

概 论

Wang Houfeng

EECS, PKU

内容

➤ 本课程基本信息

- 程序设计语言及其发展
- C/C++语言程序及上机环节

课程相关信息

主讲教师:

- 王厚峰 (wanghf@pku.edu.cn)
- 计算机学院 计算语言学研究所
- 理科1号楼1445N室

课程助教:

- 徐子遥 (xzyxzy@stu.pku.edu.cn)
- 金 晶 (11jj617@stu.pku.edu.cn)
- 李 炜 (2201213158@pku.edu.cn)
- 罗 文 (llvvvv@stu.pku.edu.cn)
- 沈天舒 (2000012989@stu.pku.edu.cn)

课程网站: <http://course.pku.edu.cn>

上课&上机实习安排

上课时间

- 每周2次
 - 星期三 5-6节（下午1:00-3:00）
 - 星期五 3-4节（上午10:00-12:00）
- 地点：二教411教室

上机练习

- 时间：星期五 7-8节（下午3:00-5:00）
- 地点：理科 1 号楼 5-6号机房

成绩计算（暂定）

- 总评成绩占比
 - 期末占50%（上机考试）
 - 期中占15%（上机考试）
 - 平时作业占20%
 - 上机作业 15%
 - 计算机基础知识（概论）大课 5%（问卷星作答）
 - 大作业（上机编程实现）占10%
 - 一个综合性较强的题目（具体后面再说明），要有实验报告
 - 平时表现 5%
 - 上课表现
 - 上机表现

重要的时间节点

- 期中考试：11月11日（暂定）只有上机考试
- 期末考试：2024年1月3日（周三）只有上机考试
- 上述两次均为程序设计考试
- 计算机基础知识（概论）讲解：11月17日开始（预计）

课程辅导相关安排

- 课程网站与日常交流

<http://course.pku.edu.cn>

有关课程的讲义、通知等均在上述网址发布

利用网址进行交流、讨论，或向助教发送E-mail问问题。

- 程序设计上机作业

程序设计作业使用编程网格

(<http://programming.pku.edu.cn>)

期中+期末考试也使用该编程网格

网上程序设计练习

- 编程量要求
 - 不少于100小时
- 方式
 - 练习和作业均在网上完成

**练习编写程序是学好程序
设计的最重要途径！**

同学们应主动练习编程

本课程的目标

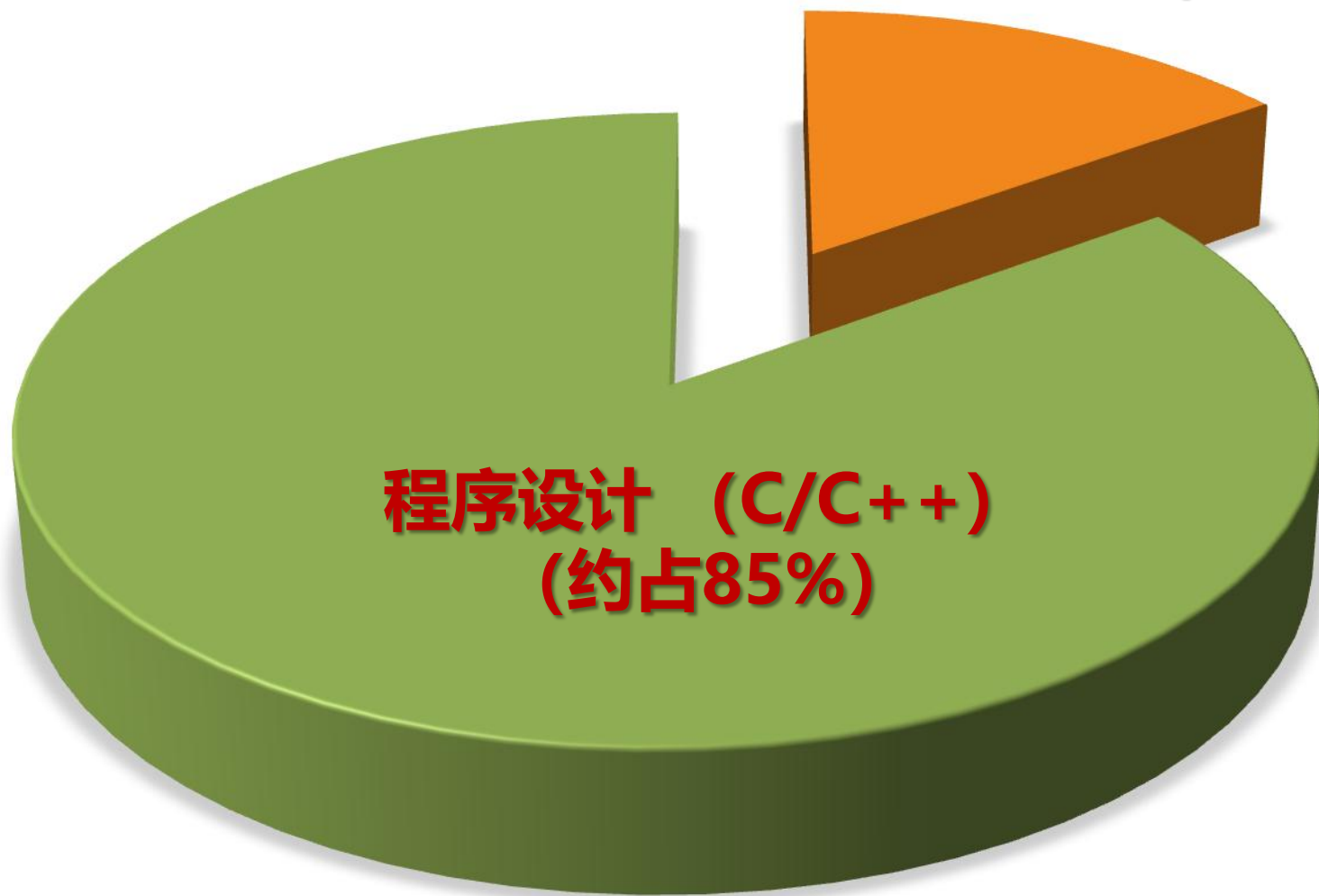
- **掌握 C/C++ 语言**

- 掌握语言规范（语法、语义），会熟练使用 C/C++ 语言设计程序
- 具备较强的程序设计技能，为今后的课程打下坚实基础

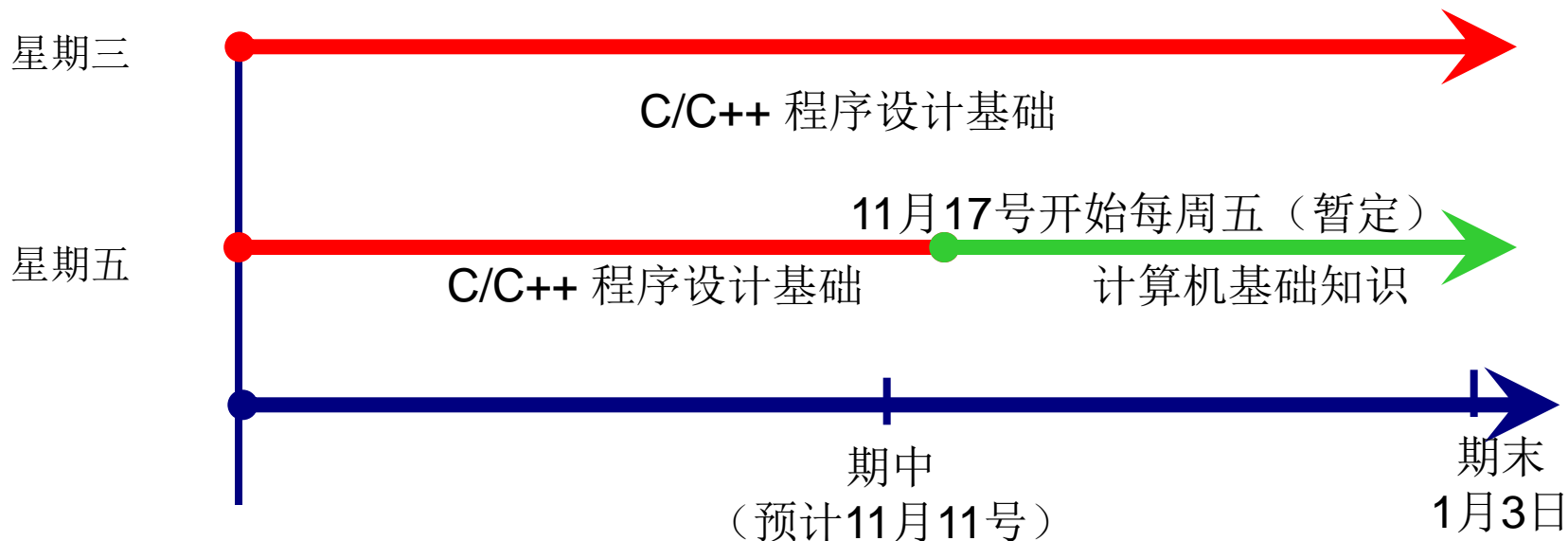
- **了解计算机的基础知识**

课程内容概览

计算机基础知识
(约占15%)



进度与内容



计算机基础知识

A1	计算机基本原理	A4	信息表示
A2	计算机的硬件原理	A5	计算机网络与存储
A3	计算机软件基础（主要是操作系统）	A6	程序语言基础

教材

- 没有指定的教材
- 参考教材
 - 程序设计基础，吴文虎主编，清华大学出版社
 - C++程序设计，谭浩强，清华大学出版社
 - 《程序设计导引及在线实践》李文新、郭炜、余华山，清华大学出版社，2007年

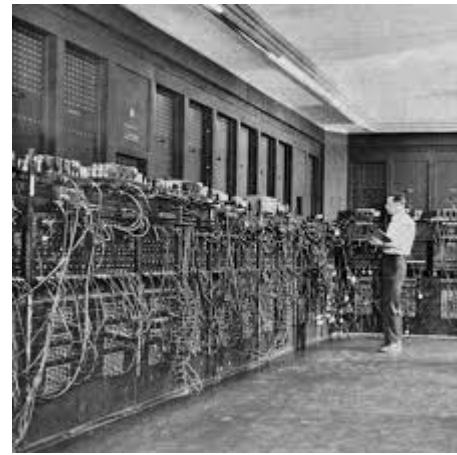
内容

- 本课程基本信息
- 程序设计语言及其发展
 - C/C++语言程序及上机环节

理解程序

- 理解程序
 - 语句（指令）序列
 - 类比文章：编写程序类比写作文，有如下特点
 - 不同：语言不同，通常的文章由自然语言的语句组成，程序则由程序语言的语句组成
 - 相同：都需要满足**语言规范**（既符合语法，也表达语义），都要达到预定的目标
- 程序的功能
 - 告诉计算机如何运行的动作，以完成特定的任务

程序设计的发展



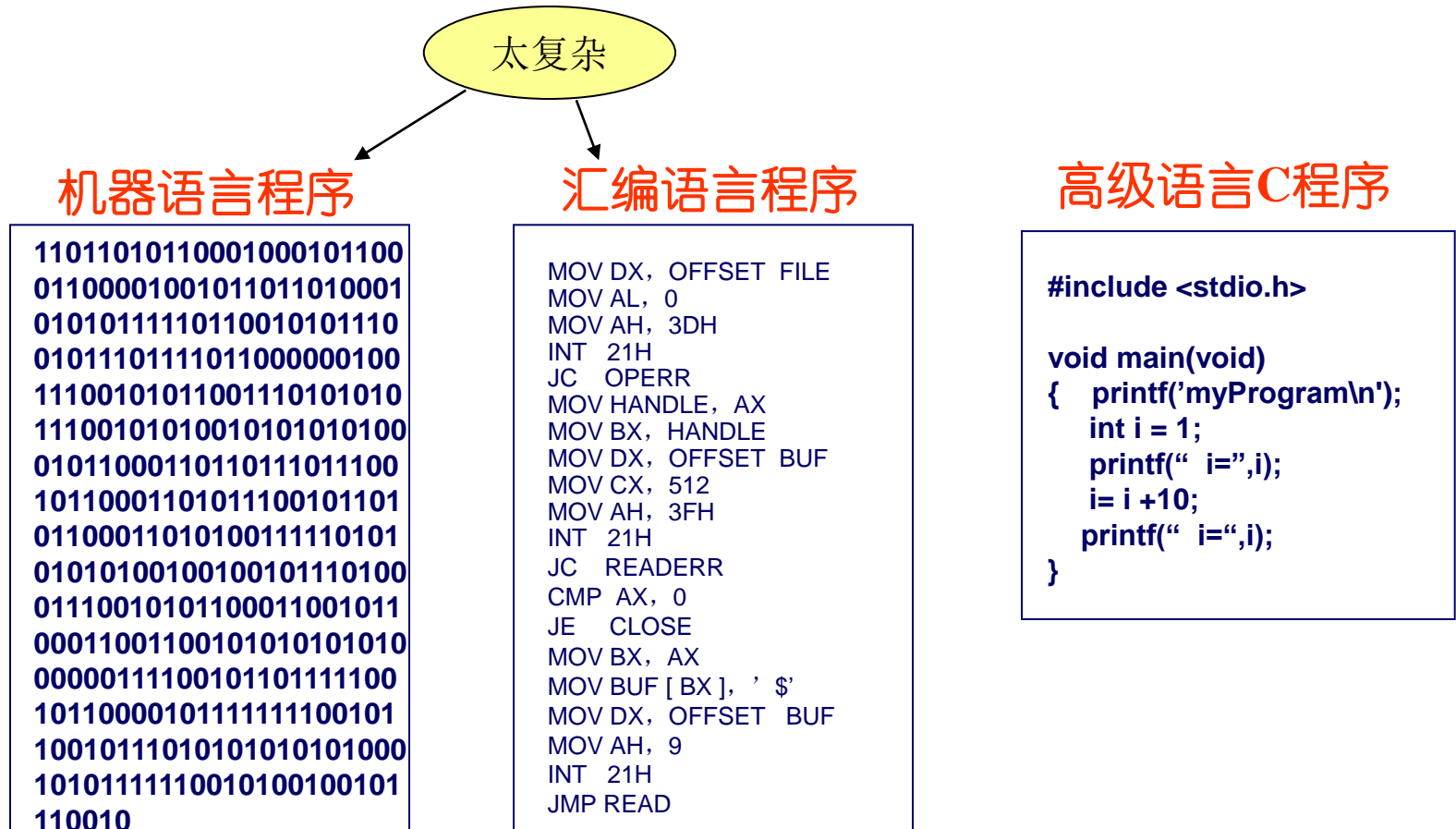
- ENIAC采用“手工插线”方式“编程”
- “编程” = 给计算机设置好运行的步骤
- 怎么告诉计算机
 - 早期插线（通电：1/不通电：0）
 - 现在写代码（编程）
 - 用特定语言描述计算逻辑（步骤）

↓
程序设计语言

↓
运行步骤



比较不同语言的程序



机器语言

- 特点

- 机器指令形成的程序
- 形式为二进制编码
- 机器可直接运行
- 优点：程序运行效率高
- 缺点：难读、难写、编程效率低

计算机只识别 **0/1** 两个符号

机器语言程序

```
11011010110001000101100
01100001001011011010001
01010111110110010101110
01011101111011000000100
11100101011001110101010
11100101010010101010100
01011000110110111011100
10110001101011100101101
01100011010100111110101
01010100100100101110100
01110010101100011001011
00011001100101010101010
00000111100101101111100
10110000101111111100101
10010111010101010101000
10101111110010100100101
110010
```

汇编语言

- 特点

- 助记符号形式
- 汇编指令与机器指令一一对应
- 写大程序仍然困难
- 汇编语言程序不能直接运行 —— 转换

汇编语言程序

```
MOV DX, OFFSET FILE
MOV AL, 0
MOV AH, 3DH
INT 21H
JC OPERR
MOV HANDLE, AX
MOV BX, HANDLE
MOV DX, OFFSET BUF
MOV CX, 512
MOV AH, 3FH
INT 21H
JC READERR
CMP AX, 0
JE CLOSE
MOV BX, AX
MOV BUF[BX], '$'
MOV DX, OFFSET BUF
MOV AH, 9
INT 21H
JMP READ
```

高级语言

高级语言C程序

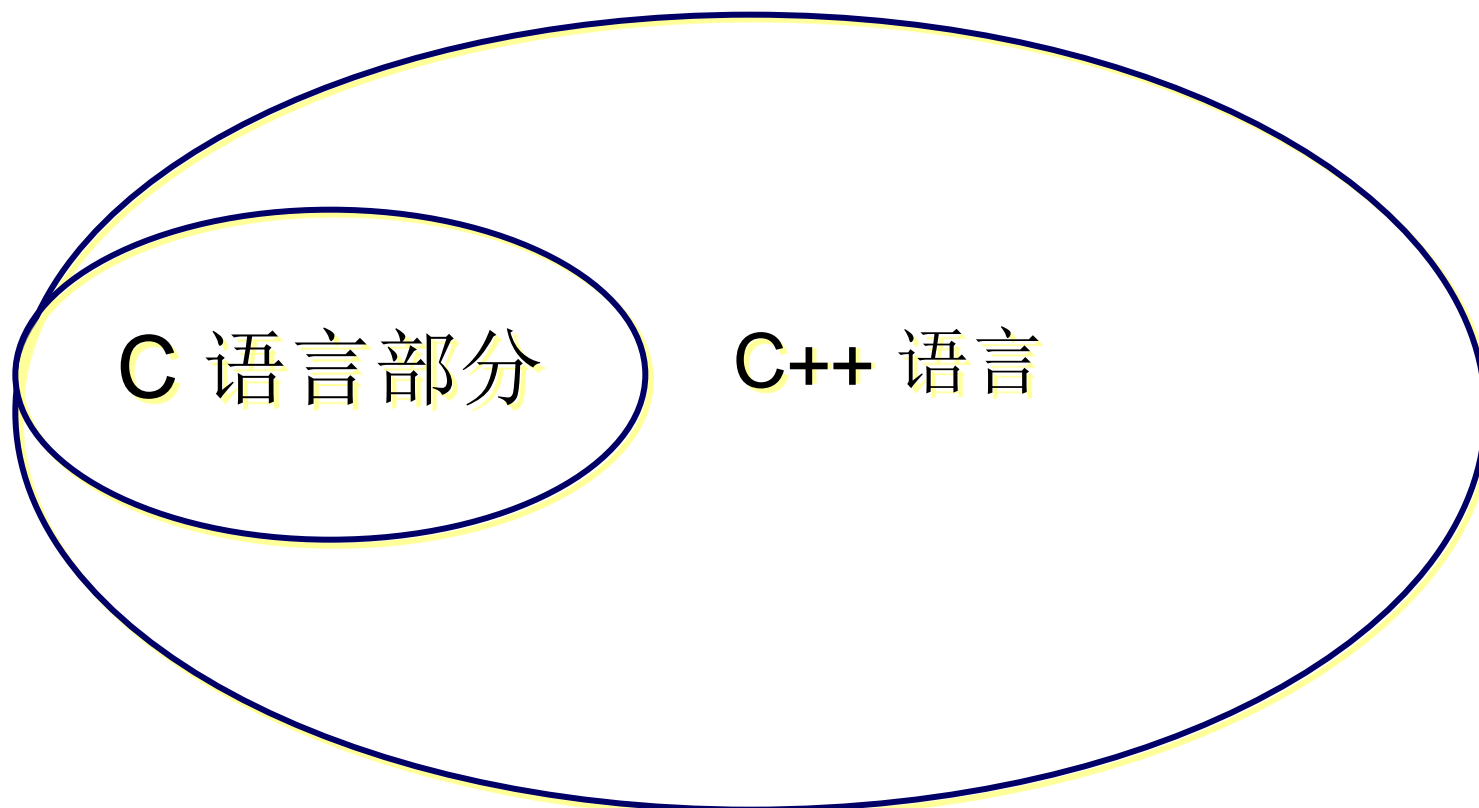
- 特点

- 用变量等概念取代低级的存储概念
- 人摆脱了繁琐的存储安排工作
- 运算类似数学式子一样的自然
- 具有类似文字的表现形式
- 提供了结构控制，表达更清晰可读
- 高级语言也需要转换为机器语言才能运行

```
#include <stdio.h>

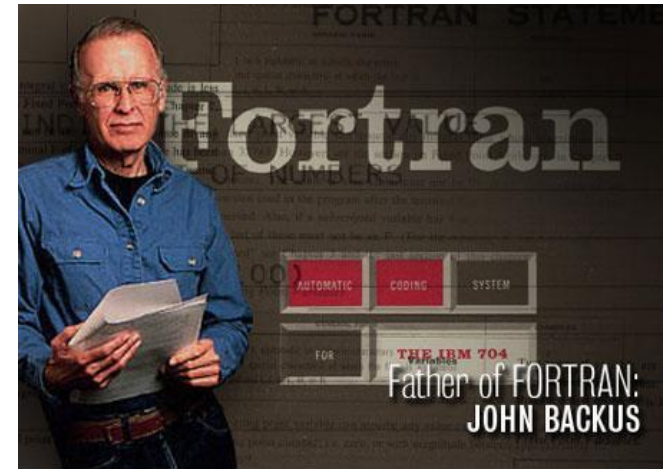
void main(void)
{   printf('myProgram\n');
    int i = 1;
    printf(" i=",i);
    i= i +10;
    printf(" i=",i);
}
```

本课程要讲的高级语言



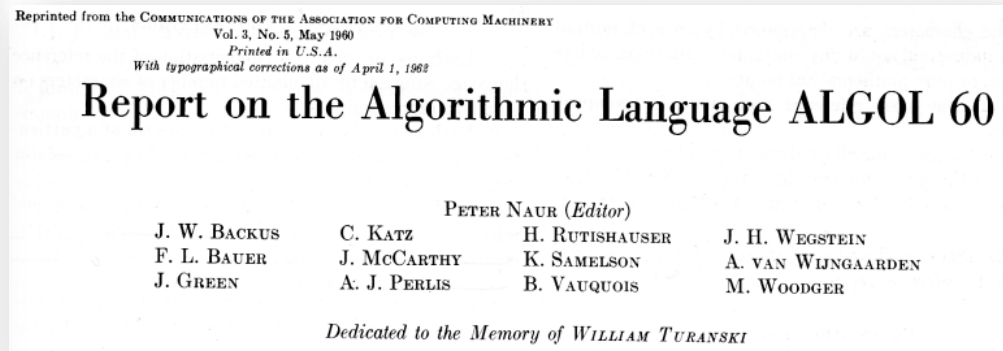
最早的高级程序设计语言

- John Backus (图灵奖获得者)
 - 高级语言之父
 - 设计了Fortran(FORmula TRANslation)
 - 大规模数值计算
 - 提出了 Backus 范式 (以巴科斯(Backus)和丹麦的诺尔(Naur)的名字命名的一种形式化的语法表示方法 BNF: Backus-Naur Form) 描述程序设计语言



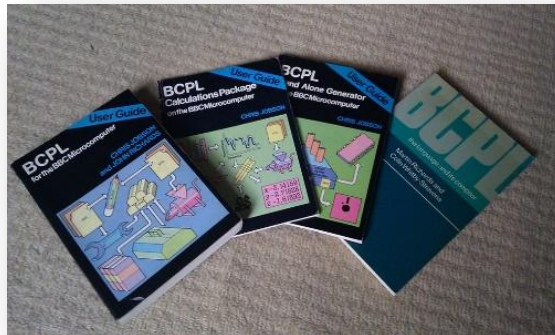
Algol60高级程序设计语言

- Algol60 : **ALGO**rithmic **L**anguage **60**
- 图灵奖获得者皮特.劳尔 (Peter Naur) 提出
 - 用巴科斯范式(BNF: Backus-Naur Form) 描述
 - 图灵奖得主Edsger Dijkstra把Algol 60描述为奇迹



CPL与BCPL高级程序设计语言

- CPL语言（ **C**ombined **P**rogramming **L**anguage ）
 - 1963年，英国剑桥大学在Algol60基础上提出
 - 更接近于硬件（如汇编语言）
 - 规模大且实现困难
- BCPL语言（ **B**asic **C**ombined **P**rogramming **L**anguage ）
 - 1967年，英国剑桥大学的 Martin Richards对CPL语言简化并得到了BCPL



B语言与C语言

- B语言
 - 贝尔实验室的Ken Thompson对BCPL进一步简化得到
 - Ken Thompson用B语言写第一个UNIX操作系统（命名为UNICS）
- C语言
 - 1972-1973年间，Dennis Ritchie和Ken Thompson在B语言的基础上发展和完善出C语言，并重写UNIX



图灵奖获得者

C语言特点



Thompson (left) and Ritchie (center)
receiving the National Medal of Technology
from President Clinton in 1999

- 具有高级语言的结构化和低级语言（汇编语言）的高效率特点（也称中级语言）
- 是一种系统程序设计语言
 - 开发操作系统
 - 编译系统等
- 源头可以追踪到 Algol 60

不同版本的C

- K&R C
 - 在1978年，Kernighan和Ritchie的《The C Programming Language》第一版出版，一直被广泛作为C语言事实上的规范，称K&R C
- ANSI C 和 ISO C
 - 1989年，C语言被ANSI标准化，对 K&R C进行了扩展，包括了一些新的特性，也规定了一套标准函数库
 - ISO成立WG14工作组来规定国际标准的C语言。通过对ANSI标准的少量修改，最终通过了ISO 9899:1990。随后ISO标准被ANSI采纳
- C99
 - 在ANSI标准化后，WG14小组继续致力于改进C语言。新的标准很快推出，就是ISO9899:1999（1999年出版）。这个版本就是通常提及的C99。它被ANSI于2000年三月采用。
- C11
 - 2011年12月8日，ISO正式公布C语言新的国际标准草案：ISO/IEC 9899:2011，即C11

C++的出现










- 1979年，贝尔实验室的Bjarne Stroustrup开发了一种语言，被称为“**C with Classes**”，后来演化为C++
- 1985年10月，Bjarne博士完成了经典巨著The C++ Programming Language 第一版
- 1998年11月ISO颁布了C++程序设计语言的国际标准ISO/IEC 14882-1998
- ISO于2011年9月1日发布了ISO/IEC 14882:2011，即人们常说的C++2011



Bjarne Stroustrup
本贾尼·斯特劳斯特卢普



程序设计语言排行（2023年8月）

Aug 2023	Aug 2022	Change	Programming Language		Ratings	Change
1	1			Python	13.33%	-2.30%
2	2			C	11.41%	-3.35%
3	4	^		C++	10.63%	+0.49%
4	3	v		Java	10.33%	-2.14%
5	5			C#	7.04%	+1.64%
6	8	^		JavaScript	3.29%	+0.89%
7	6	v		Visual Basic	2.63%	-2.26%
8	9	^		SQL	1.53%	-0.14%
9	7	v		Assembly language	1.34%	-1.41%

内容

- 本课程基本信息
- 程序设计语言及其发展
- C/C++语言程序及上机环节

C/C++ 语言程序结构

- 程序构成：若干平行的模块（函数），有且仅有一个 `main` 函数，为程序执行的起点和终点。
- 函数的构成：函数首部+函数体。
- 函数体的构成：非可执行语句+可执行语句。
- 非可执行语句：主要包含变量说明语句（如，整型，实型变量说明，也称变量声明）。
- 可执行语句，分号结束；

一个简单的C语言程序

关键字
或
保留字

变量
标识符

```
/* example calculate the sum of a and b */  
#include <stdio.h>
```

```
void main() ← 函数首部  
{ ← 非可执行语句  
  int a,b,sum; ← 非可执行语句  
  a=10; ← 可执行语句  
  b=24; ← 可执行语句  
  sum=add(a,b); ← 可执行语句  
  printf("sum= %d\n",sum); ← 可执行语句  
}
```



```
int add(int x,int y) ← 函数首部  
{ ← 非可执行语句  
  int z; ← 非可执行语句  
  z=x+y; ← 可执行语句  
  return(z); ← 可执行语句  
}
```

模块平行 没有包含关系

一个简单的C++语言程序

```
#include<iostream>
using namespace std;
int main( )
{
    int number[45] = {78, 56, 69, 31, 36, 67, 31, 47, 69, 34, 45, 74, 61,
        82, 43, 41, 76, 79, 81, 66, 54, 50, 76, 51, 53, 28, 74, 39, 45,
        61, 52, 41, 43, 75, 78, 84, 72, 51, 43, 64, 75, 81, 69, 55, 74};
    int max = 0;    //赋初值0
    int i = 0;
    for(i = 0; i < 45; i++)
    {
        if(number[i] > max)
            max = number[i];
    }
    cout<<"The Maximal Number is:"<<max;
    return 0;
}
```

本课程中
C++与C的区别

设计程序的几个注意点

```
#include<iostream>
using namespace std;
int main( )
{
    int number[45] = {78, 56, 69, 31, 36, 67, 31, 47, 69, 34, 45, 74, 61,
        82, 43, 41, 76, 79, 81, 66, 54, 50, 76, 51, 53, 28, 74, 39, 45,
        61, 52, 41, 43, 75, 78, 84, 72, 51, 43, 64, 75, 81, 69, 55, 74};
    int max = 0;    //赋初值0
    int i = 0;
    for(i = 0; i < 45; i++)
    {
        if(number[i] > max)
            max = number[i];
    }
    cout<<"The Maximal Number is:"<<max;
    return 0;
}
```

看程序，发现要点

- 1) 程序基本框架
- 2) 数据及表示
(包括数据类型)
- 3) 数据加工
- 4) 控制结构

如何设计程序

- 符合基本规范或框架
 - 如, #include, main 的使用
 - 正确使用语言的表示 (如, for、if的使用规范)
- 需要哪些数据
 - 数据的类型是什么 (整型, 实型 等)
 - 哪些数据 (包括变量、常量)
- 对数据需要什么基本的操作
 - 数据的运算及其规范
- 如何对数据的运算进行控制
 - 控制语句的使用 (for, if 等)

再看简单的C语言程序

关键字
或
保留字

变量
标识符

```
/* example calculate the sum of a and b */  
#include <stdio.h>
```

```
void main() ← 函数首部  
{ ← 非可执行语句  
  int a,b,sum; ← 非可执行语句  
  a=10;  
  b=24;  
  sum=add(a,b); } 可执行语句  
  printf("sum= %d\n",sum); }  
}
```



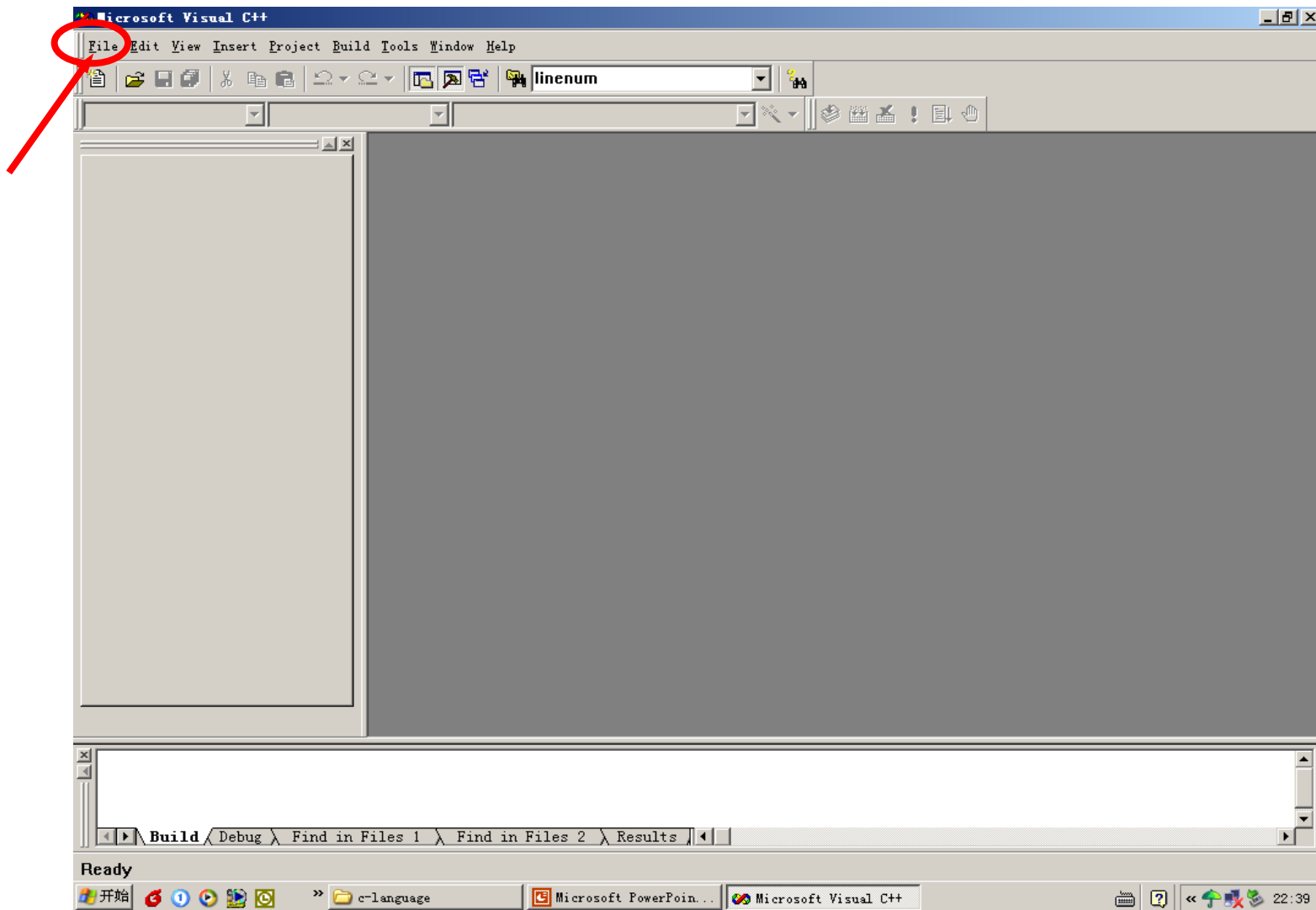
```
int add(int x,int y)  
{ int z;  
  z=x+y;  
  return(z);  
}
```

模块平行 没有包含关系

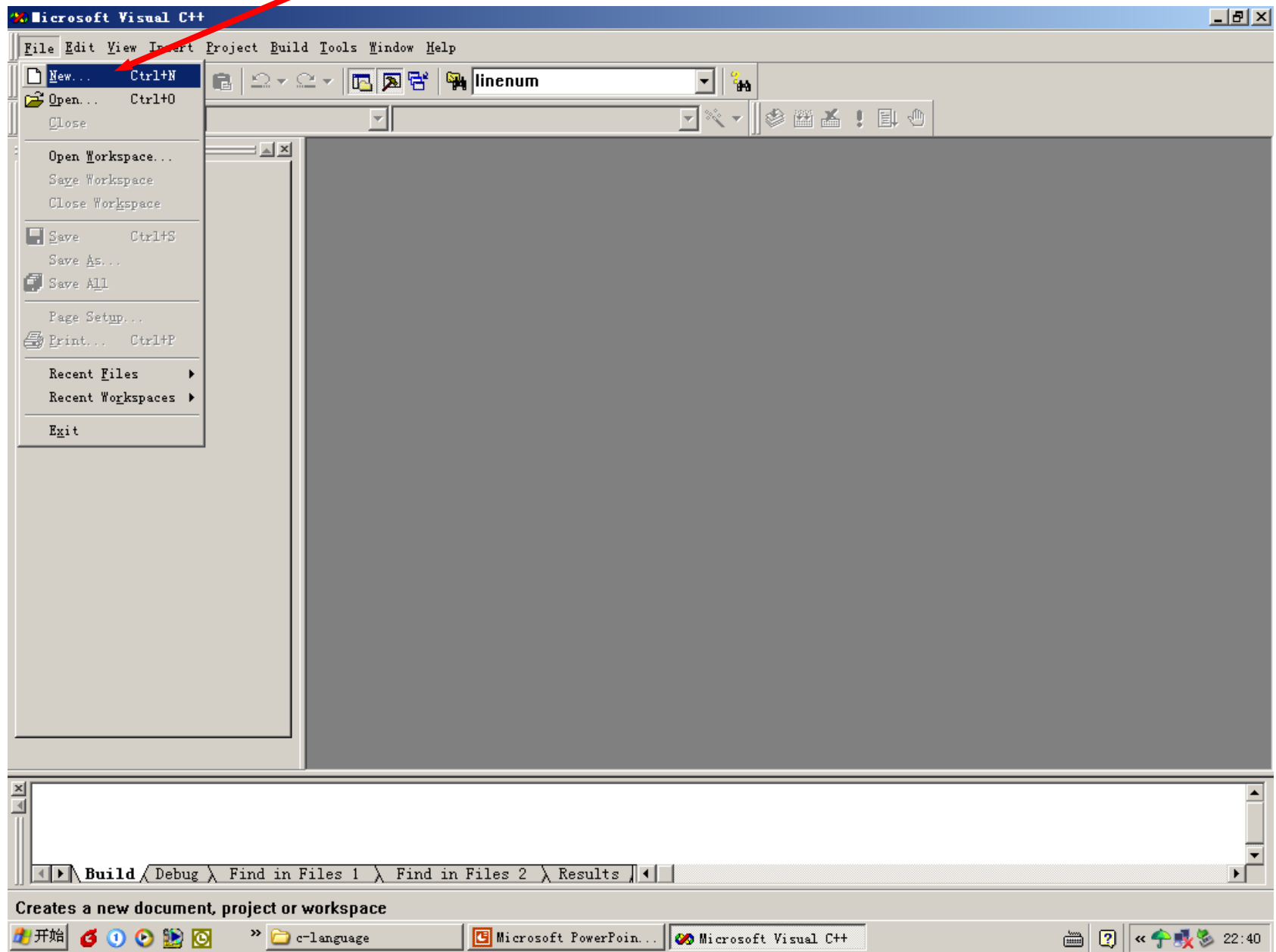
VC++编程环境 - 基本概念

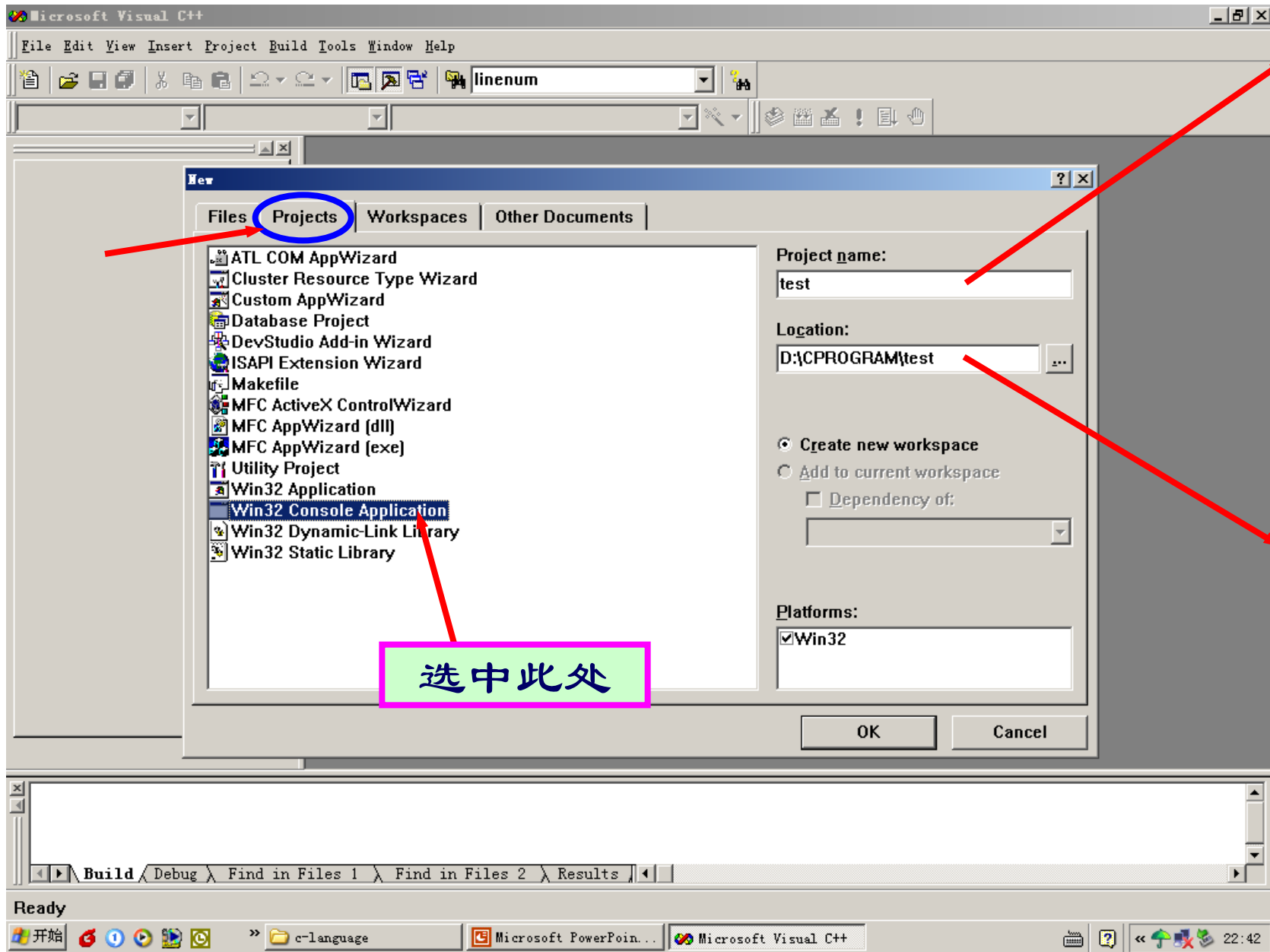
- Project（工程）：VC编程环境下，编写程序的工作是以Project为单位
- 开始一个新程序时，需先建一个Project，之后编写程序有关的文件都会包含在这个Project中
- 程序有多种不同类型，编程环境为每种类型准备了一个模版，用来生成程序的最初框架。本课程只介绍编写Win32 Console Application（控制台应用程序）类型的程序
- 该类程序的特点：程序运行中会打开一个类似于DOS操作系统的界面，所有键盘输入都是通过DOS界面进行的，而所有输出都是输出到DOS窗口中。

Visual Studio 进入、选择File



选择 New

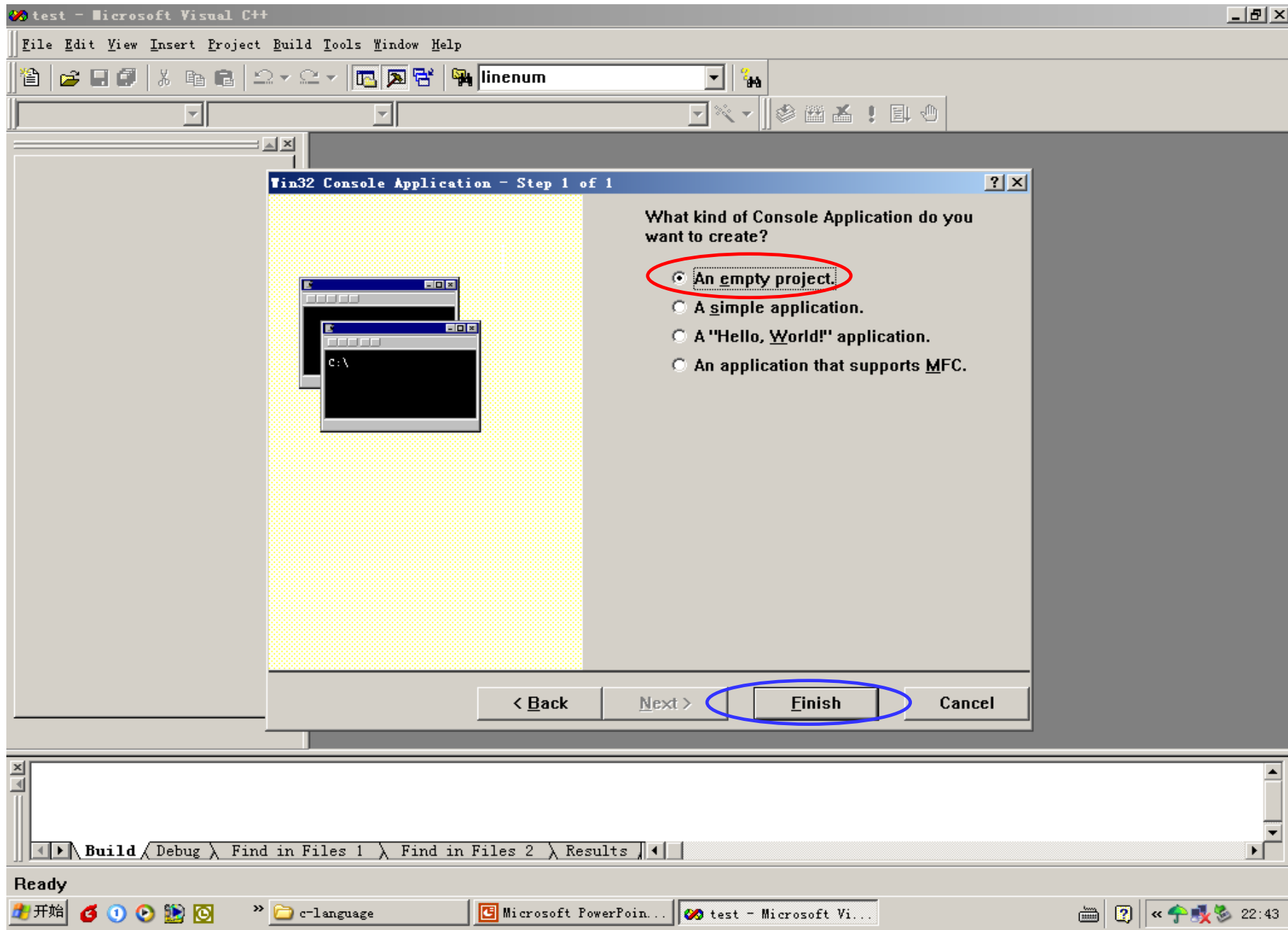


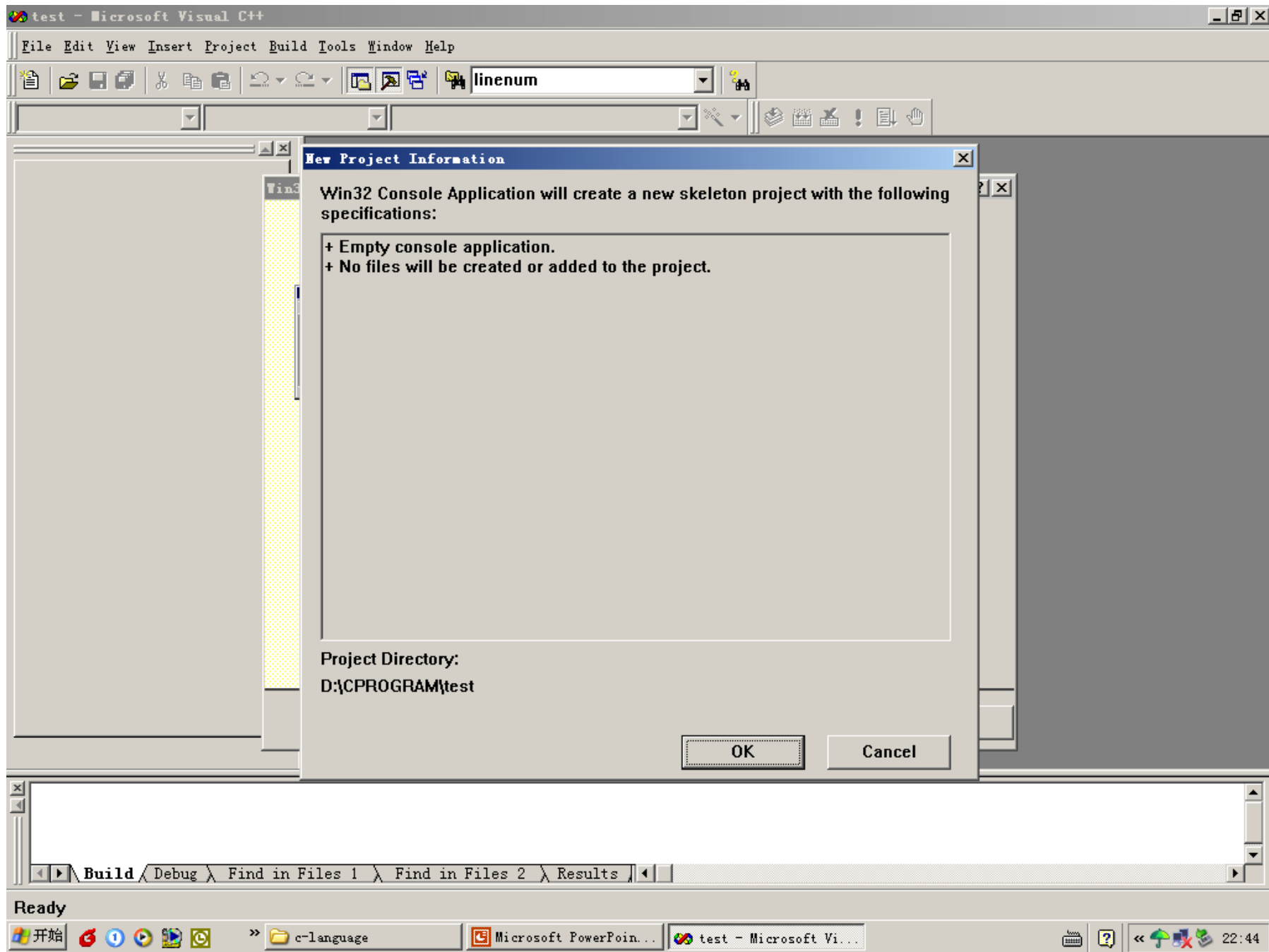


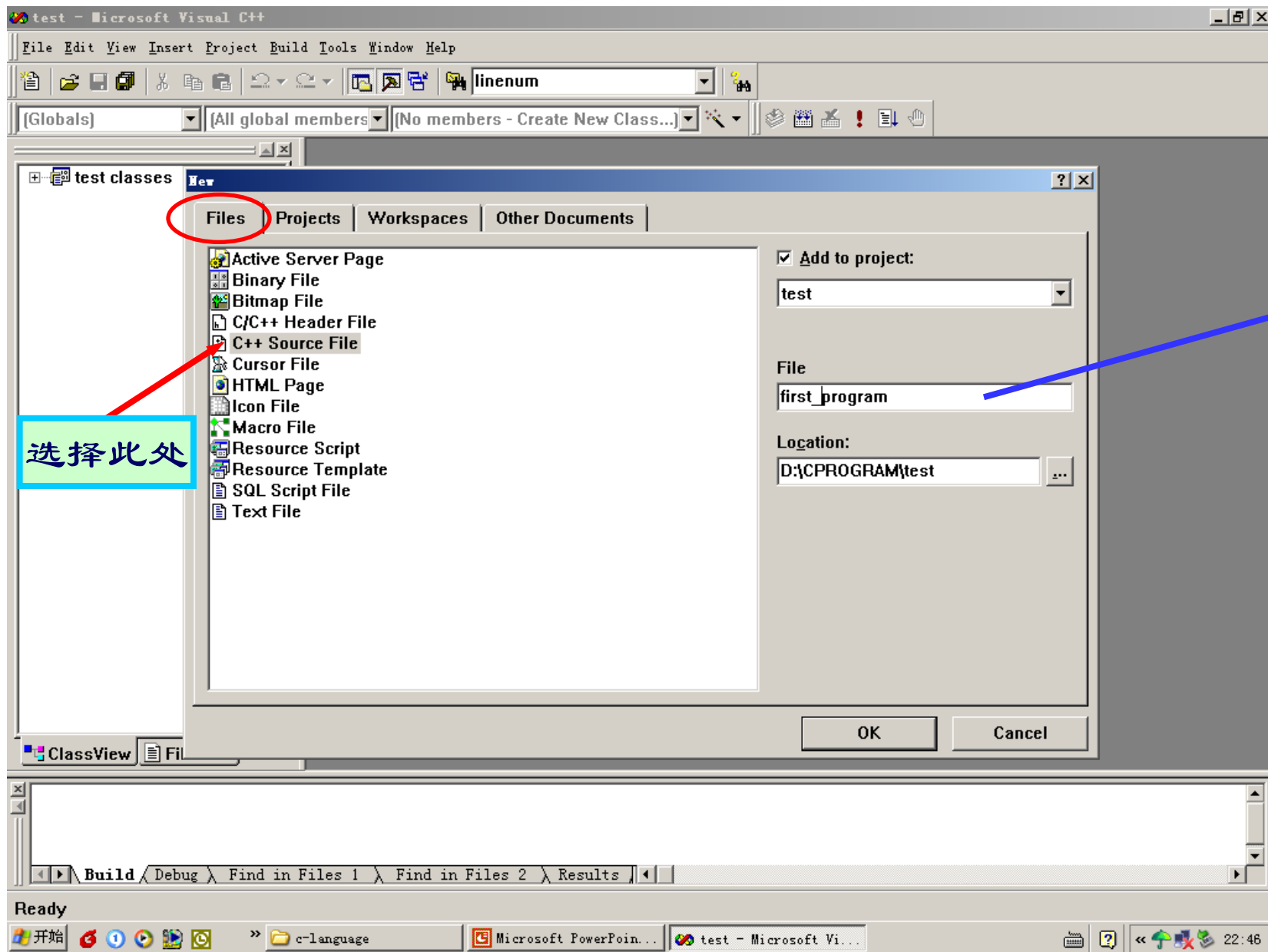
程序
所在
工程
的
名字

放在
什么
位置
(目录)

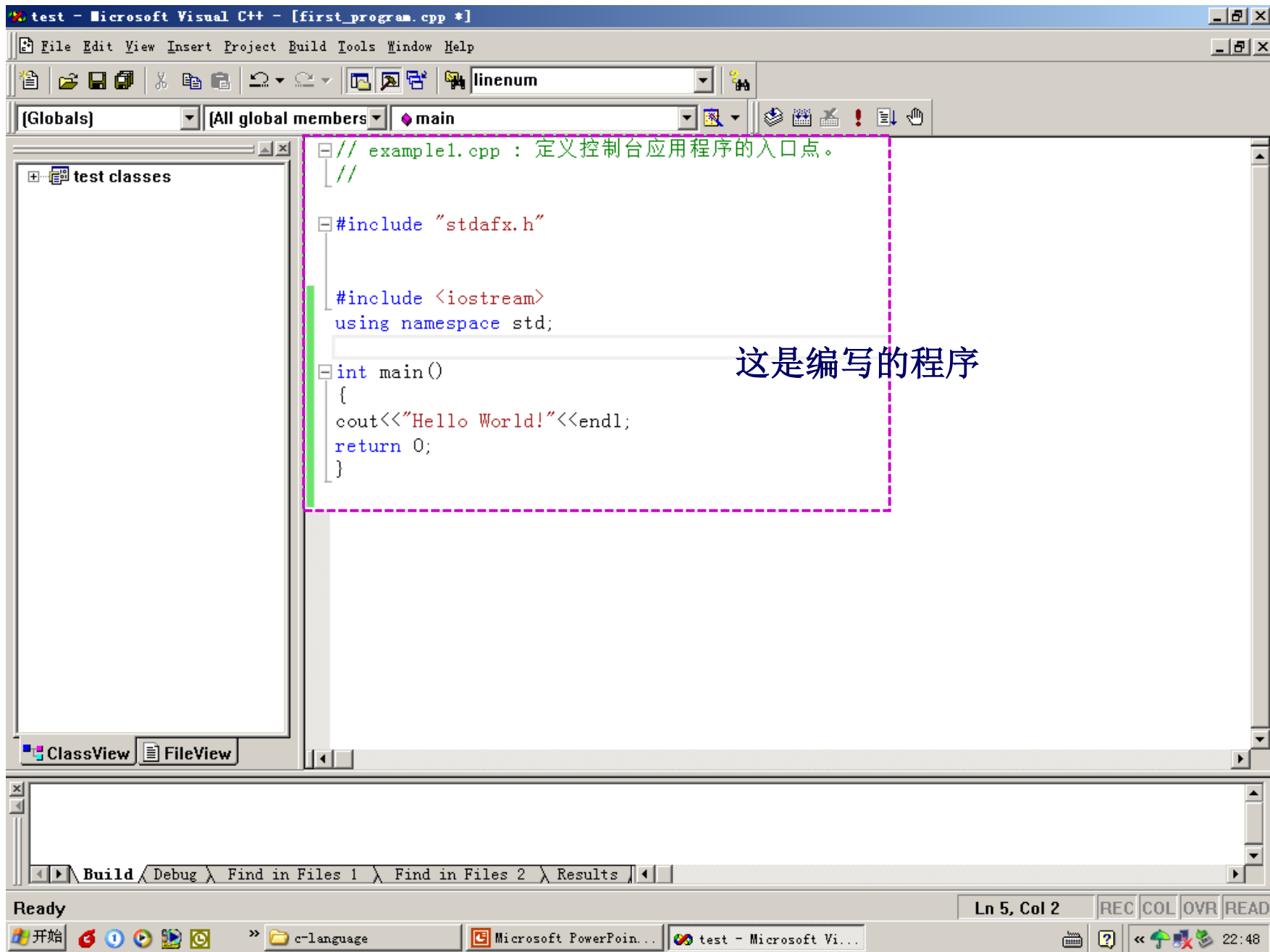
选中此处

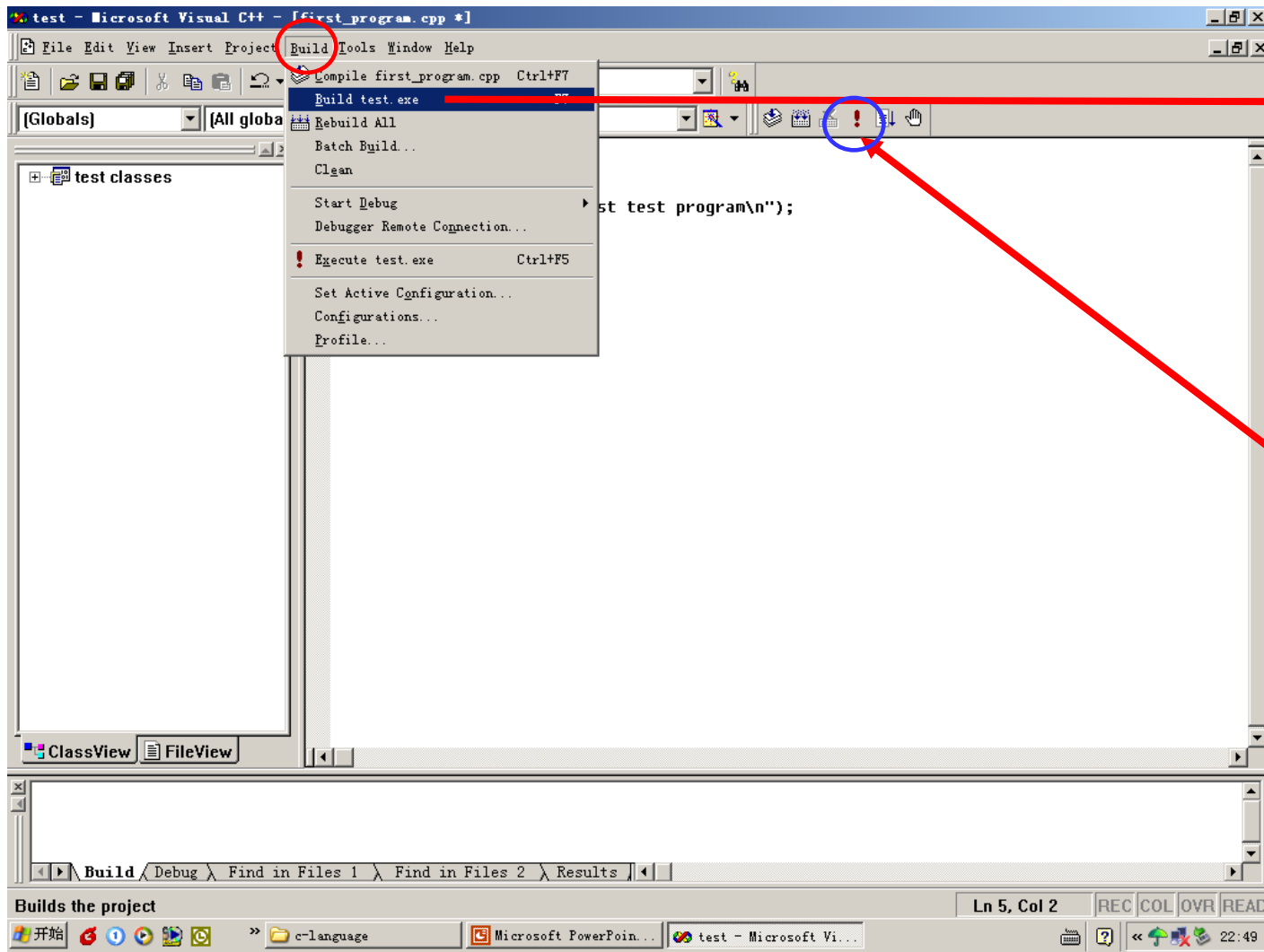






程序名字



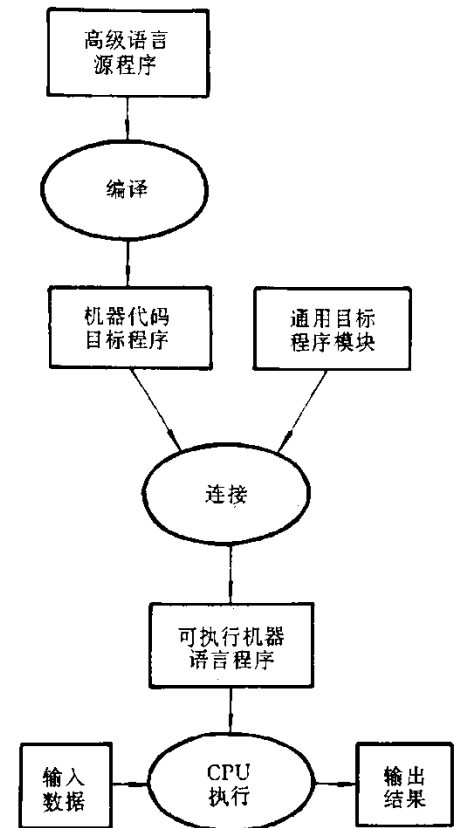


集成编译
与连接

运行程序

Source 源文件

- 用高级程序设计语言书写的程序称为**源程序**，本课程主要介绍C/C++语言编写源程序
- 源程序不能执行，需要变成机器目标代码，此过程称为**编译**；不同高级语言有不同的编译器
- 编写的程序里会调用一些功能模块，需要一个**连接**的过程整合起来形成最后的可执行程序



VC++上机步骤

- 启动VC
- 创建新工程
- 编写源代码
- 编译链接
- 运行
- 注意：
 - 在VC环境下，可能看不到结果直接退出
 - 解决办法：在主函数尾部加上语句：
 - `system("pause");`
 - 在编程网格下必须删除上面语句

编程网格

编程作业、练习以及期终考试都将在这里进行。

编程网格：

<https://programming.pku.edu.cn/>

计算概论A2023-王厚峰

编程网格主页



编程网格
PROGRAMMING GRID

2021-07-11 · [首页](#) · [课程](#) · [题目](#) · [管理](#) · [帮助](#)

用户名 口令 登录 注册

任意 ▾ 查找课程

任意 ▾ 查找题目

课程

[更多 >>](#)

- [《计算概论--程序设计阅读题解》开放课程\(北大\)](#)
- [计算概论-FP\(北京大学\)](#)
- [计算概论A-2020-期末模拟0](#)
- [计算概论B2020习题集\(张焱\)\(北京大学\)](#)
- [计算概论 B 2020 习题集 \(甘锐\)\(北京大学\)](#)
- [计算概论B2020习题集孙俊\(北京大学\)](#)
- [计算机概论B-2020陈泓捷\(北京大学\)](#)
- [《计算概论B》上机考试参考\(北京大学\)](#)
- [计算概论B-医学部-2020-张伟\(北京大学\)](#)
- [计算概论B-医学部-2020 \(罗英伟\)\(北京大学\)](#)
- [计算概论B-2020-医学部-3班-高良才\(北京大学\)](#)

题目

[更多 >>](#)

- [奥特曼打小怪兽](#)
- [田忌赛马](#)
- [电池的寿命](#)
- [多维数组的](#)
- [二分法求方](#)
- [查看实型数](#)
- [简易魔塔](#)
- [字符串排序](#)
- [子字符串计数](#)
- [3X+1猜想](#)
- [有趣的二进制](#)

1
先注册一个用户
再登录到编程网格

选课

用学号登录进入，在主页中选择相应课程，进入课程页面后，选择“选课页面”

选课页面

课程

计算概论A 2023 王厚峰

课程注册码

A2023-3

学号

*

真实姓名

*

邮件地址

提交

请务必填写正确学号和真实姓名
作为平时作业成绩和考试记录用！

开始做题

- 先在首页登录，然后进入课程
- 选择练习、作业或考试，进入题目列表
- 点击题目，开始做题

选择练习/习题

练习 - 熟悉程序设计环境 - (已结束)

编程语言: C++(G++) 教师页面 提交记录 提交排名

开放时间: 2021-09-17 14:30 ~ 2021-10-13 21:00

1. 熟悉编程网格

题目

2. 例题(2.1): Hello World

题目

3. 1000 A+B Problem

题目

4. 例题(3.12) 计算圆周长和球体积

题目

5. 例题(3.3) 求整数相除取余数

题目

6. 习题(3-3) 计算 $(a+b)*c$ 的值

题目

7. 例题(3.8) 计算分数的浮点数值

题目

✓ 8. 计算并联电阻的阻抗

题目

选择具体题目开始练习

题目 - 熟悉编程网格

来源: 2013秋 计算概论A 刘?哲 教师页面

[下一题]

[提交程序] [IDE] [提交记录]

题目描述

将如下程序在本地IDE中编译运行成功。然后，将已经编译运行成功的代码在编程网格提交并测试通过（结果显示“Passed”即为通过）。

```
#include "iostream"
#include "cmath"
using namespace std;

int main()
{
    float x, y;
    cin >> x;
    y = abs(x);
    cout << y << endl;

    return 0;
}
```

关于输入

输入仅包含一行，为一个浮点数 x ($0 \leq x \leq 10000$)

关于输出

输出仅包含一行，为浮点数 x 的绝对值

例子输入

提交程序

题目 - 熟悉编程网格

来源: 2013秋 计算概论A 刘?哲 教师页面 [下一题]

☒ 自动判断 ☐ C ☐ C++ ☐ Java ☐ Python

将程序放在该框内

可以直接编写，也可以在其他
环境下调试好后粘贴在此处

[返回](#) [提交](#)

[首页](#) | [联系我们](#) | [意见建议](#) | |

北京大学网络与信息系统研究所 © 编程网格 © Programming Grid

粘贴方式

- 先在其他环境下调试成功，例如，在Visual C++中编写程序，并调试正确
- 将调试正确的程序从VC中Copy到编程网格的“提交程序”空白处
- 点击“提交程序”，就可查看结果了

提交程序

题目 - 熟悉编程网格

来源: 2013秋 计算概论A 刘?哲 教师页面 [下一题]

☒ 自动判断 ☐ C ☐ C++ ☐ Java ☐ Python

将其他环境下调试好后粘贴在此处

返回

提交

首页 | 联系我们 | 意见建议 | |

北京大学网络与信息系统研究所 © 编程网格 © Programming Grid

也可以选择集成环境编写程序

题目 - 熟悉编程网格

来源: 2013秋 计算概论A 刘?哲 教师页面

[下一题]

[提交程序] [IDE] [提交记录]

题目描述

将如下程序在本地IDE中编译运行成功。然后，将已经编译运行成功的代码在编程网格提交并测试通过（结果显示“Passed”即为通过）。

```
#include "iostream"
#include "cmath"
using namespace std;

int main()
{
    float x, y;
    cin >> x;
    y = abs(x);
    cout << y << endl;

    return 0;
}
```

关于输入

输入仅包含一行，为一个浮点数 x ($0 \leq x \leq 10000$)

关于输出

输出仅包含一行，为浮点数 x 的绝对值

例子输入

程序正确通过时的界面



2007-10-21 [首页](#) • [课程](#) • [题目](#) • [管理](#) • [帮助](#) 汪小林 [wx1] [注销](#)

首页 >> 课程: 计算概论07... >> 练习: 测试1 >> 题目: Hello ... - [提交](#) [历史](#) [返回](#)

Passed




程序源码: [C++](#)

提示

Case 0: Time = 3ms, Memory = 180kB.

[所属题集](#) 1 2 [下一题](#)

程序提交的可能结果

- Passed  程序正确
- Wrong Answer  程序不正确（逻辑错误）
- Compile Error  程序不正确（语法错误）
- 其他错误

注意事项

- `main()` 函数的返回值必须是 `int/void` 类型

```
int main()
{
    ...
    return 0;
}
```

- 往编程网格中copy程序时，**不要将VC创建的程序项目的第1行 `#include "stdafx.h"` 拷过去**
 - AppWizard和VisualC++编译程序一起工作，自动建立了文件 `stdafx.h`

注意事项

```
//Hello World
#include <iostream>
using namespace std;
```

```
int main()
{
    cout<<"Hello World";
    return 0;
}
```

```
//a+b App
#include "stdafx.h"
```

```
#include <iostream>
using namespace std;
int main()
{
    int a, b, c;
    cin>>a>>b;
    c = a + b;
    cout<<c;
    return 0;
}
```