

第一章 初识C++语言

第一节 C++语言简介

第二节 C++语言程序结构

第一节 C++语言简介

编程竞赛是一项益智性的竞赛活动，核心是考查选手的智力和使用计算机解题的能力，选手首先应针对竞赛题目的要求构建数学模型，进而构造出计算机可以接受的算法，之后编写出计算机能够执行的程序。程序设计是编程竞赛的基本功，选手参与竞赛活动的第一步是熟练掌握一门程序设计语言，目前竞赛中允许使用的程序设计语言有C语言、C++语言、Java等。

C++ 语言概述

- C++ 由美国AT&T贝尔实验室的本贾尼·斯特劳斯特卢普博士在20世纪80年代初期发明并实现（最初这种语言被称作“C with Classes”带类的C）。开始，C++是作为C语言的增强版出现的，从给C语言增加类开始，不断的增加新特性。
- 今天C++已成为世界主流编程语言之一。

C++ 语言的特点

语言简洁紧凑，使用灵活方便

C++语言一共只有32个关键字和9种控制语句，程序书写自由，主要用小写字母表示。

运算符丰富

C++语言的运算符包含的范围很广泛，共有34个运算符。

数据结构丰富

C++语言的数据类型有：整型、实型、字符型、数组类型等等

结构化语言

结构化语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。

生成的代码质量高

C++语言在代码效率方面可以和汇编语言相媲美。

可移植性强

C++语言编写的程序很容易进行移植，在一个环境下运行的程序不加修改或少许修改就可以在完全不同的环境下运行。

C++ 程序结构

在介绍C++语言程序的结构之前，我们先来看一个简单的例子：

例1.1 在屏幕上输出“Hello World!”。

```
#include<iostream> //使用cin,cout, 须调用iostream 库, 否则编译出错
#include<cstdlib>    // 使用system( )调用cstdlib库
using namespace std; //在C语言中要省略, 例如在VC++和TC++中
int main()          //有的C语言可用void main(),例如TC++和VC++
{
    cout<<"Hello World!"<<endl; //输出"Hello World!"
    system("pause"); //暂停作用, 使用system("pause")调用cstdlib库
    return 0;        //结束整个程序
}
```

运行结果：

Hello World!

【说明】：

1、以 “//” 开头为注释行，“//” 后的内容用以对语句进行说明，输入程序时可以不输入。

2、`#include <iostream>`

告诉编译器的预处理器将输入输出流的标准头文件(iostream)包括在本程序中。这个头文件包括了C++中定义的基本标准输入输出程序库的声明。

3、`using namespace std`

使用std(标准)名字空间的意思。所谓的名字空间是标准C++中的一种机制，用来控制不同类库的冲突问题。使用它可以在不同的空间内使用相同名字类或者函数。

4、`int main()`

这一行为主函数(main function)的起始声明。main()是所有C++程序的运行的起始点。不管它是在代码的开头，结尾还是中间，此函数中的代码总是在程序开始运行时第一个被执行。所有C++程序都必须有一个main(),int main()前int在Dev C++中可省略，在TC++和VC++中最好保留。

main后面跟了一对圆括号()，表示它是一个函数。C++中所有函数都跟有一对圆括号()，括号中可以有一些输入参数。注意，圆括号中即使什么都没有也不能省略。如例题中显示，主函数main()中的内容，由一对花括号{}括起来。

5、`cout << "Hello World!" << endl`

这个语句在本程序中最重要。cout是一个输出语句，告诉计算机把引号之间的字符串送到标准的输出设备（屏幕）上。cout 的声明在头文件iostream中，所以要想使用cout 必须将头文件iostream包括在程序开始处。endl是C++语言的换行控制符，表示内容输出后换行显示后续的内容。

6、system("pause")

在4.99以前版本的Dev-C++环境下，为了查看程序运行结果，需要在main函数的return 0语句前加上：system("pause");这样程序运行到该语句时，结果显示屏幕将会停留，让大家有时间看程序的输出结果，否则结果显示屏幕将会一闪而过。在4.99之后的版本可不必加该语句，运行结束后结果显示屏幕将自动停留。

7、return 0

主函数main()的返回语句，一般是函数的最后一条可执行语句。main()函数末尾使用return语句时，数值0表示程序顺利结束，其他数表示有异常。在后面的例子中你会看到C++程序都以类似的语句结束。

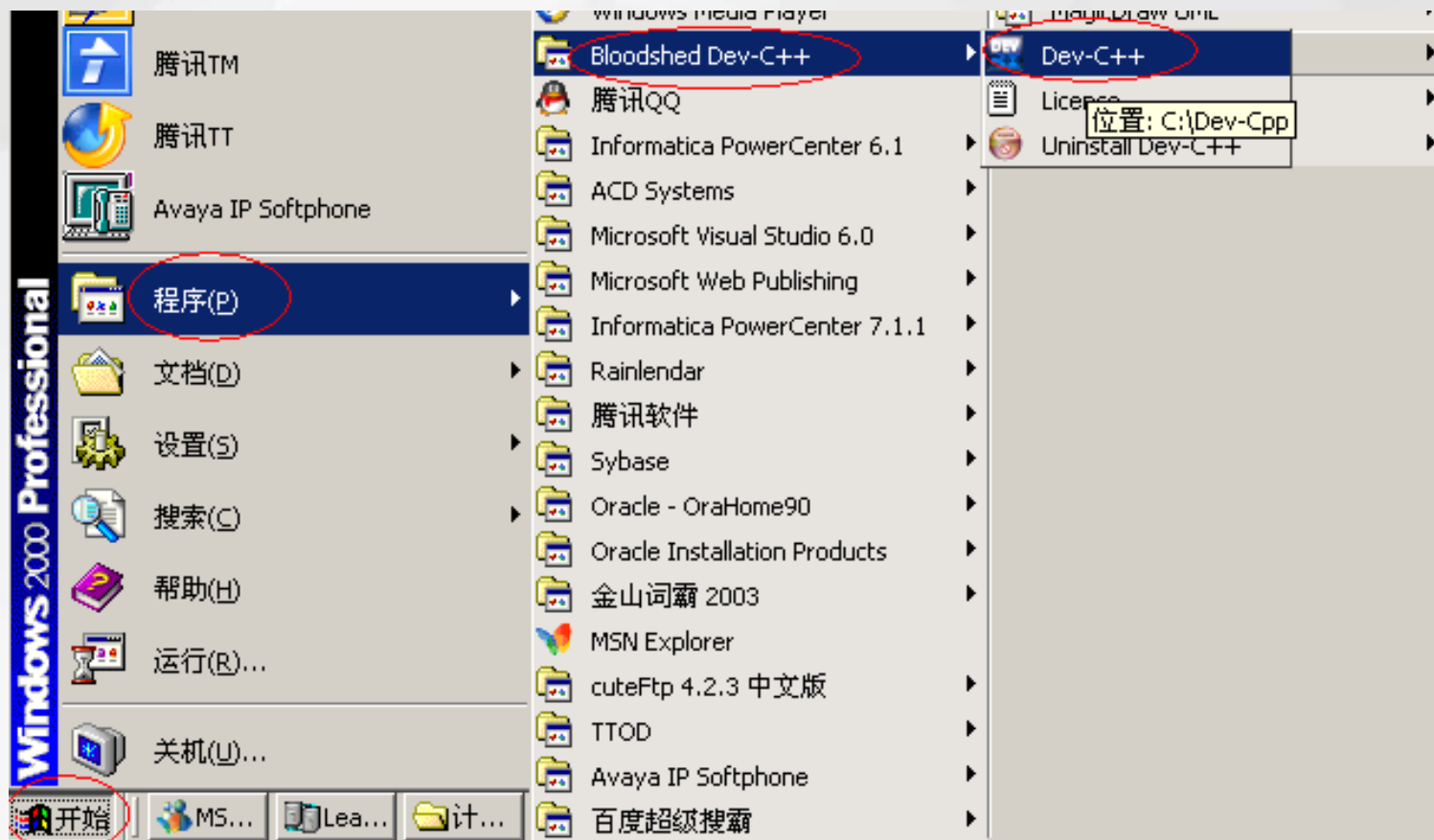
在C++中，语句的分隔是以分号“;”为分隔符的，分行写代码只是为了更方便人阅读。

C++语言系统的使用

Dev-C++是一个可视化集成开发环境，可以用此软件实现C/C++程序的编辑、预处理/编译/链接、运行和调试。现在介绍Dev-C++常用的一些基本操作，每一位同学都要掌握。

一.启动Dev-C++

1. 鼠标点击任务栏中的“开始”按钮，选“程序”菜单项，然后选“程序”下的子菜单项“Bloodshed Dev-C++”项，显示该项下的子菜单。
2. 方法一:单击“Dev-C++”菜单项，即可启动Dev-C++集成开发工具。（如图1所示）



方法二：直接双击桌面上的Dev-C++的图标。

如果大家看到界面上的字是英文的，则可以点击主菜单“Tools > Environment Options”（如图3），在弹出的对话框中选择第二个标签页“interface”（如图4），在Language下拉列表中选择Chinese即可，将操作界面改为中文的。

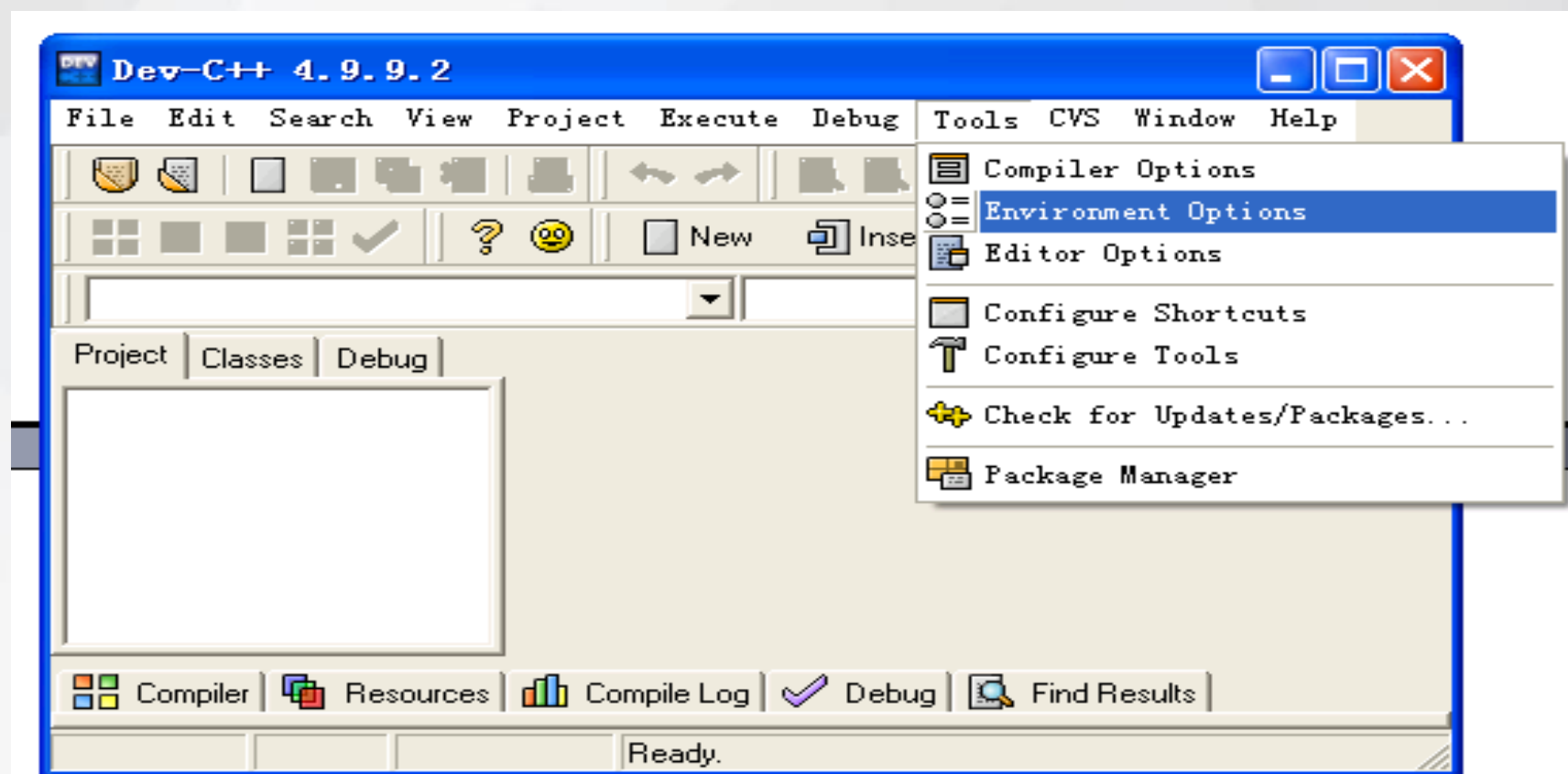


图3

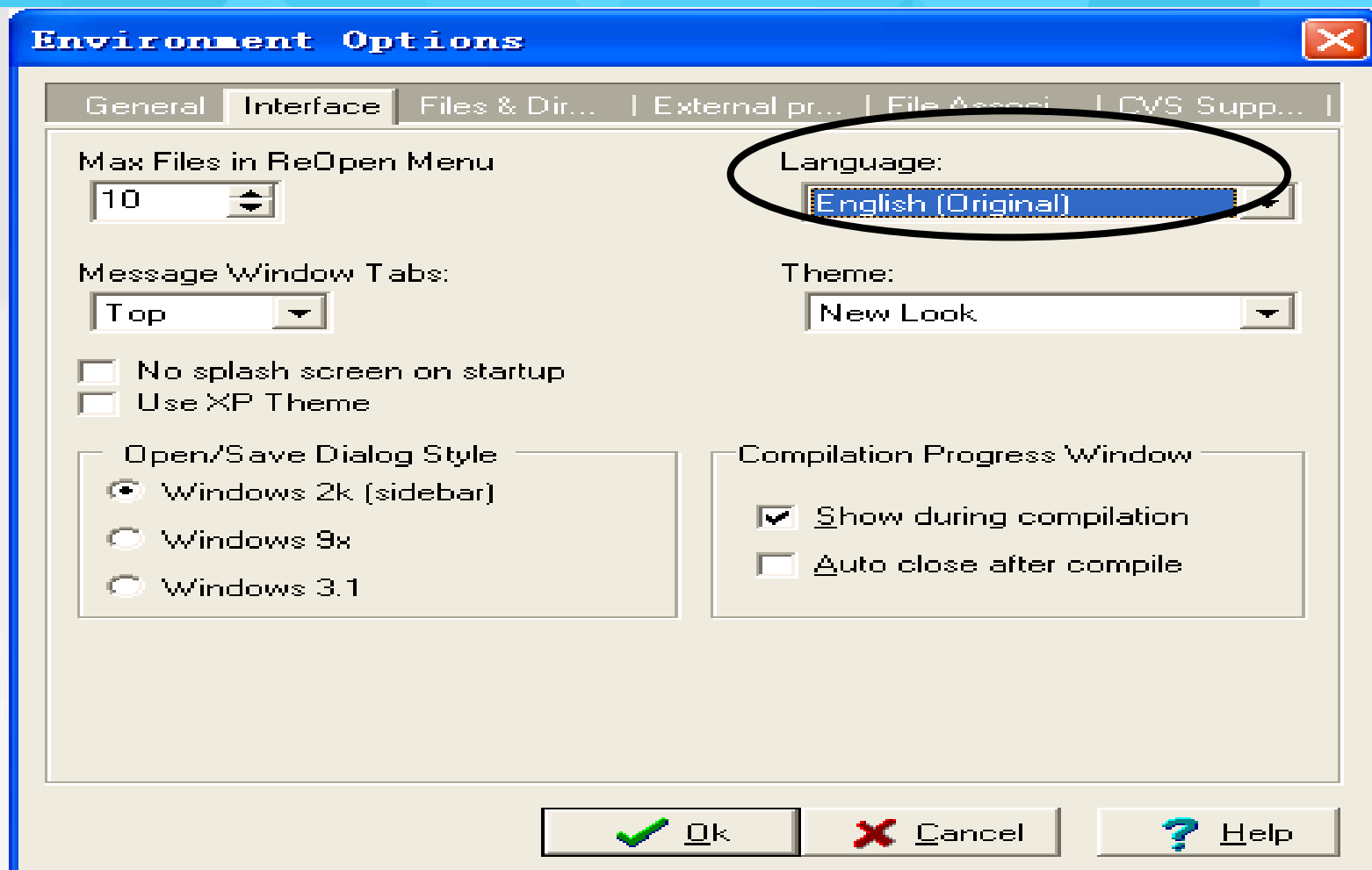


图4

二、新建源程序

1)从主菜单选择“文件”〉“新建”〉“源代码”即可（如图2所示）。

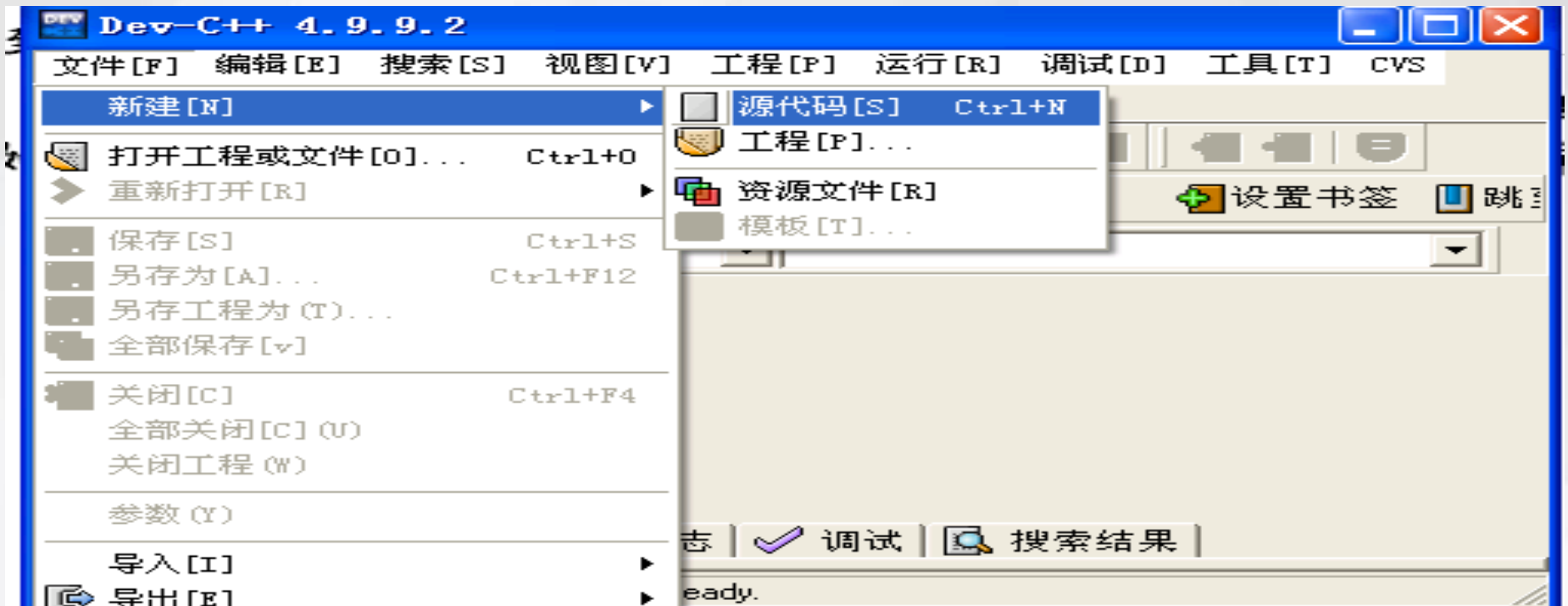
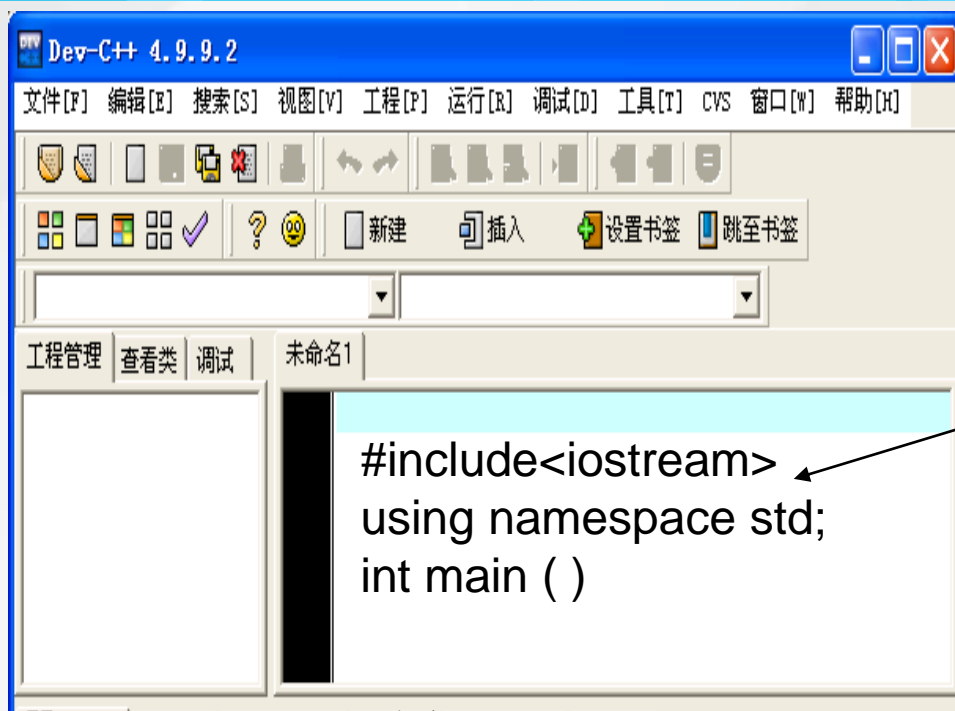


图2

2).此时屏幕右下侧出现一片白色区域，称为“源程序编辑区域”；可以在此输入程序。如(图5)所示。



源程序编辑区

图5

注意：

1. 必须在英文输入环境下编辑程序（如果你当前能在程序编辑区输入中文，说明你是在中文输入环境下。为了输入程序，你必须切换到英文输入环境下）。

2. 在4.99以前的版本的Dev-C++环境下，为了查看程序运行结果，需要在main函数的return语句前加上：`system("pause")`或`getchar()`，这样程序运行到该语句时，结果显示屏幕将会停留，让大家有时间看程序的输出结果。否则结果显示屏幕将会一闪而过。在4.99之后的版本可不必加该语句，运行结束后结果显示屏幕将自动停留。

三、保存源程序到硬盘

一个好的习惯是创建了一个新程序后，在还未输入代码之前先将该程序保存到硬盘某个目录下，然后在程序的编辑过程中经常性地保存程序，以防止机器突然断电或者死机。要保存程序，只需从主菜单选择“文件”——“保存”就可以将文件保存到指定的硬盘目录。如图6所示。

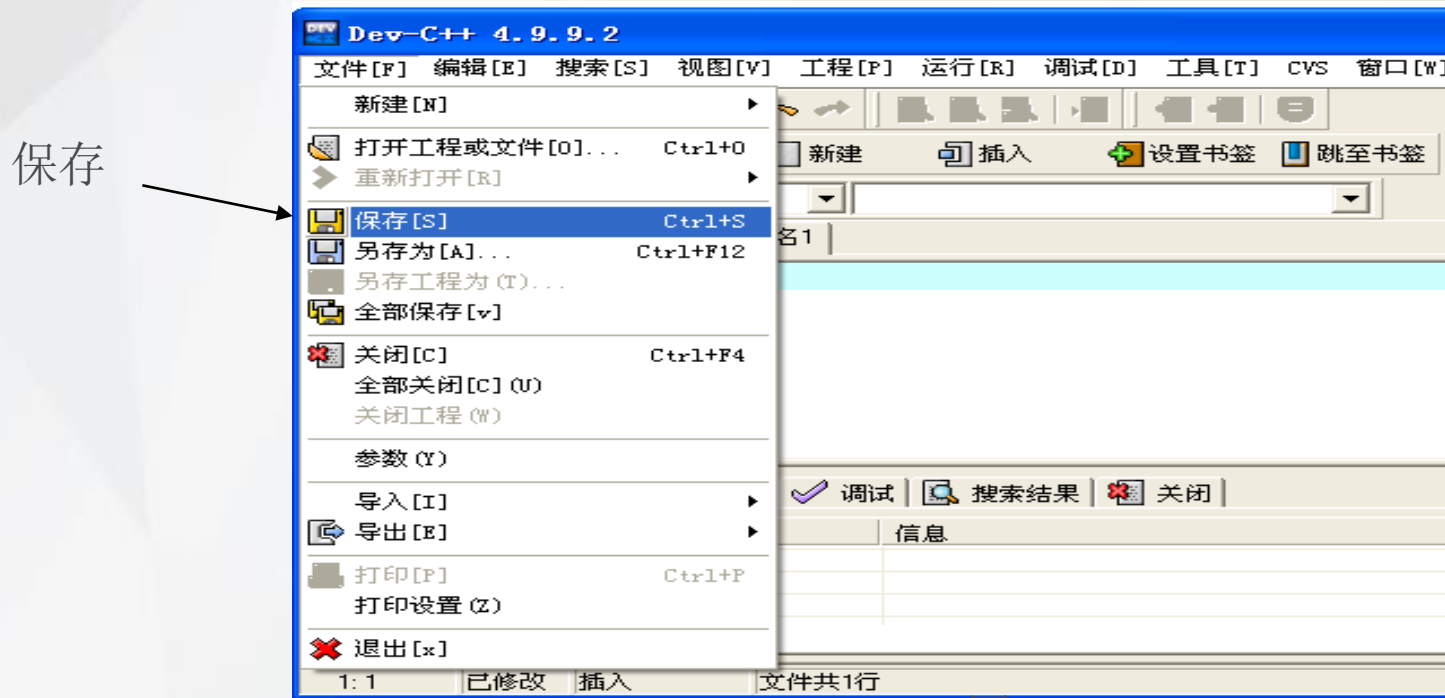


图6

此时会弹出一个对话框。在此你需要指定文件要存放的磁盘目录（例如：**D:\temp**），自定文件名称(例如：**ex1**)以及保存类型。在点击右下角的保存按钮后，在**D**盘上的**temp**目录下将会出现一个名为**ex1.cpp**的源文件。

四、编译、运行

编译：从主菜单选“运行”——“编译运行”或快捷键“F9”（图7）。如果程序中存在词法、语法等错误，则编译过程失败。

编译器会在屏幕右下角的“Compile Log”标签页中显示错误信息,如（图8）所示，并且将源程序相应的错误行标成红色底色。

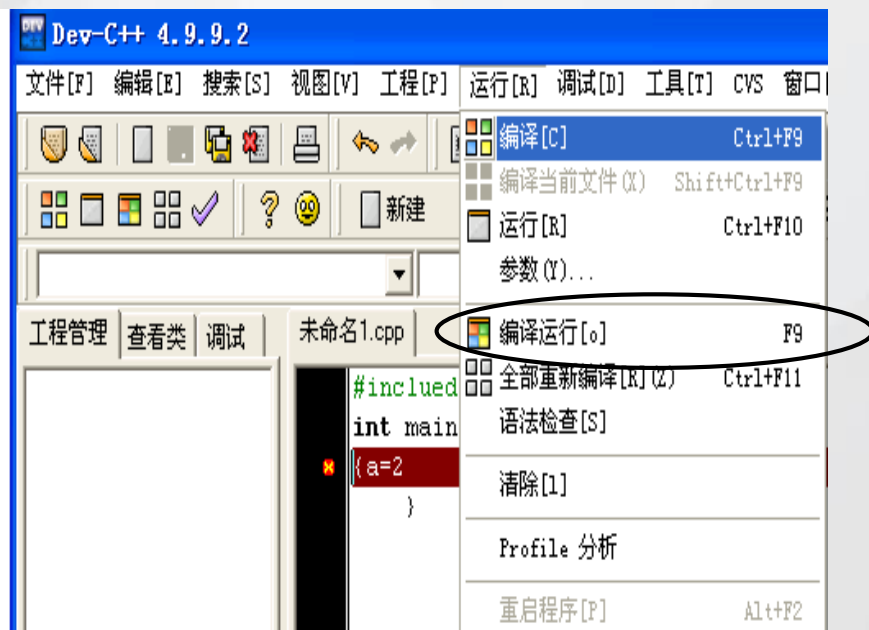


图7

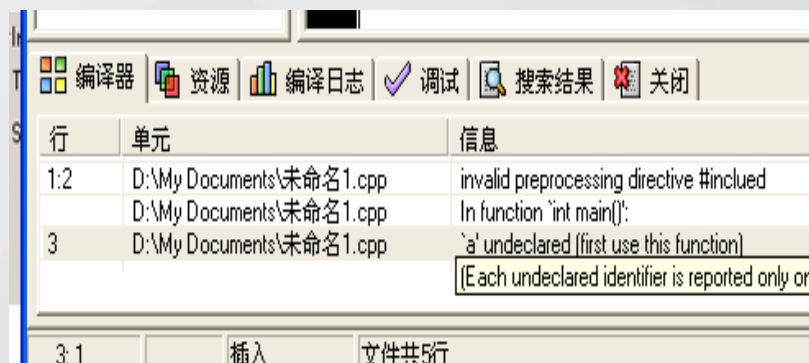


图8

五、调试程序

1. 设置程序断点

通过预处理、编译和链接的程序仅仅是该程序中没有词法和语法等错误，而无法发现程序深层次的问题（譬如算法不对导致结果不正确）。当程序运行出错时，需要找出错误原因。仔细读程序来寻找错误固然是一种方法，但是有时光靠读程序已经解决不了问题，此时需要借助于程序调试（Debug）手段。这是一种有效的排错手段，每一位同学都需要掌握。

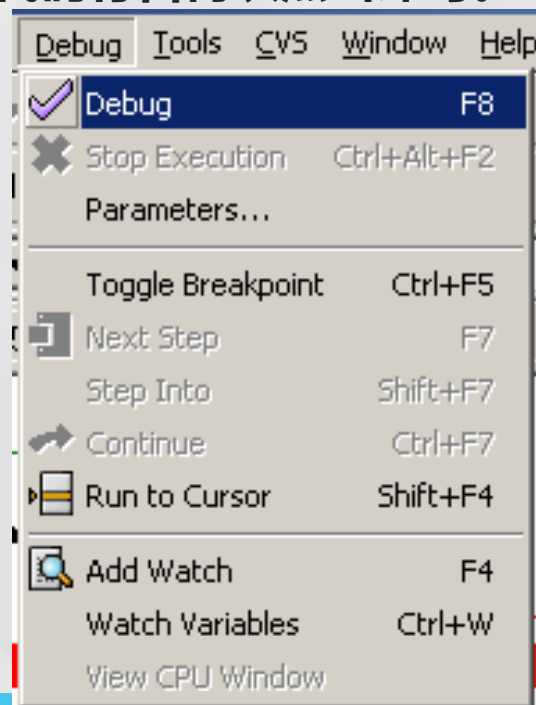
如果想让程序运行到某一行前能暂停下来，就需要将该行设成断点。具体方法是在代码所在行行首单击，该行将被加亮。默认的加亮颜色是红色。

如果想取消不让某行代码成为断点，则在代码行首再次点击即可。

2. 运行程序

设置断点后，此时程序运行进入debug状态。要想运行程序，就不能使用主菜单

“Execute” - > “Run”，而是需要用主菜单“Debug” —> “Debug”（或者按快捷键F8），如图12所示。



3. 单步执行程序

要想运行蓝色底色的代码，可以使用图14所示的“Next Step (F7)”、“Step Into (Shift F7)”、“Continue (Ctrl F7)”，“Run to Cursor (Shift F4)”等（具体含义见下面的介绍）。在学习函数之前，一般用的是Next Step和Continue。学习函数后，还会用到Step Into。

4. 设置watch窗口

在调试程序时，可能要看程序运行过程中变量的值，以检测程序对变量的处理是否正确，可以在调试时通过调试菜单下的添加变量（Add Watch）窗口来增加变量watch，新增的变量将会显示在最左边Explore的Debug页中。如果左边Explore中的当前页不是Debug页，则可以点击Debug标签使之成为当前页。

第二节 C++语言程序结构

无论做任何事情，都要有一定的方式方法与处理步骤，所谓“无规矩无以成方圆”。计算机程序设计比日常生活中的事务处理更具有严谨性、规范性和可行性。为了使计算机有效地解决实际问题，必须将处理步骤编排好，用计算机能理解的计算机语言编写成“序列”，让计算机自动识别并执行这个“序列”，达到解决实际问题的目的。将处理问题的步骤编排好，用计算机语言组成序列，就是常说的编写程序。在C++语言中，执行每条语句都是由计算机完成相应的具体操作，编写程序是利用C++语句的功能来实现预定的处理要求。“千里之行，始于足下”，我们从简单程序学起，逐步了解和掌握怎样编写程序。

在学习C++语言之前，让我们绕过那些繁琐的语法规则细节，通过一些简单的例题，来熟悉程序的基本组成和基本语句的用法，选手刚接触编程时，多动手模仿是一条捷径。

例1.2 在梯形中阴影部分面积是150平方厘米，求梯形面积。

【分析】 已知梯形上、下底长为15和25。令梯形的高为h，则由已知三角形面积为150平方厘米，有 $150 = (15 \cdot h) / 2$ ，得h为20，然后根据梯形面积公式算出梯形面积。程序如下：

```
#include<stdio>           //printf和scanf调用cstdio库,在C语言中可调用stdio.h库
#include<stdlib>           // 使用system( )调用stdlib库
using namespace std;       //在C语言中要省略，例如在VC++和TC++中
int main()                 //有的C语言可用void main(),例如TC++和VC++
{                           //整个程序开始
    float s,h,up,down;     // float定义s,h,up,down为单精度实型变量
    up=15;                 //已知上底
    down=25;               //已知下底
    h=2*150/up;            //根据上底求出梯形的高
    s=(up+down)*h/2;       //求出梯形的面积
    printf("s=%0.2f\n",s); // \n是换行控制符，0.2f按实际位数输出，保留2位小数
    system("pause");       //暂停作用，TC++和VC++中使用system( )调用stdlib.h库
    return 0;             //结束程序，在TC++和VC++中也要保留
}
```

运行结果：
s=400.00

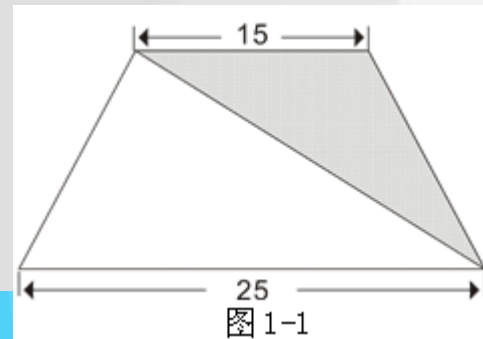


图 1-1

【说明】：

所谓namespace，是指标识符的各种可见范围。C++标准程序库中的所有标识符都被定义于一个名为std的namespace中。

当使用<iostream.h>时，相当于在C中调用库函数，使用的是全局命名空间，也就是早期的C++实现；当使用<iostream>的时候，该头文件没有定义全局命名空间，必须使用namespace std；这样才能正确使用cout、cin和endl。

例1.3 已知一位小朋友的电影票价是10元，计算x位小朋友的总票价是多少？

【分析】 假设总票价用y来表示，则这个问题可以用以下几个步骤来实现：

- ①输入小朋友的数目x；
- ②用公式 $y=10*x$ 计算总票价；
- ③输出总票价y的值。

程序如下：

```
#include<iostream>           //使用cin,cout, 须调用iostream库
using namespace std;
int main()
{
    int x,y;                   //定义整型变量
    cout<<"Input x=";         //输入提示
    cin>>x;                    //小朋友的人数
    y=10*x;                    //计算总票价
    cout<<"total="<<y<<endl; //输出总票价
    return 0;                  //结束程序
}
```

【说明】：

- 1、int定义x,y为整型变量，有关整型变量将在第二章详细介绍。
- 2、cout<<的作用是在屏幕中显示字符串"Input x="。
- 3、cin>>x的作用是输入一个数，赋予变量x。
- 4、cout<<"total="<<y<<endl的作用是输出提示符total=，输出y的值，endl是换行的意思。

通过以上例1.2和例1.3两个例子，可以总结出C++语言程序的结构如下：

- 1) C++语言变量在使用之前必须先定义其数据类型，未经定义的变量不能使用。
 - 2) 程序由一个或多个函数组成，一个程序中必须有且只有一个主函数，主函数的名字为main。不论main()函数在程序中什么位置，程序都是从main()函数开始执行，main()函数执行完毕，程序也就结束了。
 - 3) 在程序中可以调用系统提供的库函数。在调用库函数之前，必须将相应头文件包含在程序中。
 - 4) 程序中可以有注释行。注释是为了使程序更易于理解和提示。
 - 5) 程序的语句以分号结束。分号是C++语言不可缺少的组成部分。
 - 6) 程序的书写要注意适当的缩进，一般采用“逐层缩进”形式，以便使程序更加清晰易读。
- 把处理问题的步骤编成能从上到下顺序执行的程序，是简单程序的基本特征。再来分析下面一道例题的程序结构，同时继续学习基本语句。

例1.4 有一个牧场，牧场上的牧草每天都在匀速生长，这片牧场可供15头牛吃20天，或可供20头牛吃10天，那么，这片牧场每天新生的草量可供几头牛吃1天？

【分析】 解决这类问题的关键是利用牛吃的草量，最终求出这片牧场每天新生长的草量，我们设1单位的草量为1头牛1天所需的草量，于是15头牛20天所食的草量为300单位（包括这20天内的新生草量），20头牛10天所食的草量为200单位（包括这10天内的新生草量），两者的差值即为10天内的新生草量。程序如下：

```
#include<iostream>           //使用cin,cout, 须调用iostream库
using namespace std;
int main()
{
    int s1,s2,s3;              //变量定义
    s1=15*20;                  //15头牛20天所食的草量
    s2=20*10;                  //20头牛10天所食的草量
    s3=(s1-s2)/(20-10);        //每天新生的草量单位数
    cout<<"s="<<s3<<endl;    //1单位为1头牛1天的食量
    return 0;                  //结束程序
}
```

运行结果：

s=10

例1.5 给定一个字符，用它构造一个底边长5个字符，高3个字符的等腰字符三角形。

- #
- ###
- #####

○ 程序如下：

- `#include<iostream>` //使用cin,cout, 须调用iostream库
- `using namespace std;`
- `int main()`
- `{`
- `char a;` //定义字符变量,
- `cin>>a;` //输入给定一个字符
- `cout<<" "<<a<<endl;` //输出1个字符,先输出2个空格
- `cout<<" "<<a<<a<<a<<endl;` //输出3个字符,先输出1个空格
- `cout<<a<<a<<a<<a<<a<<endl;` //输出5个字符
- `return 0;` //结束程序
- `}`

【Dev C++版本注意事项】

在 5.0 版本 Dev C++ 中，<iostream> 不是万能库，不包括 cstdio, cstring, cstdlib，其他常用的 min, max 函数也不保证包括，需要注意如下几个细节：

- ①、getchar(), stdin, stdout, freopen(), fclose(), scanf() 在 c++ 中需要调用 <cstdio>
- ②、memset() 需要包含 C 头文件 <string.h>，在 c++ 中需要调用 <cstring>
- ③、qsort(), exit() 需要包含 C 头文件 <stdlib.h>，在 c++ 中需要调用 <cstdlib>

在 5.0 版本 C++ 中，根据标准去除了 iostream 对 cstdio, cstring, algorithm 的依赖性，需要显式包括。你现在还可能不透彻理解以上注意事项，就慢慢消化吧，在编程竞赛中一定要注意以上细节，请记住：“细节决定成败”。

【上机练习】

1.Hello,World! 【1.1编程基础之输入输出01】

编写一个能够输出“Hello,World!”的程序，这个程序常常作为一个初学者接触一门新的编程语言所写的第一个程序，也经常用来测试开发、编译环境是否能够正常工作。

输入：

无。

输出：

一行，仅包含一个字符串：“Hello, World!”

样例输入：

(无)

样例输出：

Hello, World!

【上机练习】

2. 输出第二个整数【1.1编程基础之输入输出02】

输入三个整数，整数之间由一个空格分隔，整数是32位有符号整数。
把第二个输入的整数输出。

输入：

只有一行，共三个整数，整数之间由一个空格分隔。整数是32位有符号整数。

输出：

只有一行，一个整数，即输入的第二个整数。

样例输入：

123 456 789

样例输出：

456

【上机练习】

3. 对齐输出【1.1编程基础之输入输出03】

读入三个整数，按每个整数占8个字符的宽度，右对齐输出它们，按照格式要求依次输出三个整数，之间以一个空格分开。

输入：

只有一行，包含三个整数，整数之间以一个空格分开。

输出：

只有一行，按照格式要求依次输出三个整数，之间以一个空格分开。

样例输入：

123456789 0 -1

样例输出：

123456789 0 -1

【上机练习】

4.字符三角形【1.1编程基础之输入输出08】

给定一个字符，用它构造一个底边长5个字符，高3个字符的等腰字符三角形。

输入：

输入只有一行， 包含一个字符。

输出：

该字符构成的等腰三角形，底边长5个字符，高3个字符。

样例输入：

*

样例输出：

*

【上机练习】

5. 地球人口承载力估计【小学奥数7653】

假设地球上的新生资源按恒定速度增长。照此测算，地球上现有资源加上新生资源可供 x 亿人生活 a 年，或供 y 亿人生活 b 年。

为了能够实现可持续发展，避免资源枯竭，地球最多能够养活多少亿人？

输入：

一行，包括四个正整数 x ， a ， y ， b ，两个整数之间用单个空格隔开。
 $x > y$ ， $a < b$ ， $ax < by$ ，各整数均不大于10000

输出：

一个实数 z ，表示地球最多养活 z 亿人，舍入到小数点后两位。

样例输入：

110 90 90 210

样例输出：

75.00