



《计算概论A》课程 程序设计部分

感性认识C++程序

李 戈

北京大学 信息科学技术学院 软件研究所

2010年9月15日



北京大学



什么是程序？

■ 计算机也是机器！

- ◆ 必须“设置”好才能运行；
 - ENIAC采用“手工插线”的方式“编程”；
- ◆ “编程序” = 给计算机设置好运行的步骤；

■ 程序

- ◆ 人们用来告诉计算机应该做什么的东西；

■ 怎么告诉它呢？（程序到底是什么样子？）

- ◆ 告诉计算机一些什么东西，它才能运行？
- ◆ 以什么形式告诉它，它能够明白？



北京大学



问题一

■ 告诉计算机一些什么东西，它才能运行？

◆ 给你一个数列：

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



北京大学



你怎么做的？

■ 哪个数字最大？

- ◆ 把某一个数字取出来，当作一个临时的“特别数字”记住，并假设这个数字最大；
- ◆ 拿这个临时的“特别数字”与其他数字相比较；
- ◆ 如果有其他数字比临时的“特别数字”更大，就把“特别的数字”换成这个更大的数字；
- ◆ 重复上述过程直到把所有的数字都比较完毕；
- ◆ 那么大脑中这个“特别数字”就记录了最大的数字；



北京大学



我来说说做了些什么

■ 把自己的大脑当作计算机:

- ◆ 在大脑中开辟一片“存储空间”存放输入的数字;
- ◆ 使用了一个另一片“存储空间”存放“特别数字”;
- ◆ 按照某种规律“反复”选定“输入存储空间中的数字”与“特别数字”比较;
- ◆ 每次比较时, 判断“选定的数字”是否大于“特别数字”;
- ◆ 如果大于, 重新“刷新”“特别数字”;
- ◆ 如果“特别数字”与其他数字都进行了比较, 说出“特别数字”;



北京大学



规整一下

■ 更清楚的描述方式:

- ◆ 在你的大脑里开辟一片存储空间存放输入的数字;
- ◆ 开辟另一个片存储空间存放“特别数字”;
- ◆ 从存储空间中的第一个数字开始, 直到最后一个数, 重复以下操作:
 - 比较“存储空间中的数字”与“特别数字”;
 - 如果“存储空间中的数字”大于“特别数字”;
 - 那么, 将“特别数字”换成“存储空间中的数字”;
- ◆ 说出“特别数字”;



北京大学



你能看懂多少？

```
#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;
    int i = 0;
    for(i = 0; i < 45; i++)
    {
        if(number[i] > max)
            max = number[i];
    }
    cout<<"The Maximal Number is:"<<max;
    return 0;
}
```



北京大学



问题来了！

1. 是不是“无论我们在程序里写什么，‘计算机’都能明白”？
2. 是不是“无论什么样的数字、符号都能够 在‘计算机’中进行存储”？
3. 用有限的几个关键字，能表达“各种各样、纷繁复杂”的处理逻辑吗？



北京大学



问题之一

1. 是不是“无论我们在程序里写什么，‘计算机’都能明白”？

◆ NO!

◆ 计算机语言提供了一些具有特定含义的“关键字”、“运算符”等；‘计算机’“只能明白”这些元素的特定含义。

■ 接下来的问题：

◆ 哪些关键字？含义是什么？



北京大学



计算机认识哪些词？

■ 计算机能够明白的“单词”

◆ 这些关键词在程序中有特定含义

auto	break	case	char	const
continue	default	do	double	else
enum	extern	float	for	goto
if	int	long	register	return
short	signed	sizeof	static	struct
switch	typedef	unsigned	union	void
volatile	while	bool	catch	class



北京大学

```

#include<iostream>
using namespace std;
int main()
{
    enum day{Mon, Tue, Wed, Thu, Fri, Sat, Sun};
    day workDay;
    double times, wages, hourlyRate, hours;
    cout<<"Enter the hourly wages rate."<<endl;
    cin>>hourlyRate;
    cout<<"Enter hours worked daily\n";
    for (workDay=mon; workDay<=sun; workDay++)
    {
        cin>>hours; //输入周一到周日的工作时间;
        switch(workDay)
        {
            case sat: times=1.5*hours; break;
            case sun: times=2.0*hours; break;
            default: times=hours; }
        wages = wages + times*hourlyRate;
    }
    cout<<"The wages for the week are "<<wages;
    return 0;
}

```



问题之二

2. 是不是“无论什么样的数字、符号都能够 在 ‘计算机’ 中进行存储”？

◆ NO!

◆ 计算机能够处理的数据的种类是有限制的。

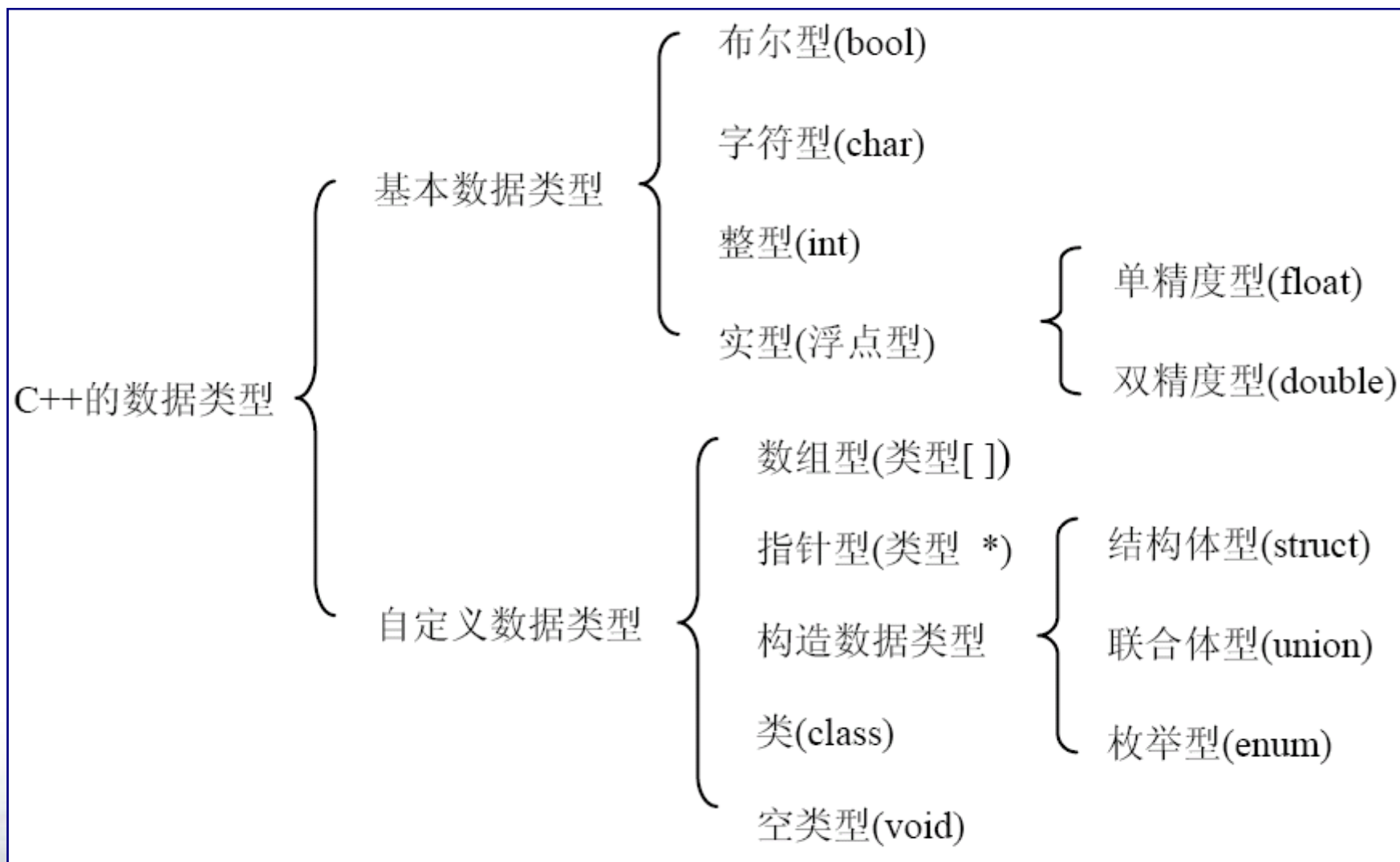
■ 接下来的问题:

◆ 哪些类型？类型的限制是什么？



北京大学

计算机能够哪种类型的数据?





关于变量的定义和使用

■ 变量的定义

◆ 变量类型 变量名称;

◆ **int i;**

◆ **float result;**

```
#include <iostream.h>
void main()
{
    int i;
    cout<<"请输入一个整数: ";
    cin>>i;
    cout<<"您输入的整数是: "<<i<<endl;
}
```



北京大学



程序如何输入/输出?

■ 如何从程序中输出信息?

◆ `cout<<“Welcome to Beijing!”<<endl;`

- `cout`是**输出流对象**（支持输出的东西）
- “<<”（插入操作符）
- `endl`表示换行，其作用相当于 “\n”.

■ 如何向程序中输入信息?

◆ `int x; cin>> x;`

◆ `cin`是输入流对象（支持输出的东西）

◆ “>>”（提取操作符）



北京大学



问题之三

3. 用有限的几个关键字，能表达“各种各样、纷繁复杂”的处理逻辑吗？

- ◆ 嗯，可以

- ◆ 程序的控制结构有几种基本的逻辑结构。

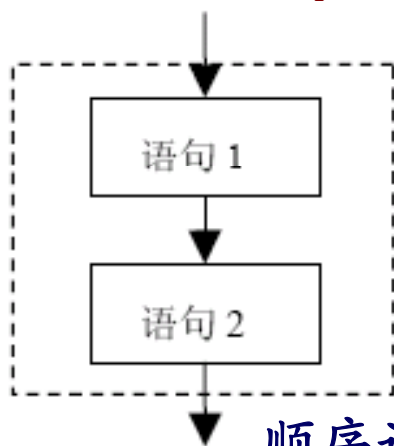
■ 接下来的问题：

- ◆ 哪几种？为什么只有这几种？怎么使用？

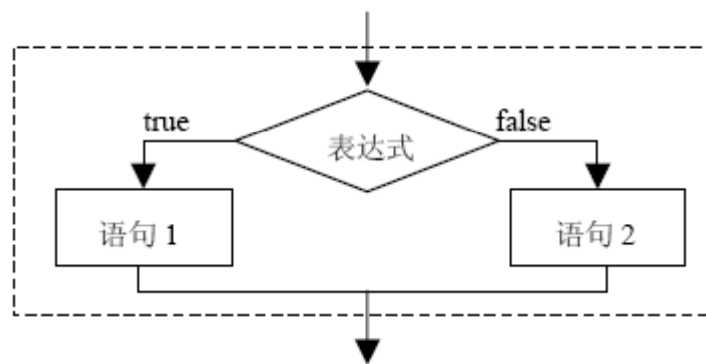


北京大学

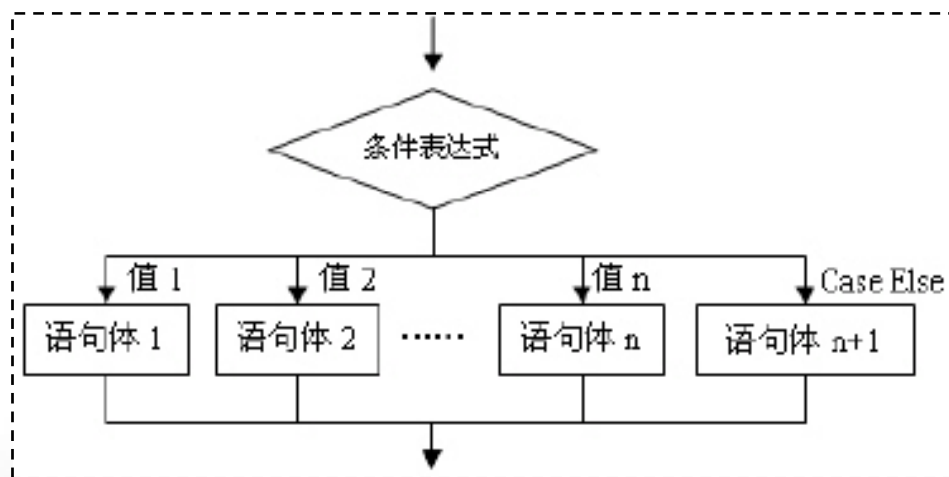
如何表达纷繁复杂的计算逻辑？



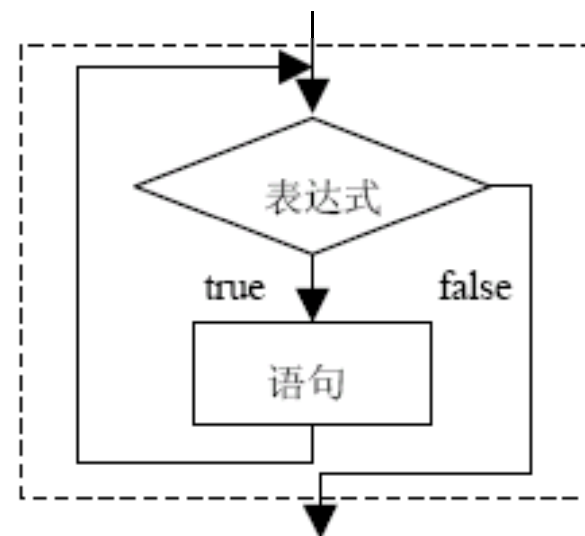
顺序语句



判断语句



分支语句



循环语句



北京大学



简单程序的组成

```
/** *****
/*  example.cpp  *
/** *****
#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;
    int i = 0;
    for(i = 0; i < 45; i++)
    {
        if(number[i] > max)
            max = number[i];
    }
    cout<<"The Maximal Number is:"<<max;
    return 0; //函数结束返回
}
```

注释

预编译：文件包含命令

声明名字空间

函数名

变量定义

循环控制语句

运算表达式

赋值语句

输入输出语句

函数体



北京大学



很简单的程序 (1)

```
/**  
**exam1.cpp  
**  
**  
#include <iostream>  
using namespace std;  
int main()  
{  
    cout <<"I am a student." << endl;  
    return 0;  
}
```



北京大学



最简单的程序

```
#include<iostream>
using namespace std;
int main()
{
    return 0;
}
```

```
#include<iostream.h>
void main()
{

}
```



北京大学



很简单的程序 (2) —— 顺序结构

```
#include <iostream>
using namespace std;
int main()
{
    int a, b, result;
    cout << "Please input two numbers:";
    cin >> a >> b;
    result = 3*a-2*b+1;
    cout << "result is" << result << endl;
    return 0;
}
```



北京大学




很简单的程序 (3) —— 顺序结构

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,temp;
    cout<<"Input a and b:"<<endl;
    cin>>a>>b;
    cout<<"a="<<a<<" , b="<<b;
    temp=a; a=b; b=temp;
    cout<<"a="<<a<<" , b="<<b;
    return 0;
}
```



北京大学




很简单的程序 (4) —— 分支结构

```
#include<iostream>
using namespace std;
int main()
{
    int x, y;
    cin>>x>>y;
    if(x>y)
        cout<<"Max number is: "<<x<<endl;
    else
        cout<<"Max number is: "<<y<<endl;
    return 0;
}
```



北京大学



很简单的程序（4）——分支结构

```
#include<iostream>
using namespace std;
int main( )
{
    int age;
    cout<<"Input your age:"<<endl;
    cin>>age;
    if(age >= 18)
        cout<<"You are a man/woman!"<<endl;
    else
        cout<<"You are a boy/girl!"<<endl;
}
```





很简单的程序 (5) —— 循环结构

```
#include<iostream>
using namespace std;
int main()
{
    int i = 0, sum = 0;
    for(i = 0; i <= 99; i = i + 1)
    {
        if(i % 2 == 0)
            sum = sum + i;
    }
    cout << "The total is " << sum << endl;
    return 0;
}
```

- 计算并输出前N(设N为99)个自然数中所有偶数之和。

```
#include <iostream.h>
void main()
{
    int score;
    char grade;
    cout<<"请输入一个成绩(0-100): ";
    cin>>score;
    if(score<0 || score>100)
    {
        cout<<"出错! 输入成绩超出(0-100)边界! "<<endl;
        return;
    }
    else if(score>=90)
        grade='A';
    else if(score>=80)
        grade='B';
    else if(score>=70)
        grade='C';
    else if(score>=60)
        grade='D';
    cout<<"输入的成绩是"<<grade<<"级! "<<endl;
}
```



关于C++程序的几点说明 (1)

■ 关于C++程序本身

- ◆ 每个C++程序都由很多个“函数”组成
 - 每个程序都以`main()`函数开始（程序入口）
 - 每个函数的形式都有“输入”和“输出”

<code>int</code>	<code>Max</code>	<code>(int x, int y)</code>
↑	↑	↑ ↑
函数类型	函数名	参数类型 参数名

- ◆ 如 `int main()`; 如 `double pow(double x, double y)`
- ◆ 语句分号结尾，一行可以写多个语句，一个语句可以写多行；格式变化不会影响程序运行；
- ◆ 程序中的注释可以放在`/*...*/`之间，或`//`之后；



北京大学

```
#include <iostream>
using namespace std;
int add(int x, int y);    //这是函数的声明;
int main( )
{
    int a = 30, b = 40, c = 0;
    cout<<"a="<<a<<" ,b="<<b<<endl;
    c = add(a, b);        //这是函数的调用;
    cout<<"a + b = "<<c<<endl;
    return 0;
}
int add(int x, int y)    //这是一个函数的定义;
{
    int z=x;
    x=y;
    y=z;
    return z;
}
```





关于C++程序的几点说明 (2)

■ 关于C++程序的运行环境

- ◆ C++程序均保存在 **xxx.cpp** 的文件之中;
- ◆ 不同的文件之间可以互相“引用”;
 - 通过“**include**”实现
- ◆ **xxx.cpp** 文件不能直接执行, 必须用一个软件进行“编译”..., 然后运行
 - **Visual C++ 6.0**
 - **Visual Studio .NET**
 - **Eclipse + CDT**



北京大学

C++程序的加工和执行



- 源程序，不能直接执行，必须先把它转换为可执行程序。

- 加工通常分两步：

- ◆ 编译

- 编译程序处理源程序，生成机器语言目标文件。目标文件不能执行，缺少必要的C++程序运行系统和库功能。

- ◆ 连接

- 把目标模块与运行系统、库模块组合起来，构成完整的可执行程序。



北京大学



关于C++程序的书写

■ 注意事项:

- ◆ 每行程序的书写必须按照逻辑进行**缩进**;
- ◆ 编写如下程序行时, 应该提供**注释**:
 - 变量的定义; //说明变量代表的意义或作用;
 - 函数的定义; //说明该函数的主要作用;
 - 复杂逻辑出现时; //解释一段语句的作用;
- ◆ **变量的命名**应该尽可能体现变量的含义;



北京大学



关于高级程序设计语言C++



北京大学



程序释义

■ 程序

◆ 计算任务的处理对象和处理规则的描述。

- 计算任务：任何以计算机为处理工具的任务都是计算任务。
- 处理对象是数据（如数字、文字、图形、图像、声音等，它们只是表示，而无含义）或信息（数据及有关的含义）。
- 处理规则一般指处理动作和步骤。

——计算机科学技术百科全书



北京大学



程序设计语言的产生

■ 通过何种语言与机器进行沟通

- ◆ 机器语言
- ◆ 汇编语言
- ◆ 高级语言



北京大学



程序设计语言的分类

■ 低级语言 之一

◆ 机器语言

- 机器语言是机器指令形成的语言;
- 形式为二进制编码, 机器可直接执行;
 - ◆ 00000001000000001000 数据装入寄存器0
 - ◆ 00000001000100001010 数据装入寄存器1
 - ◆ 000001010000000000001 寄存器0与1的数据乘
 - ◆ 00000001000100001100 数据装入寄存器1
 - ◆ 000001000000000000001 寄存器0与1的数据加
 - ◆ 000000100000000001110 保存寄存器0里的数据
- 难写难读, 人使用不便, 程序开发效率极低



北京大学



程序设计语言的分类

■ 低级语言 之二

◆ 汇编语言

- 汇编语言采用助记符号形式，有利于人阅读和使用。
汇编指令与机器指令对应。

- ◆ load 0 a 数据装入寄存器0
- ◆ load 1 b 数据装入寄存器1
- ◆ mult 0 1 寄存器0与1的数据乘（结果存在寄存器0）
- ◆ load 1 c 数据装入寄存器1
- ◆ add 0 1 寄存器0与1的数据加
- ◆ save 0 d 保存寄存器0里的数据

- 比机器语言好多了，但还不够好用！



北京大学



程序设计语言的分类

■ 汇编语言的特点：

- ◆ 每条指令的意义容易理解。
- ◆ 程序粒度太小，细节太多。
- ◆ 程序无结构，缺乏组织手段。
- ◆ 写大程序仍然很困难。
- ◆ 最早时通过手工翻译为机器指令。
 - 后来人们开发出“汇编系统”，让计算机去完成程序翻译工作

■ 计算机无法直接执行汇编语言程序，执行前需要翻译为机器指令程序。



北京大学



程序设计语言的分类

■ 高级语言

- ◆ 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；
- ◆ 用 **类似数学** 的表达式形式描述基本计算；
- ◆ 具有 **类似文字** 的表现形式；
- ◆ 提供高级操作 **流程控制** 手段和 **程序组织** 手段。

```
00000001000000001000 load 0 a
00000001000100001010 load 1 b
00000101000000000001 mult 0 1
00000001000100001100 load 1 c
00000100000000000001 add 0 1
00000010000000000110 save 0 d
```

■ C语言写同样的程序：

- ◆ $d = a * b + c;$



北京大学



我们学的是什么高级语言？



C 语言部分

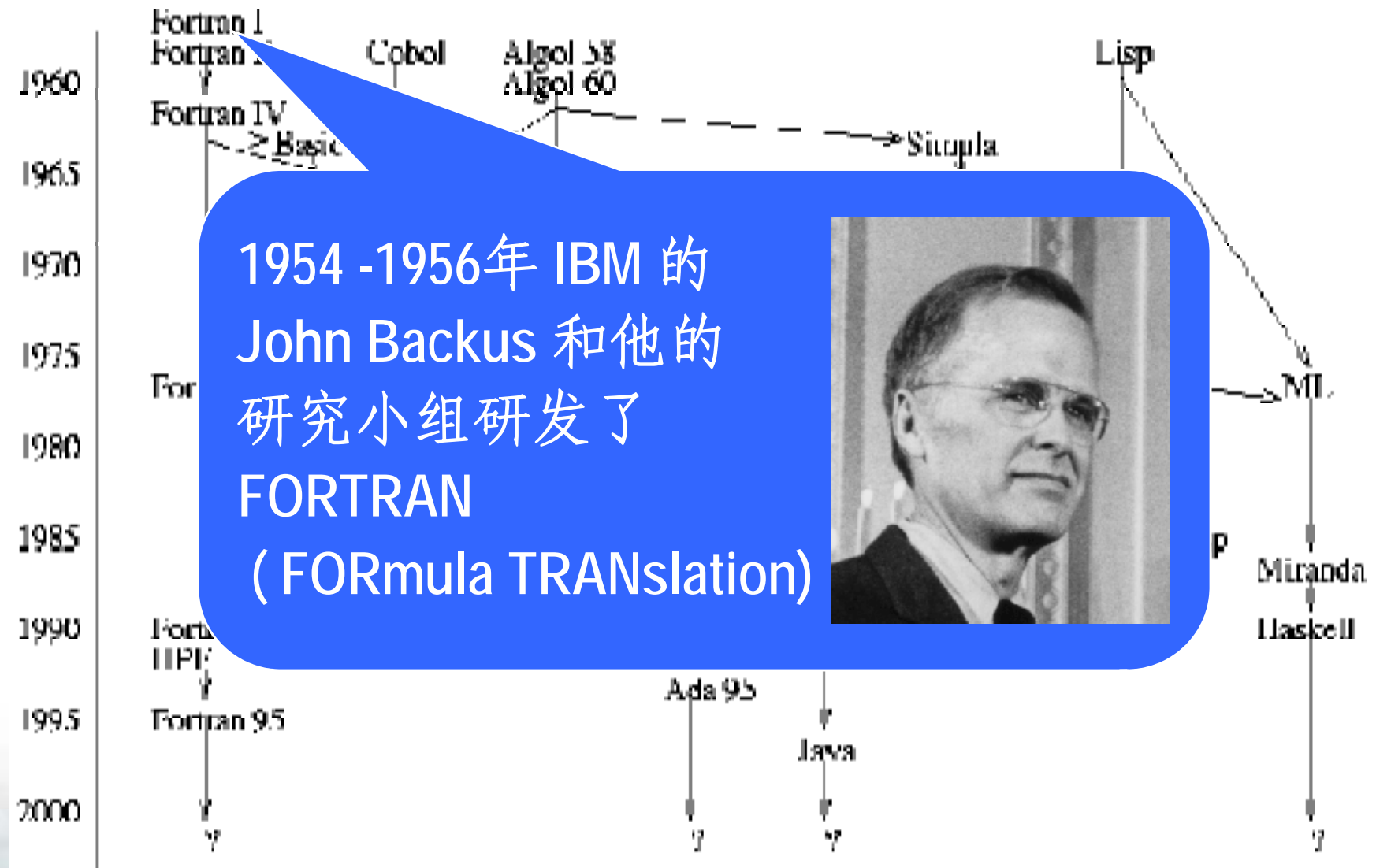
C++ 语言
(面向对象部分)



北京大学



高级语言的发展历程



北京大学



C程序设计语言的历史

- ◆ Algol 60: 1960年1月, 图灵奖获得者Alan J.Perlis在巴黎举行的软件专家讨论会上, 发表了“**算法语言Algol 60报告**”, 确定了程序设计语言Algol 60。 (**A语言**)
- ◆ 1963年, 剑桥大学在ALGOL 60的基础上推出了CPL (Combined Programming Language) 语言, 但规模比较大, 难以实现。1967年, 剑桥大学的Martin Richards对CPL语言作了简化, 推出了BCPL (Basic Combined Programming Language) 语言。1970年贝尔实验室的Ken Thompson进一步简化, 设计出了B语言, 并用B语言写第一个UNIX操作系统, 在PDP-7上实现。
- ◆ C语言是在1972-1973年间由UNIX的研制者Dennis Ritchie和Ken Thompson在B语言的基础上发展和完善起来的。



北京大学



C 语言的版本

■ K&R C

- ◆ 在1978年，Kernighan和Ritchie的《The C Programming Language》第一版出版。在以后的几年里，《The C Programming Language》一直被广泛作为C语言事实上的规范。在这本书中，C语言通常被表述成“K&R C”。

■ ANSI C和ISO C（1989年）

- ◆ 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ISO/IEC JTC1/SC22/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。

■ ANSI C和ISO C（1995年）

- ◆ 作为对标准的维护与更新，WG14工作小组在1995年对1985年颁布的标准做了两处技术修订（缺陷修复）和一个补充（扩展）。

■ 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。

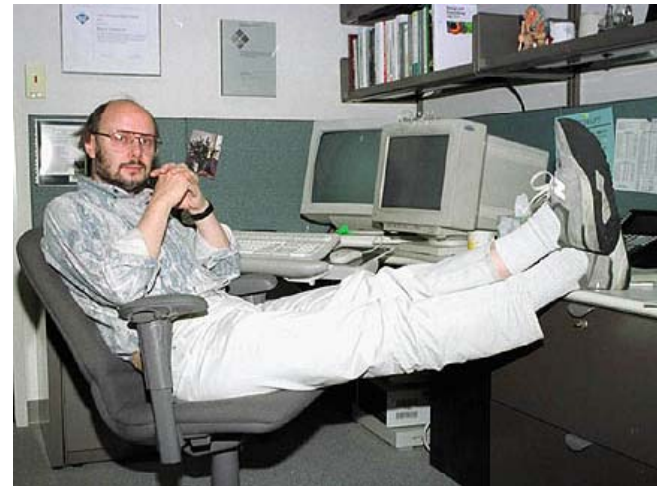


北京大学



From C to C++

- 美国AT&T贝尔实验室的本贾尼（Bjarne Stroustrup）在20世纪80年代初期发明并实现了C++（最初被称作“C with Classes”）。
- 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。
- 2003年发布一个C++标准的修正版本；每5年视实际需要更新一次标准。



北京大学



关于课程资源

■ 课程网站:

- ◆ <http://course.pku.edu.cn/webapps/login/>
- ◆ 用户名, 密码: 学号;

■ 助教:

- ◆ 王乐业 (wangleye@gmail.com)
- ◆ 王松林 (wangsonglin@icst.pku.edu.cn)
- ◆ 刘 飞 (liufei08@sei.pku.edu.cn)
- ◆ 何 宇 (heyu813@pku.edu.cn)

■ 邮件公约:

- ◆ 为了便于区分邮件来源, 请各位同学给我和助教发信时, 请将邮件题目定制为:
- ◆ “计算概论A + 邮件标题” 的形式;



北京大学



好好想想，有没有问题？

谢谢！



北京大学