长及三角形类型
private:
    double a,b,c; //三角形三条边，规定三条边长度范围为[1,200]
};

Triangle::Triangle(double x,double y,double z)
{
    a=x;
    b=y;
    c=z;
}

Triangle::Triangle(double x,double y)
{
    a=x;
    b=y;
    c=5;
}

Triangle::Triangle(double x)
{
    a=x;

```

```

        b=4;
        c=5;
    }
    Triangle::Triangle()
    {
        a=3;
        b=4;
        c=5;
    }
    double Triangle::GetPerimeter()
    {
        return (a+b+c);
    }
    double Triangle::GetArea()
    {
        double m=GetPerimeter()/2;
        return sqrt(m*(m - a)*(m - b)*(m - c));;
    }
    void Triangle::Tri_Type()
    {

        if (a + b > c&&b + c > a&&a + c > a)
        {

            //判断三条边是否相等;
            if (a == b&&a == c)
            {
                cout<<"等边三角形\n";
            }
            //判断三角形是否有两天边相等;
            else if (a == b || a == c || b == c)
            {
                cout<<"等腰三角形\n";
            }
            //判断是否有两边的平方和大于第三边的平方;

```

```

else if ((a*a + b*b == c*c) || (a*a + c*c == b*b) || (b*b + c*c == a*a))
{
    cout<<"直角三角形\n";
}
else
{
    cout<<"一般三角形\n";
}
}
else
{
    cout<<"三边不能构成三角形\n";
}
}

```

```

void Triangle::Show()
{
    cout<<"面积为: "<<GetArea()<<endl;
    cout<<"周长为: "<<GetPerimeter()<<endl;
    Tri_Type();
    cout<<endl;
}
int main()
{
    Triangle tr1(6.0);
    Triangle tr2(5.0,4.0);
    Triangle tr3(6.0,6.0,6.0);
    Triangle tr4;
    tr1.Show();
    tr2.Show();
    tr3.Show();
    tr4.Show();
    return 0;
}

```

(2) 定义一个 Dog 类。

该类包含 name、age、sex、weight 等属性，以及对这些属性操作的方法，并测试这个类。

```
#include <iostream>

using namespace std;

class dog
{
    int age;
    double weight;
    char sex;
    string name;
public:
    dog(string pName, int pAge, char pSex, double pWeight)
    {
        name = pName;
        age = pAge;
        sex = pSex;
        weight = pWeight;
    }
    int getAge()
    {
        return age;
    }
    char getSex()
    {
        return sex;
    }
    double getWeight()
    {
        return weight;
    }
    string getName()
    {
        return name;
    }
}
```



```

    }
    void changeName(string pName)
    {
        name = pName;
    }
    void setAge(int pAge)
    {
        age = pAge;
    }
    char setSex(char pSex)
    {
        sex = pSex;
    }
    void setWeight(double pWeight)
    {
        weight = pWeight;
    }
    void showInfo()
    {
        cout<<"狗狗的信息: "<<endl;
        cout<<"name\t age\t sex\t weight"<<endl;

        cout<<getName()<<"\t"<<getAge()<<"\t"<<getSex()<<"\t"<<getWeight()<<endl;
    }
};

int main()
{
    dog dog1("Tom", 2, 'M', 5.3);
    dog1.showInfo();
    dog1.changeName("Mary");
    dog1.setAge(3);
    dog1.setSex('F');
    cout<<"修改狗狗的信息之后: "<<endl;
    dog1.showInfo();

```

```
        return 0;
    }
}
```

2.进阶部分

定义日期类型 `Date`。

要求：

(1) 包含年月日数据成员；提供默认构造函数、带年月日参数的构造函数；定义 `set` 成员函数可以设置日期；定义 `input` 成员函数，输入日期信息，输入格式为“YYYY-MM-DD”；定义 `print` 成员函数输出对象的日期。

(2) 编写一个可运行的完整程序，构造几个日期类型的对象，并输出相应的日期信息以对程序进行测试。

```
#include<iostream>
#include<string>
#include<stdlib.h>
using namespace std;
```

```
class Date{
    int year, month, day;
public:

    Date(int y=2020, int m=1, int d=1);
    void set(int y, int m, int d);
    void input(const string& s);
    void print();
};
```

```
Date::Date(int y, int m, int d)
{
    year=y, month=m, day=d;
}
void Date::input(const string& s)
{
```

//atoi(ascii to integer)把字符串转换成整型

//s.substr(a,b)是 `string` 类的成员函数，从 `s` 的第 `a` 个字符开始取 `b` 个字符的长度的子字符串

```

        //c_str 也是 string 类的成员函数，把 string 类型的字符串转换成 c 串
        year = atoi(s.substr(0,4).c_str());
        month = atoi(s.substr(5,2).c_str());
        day = atoi(s.substr(8,2).c_str());
    }
    void Date::set(int y, int m, int d)
    {
        year=y, month=m, day=d;
    }

    void Date::print()
    {
        cout<<month<<"/"<<day<<"/"<<year<<endl;
    }

    int main()
    {
        Date a;
        a.print();
        a.set(2020,2,2);
        a.print();
        Date b(2019,12,6);
        b.print();
        Date c(2019);
        c.print();
        Date d(2019,12);
        d.print();

        return 0;
    }

```

实验9 类和对象（2）

实验目的

- 巩固构造函数、析构函数的定义使用方法。
- 学习拷贝构造函数方法。
- 对象数组的使用。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

1.基础部分

定义日期类型 Date。

要求：

（1）包含年月日数据成员；提供默认构造函数、带年月日参数的构造函数、和拷贝构造函数；定义 set 成员函数可以设置日期；日期加一天操作；定义 input 成员函数，输入日期信息，输入格式为“YYYY-MM-DD”；定义 print 成员函数输出对象的日期。

```
#include<iostream>
#include<string>
#include<stdlib.h>
using namespace std;

class Date{
    int year, month, day;
    bool IsLeapYear();
public:

    Date(int y=2020, int m=1, int d=1);
    Date(Date & d);
    void set(int y, int m, int d);
    void input(const string& s);
```

```

    void print();
    Date NextDay();
};

```

```

Date::Date(Date & d) //拷贝构造函数

```

```

{
    year=d.year, month=d.month, day=d.day;
}

```

```

Date::Date(int y, int m, int d)

```

```

{
    year=y, month=m, day=d;
}

```

```

void Date::input(const string& s)

```

```

{
    //atoi(ascii to integer)把字符串转换成整型
    //s.substr(a,b)是 string 类的成员函数，从 s 的第 a 个字符开始取 b 个字符的长
度的子字符串

```

```

    //c_str 也是 string 类的成员函数，把 string 类型的字符串转换成 c 串

```

```

    year = atoi(s.substr(0,4).c_str());
    month = atoi(s.substr(5,2).c_str());
    day = atoi(s.substr(8,2).c_str());
}

```

```

void Date::set(int y, int m, int d)

```

```

{
    year=y, month=m, day=d;
}

```

```

void Date::print()

```

```

{
    cout<<month<<"/"<<day<<"/"<<year<<endl;
}

```

```

Date Date::NextDay()

```

```

{
    Date t;

```

```

t.day=day;
t.month=month;
t.year=year;
t.day++;
    if(IsLeapYear())
    {
        if(t.month == 2 && day == 30)
        {
            t.day = 1;
            t.month++;
        }
    }
    else
    {
        if(t.month == 2 && t.day == 29)
        {
            t.day=1;
            t.month++;
        }
    }
    if(t.day > 31)
    {
        if(t.month == 12)
        {
            t.day = 1;
            t.month = 1;
            t.year++;
        }
        else
        {
            t.day = 1;
            t.month++;
        }
    }
    else if(t.day == 31)

```

```

    {
        if(t.month==4||t.month==6||t.month==9||t.month==11)
        {
            t.day=1;
            t.month++;
        }
    }
    return t;
}

```

```

bool Date::IsLeapYear()
{
    bool bleap;
    if((year%4 == 0 && year%100 != 0) || year%400 == 0)
        bleap = true;
    else
        bleap = false;
    return bleap;
}

```

```

int main()
{
    Date a;
    a.print();
    Date aa(a);
    aa.print();
    a.set(2020,2,2);
    a.print();
    Date b(2019,2,28);
    b.print();
    a=b.NextDay();
    a.print();
    Date c(2019);
    c.print();
    Date d(2019,12);
    d.print();
}

```

```
    return 0;  
}
```

2.进阶部分

（1）在完成基础部分的基础上去实现（2）。

（2）分别定义不同的日期对象，定义对象数组分别用不同的对象初始化，使用不同的数组元素。

实验 10 类和对象 (3)

实验目的

- 学习友元函数及友元类，领会友元函数的作用。
- 学习运算符重载的概念和方法，掌握基本运算符重载方法。

实验要求

- 根据实验内容的相应描述和要求，自行设计并调试代码。
- 认真填写实验报告。

实验内容

设计实现一个分数类 Fraction。要求：

- (1) 包含分子 num，分母 den 数据成员；提供默认构造函数。
- (2) 可以调用 standardize() 方法，对分数类对象进行标准化。一个标准的 Fraction 表示的分数必须满足如下条件：1. 分母永远为正，分数的符号由分子表示；2. 分子分母互质，即总表示为最简分数。
- (3) 对基本的运算符进行重载，能够完成分数的加减乘除等运算，能使用输入输出流输出。

```
#include <iostream>
#include <stdlib.h>
#include <cmath>

using namespace std;
class Fraction
{
    long num;//分子
    long den;//分母
    void standardize();//分数标准化函数
public:
    Fraction(long n=0, long d=1);//构造函数声明
    Fraction operator + (Fraction b) const;//重载加法运算符 +
    Fraction & operator ++()//前缀++运算符重载
```

```

{
    num+=den;
    return *this;
}

```

Fraction operator ++(int)//后缀++运算符重载。int 是一个虚拟参数，做后缀++标记用，无实际意义

```

{
    num+=den;
    return Fraction(num-den, den); //返回的是原值
}

friend istream & operator >> (istream&, Fraction &);
friend ostream & operator << (ostream&, const Fraction f);
};

```

```

static long gcd(long m1, long m2) //求最大公约数
{
    while(m2)
    {
        long m=m1;
        m1=m2;
        m2=m%m2;
    }
    return labs(m1); //返回 m1 的绝对值
}

```

```

void Fraction::standardize()
{
    if(num==0L || den==0L) //分子或分母有为 0 的直接把分数标准化为 0/1
    {
        num = 0L;
        den = 1L;
        return;
    }
    if(den<0L) //如果分母<0，把负号移到分子上

```

```

    {
        num = -num;
        den = -den;
    }
    //保证分数为最简分数
    long g=gcd(num, den);
    num/=g;
    den/=g;
}

```

//构造函数的定义，为数据成员赋值并标准化

```

Fraction::Fraction(long n, long d):num(n), den(d)
{
    standardize();
}

```

istream & operator >>(istream& is, Fraction & f)//>>运算符重载函数定义

```

{
    char c;
    is>>f.num>>c>>f.den;
    f.standardize();
    return is;
}

```

ostream & operator << (ostream& os, const Fraction f)//<<运算符重载函数定

义

```

{
    os<<f.num<<"/"<<f.den;
    return os;
}

```

Fraction Fraction::operator + (Fraction f) const

```

{
    return Fraction(num*f.den+den*f.num, den*f.den);
}

```

int main()

```

{

```

```
Fraction f1(1,2);  
Fraction f2;  
cin>>f2;  
f2++;  
Fraction f3;  
f3=f1+f2; //赋值符号重载使用默认的就可以了。  
cout<<f3;  
  
return 0;  
}
```