

§ 1. C++的初步知识

1. 1. 发展背景

1. 1. 1. 计算机内数据的表示

采用二进制，只有两个数码（0/1），任何复杂的数据都是由0/1的基本表示组成的

{ 数字(有数值含义的数字序列)
文本(包括数码，即无数值含义的数字序列)
静态图像
动态视频
声音

1. 1. 2. 计算机程序设计语言的发展

机器语言：只有0/1，面向具体机器，可读性差，无移植性

汇编语言：ADD、MOV等助记符，面向具体机器，可读性差，可移植性差

高级语言：自然语言表达，与具体机器无关，可读性好，可移植性好；但与硬件差距较大，
对系统控制不易

★ 计算机能直接识别的只有机器语言

★ 将汇编、高级语言转换为机器语言的工具就是语言处理程序，转换的过程称为编译和链接

§ 1. C++的初步知识

1. 1. 发展背景

1. 1. 3. C++的发展历史

1960	-	1963	-	1967	-	1970	-	1972	-	1983	-	1989	-	1990	-	1999	-	2011
ALGOL60		CPL		BCPL		B		C		开始ANSI		ANSI-C89		ISO-C90		C99		C11
1982	-	1997	-	1998	-	2003	-	2011	-	2014								
C++		ANSI		ISO		C++03		C++11		C++14								

1. 1. 4. C语言的特点

- ★ 语言简洁，使用方便，书写自由
- ★ 运算符丰富
- ★ 数据结构种类多
- ★ 面向过程，具备结构化的控制语句, 易于实现结构化的程序设计
- ★ 语法限制不严，灵活程度大（也易带来错误）
- ★ 能直接访问物理地址，实现位操作，既具有高级，又具有低级语言的特性，适用于编写系统软件（UNIX、LINUX）
- ★ 目标代码质量高，执行速度快
- ★ 可移植性好

1. 1. 5. C++语言的特点

- ★ 与C完全兼容，原来的C语言代码可直接在C++下编译
- ★ 对C语言的功能进行了扩充
- ★ 支持面向对象机制，易于开发大型的软件

§ 1. C++的初步知识

1.2. 最简单的C++程序

1.2.1. 例

P.5~9 例 1.1、1.2、1.3及说明 (P.9~11 例 1.4及说明暂忽略, 不必看)

```
#include <iostream>
using namespace std;
int main()
{
    cout << "This is a C++ program";
    return 0;
}
```

例 1.1

功能: 在屏幕上输出"This is a C++ Program."

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

例 1.2

功能: 从键盘上输入两个整数, 在屏幕上输出和

```
#include <iostream>
using namespace std;
int max(int x, int y)
{
    int z;
    if (x>y) z=x;
    else z=y;
    return (z);
}
```

例 1.3

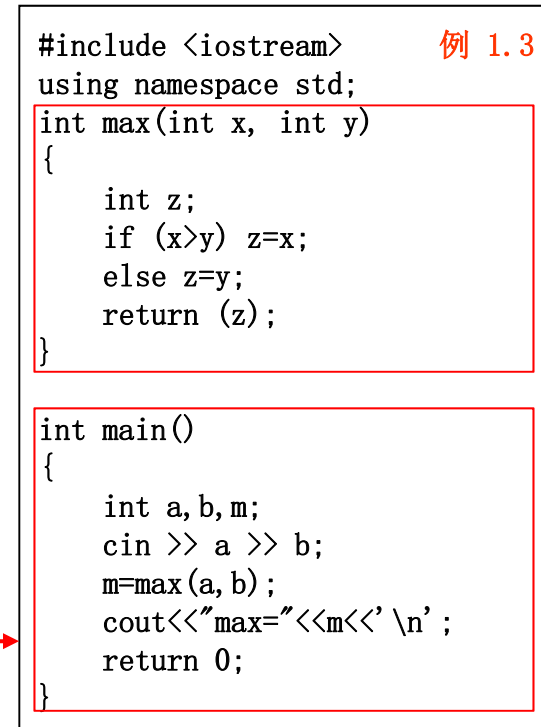
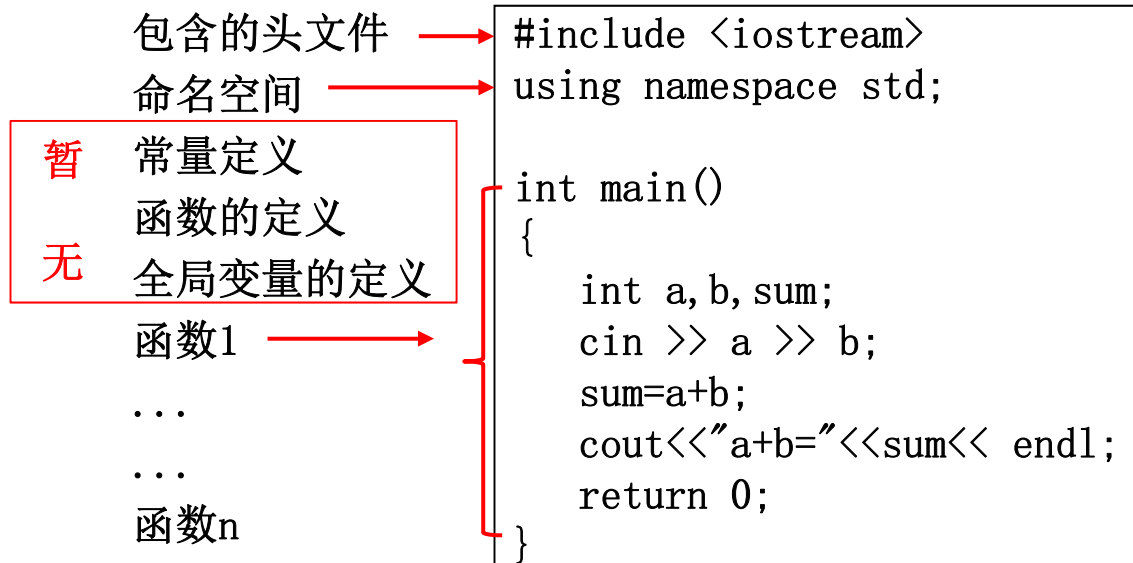
```
int main()
{
    int a, b, m;
    cin >> a >> b;
    m=max(a, b);
    cout << "max=" << m << '\n';
    return 0;
}
```

功能: 从键盘上输入两个整数,
求最大值并在屏幕上输出

§ 1. C++的初步知识

1. 2. 最简单的C++程序

1. 2. 2. 程序结构的基本形式

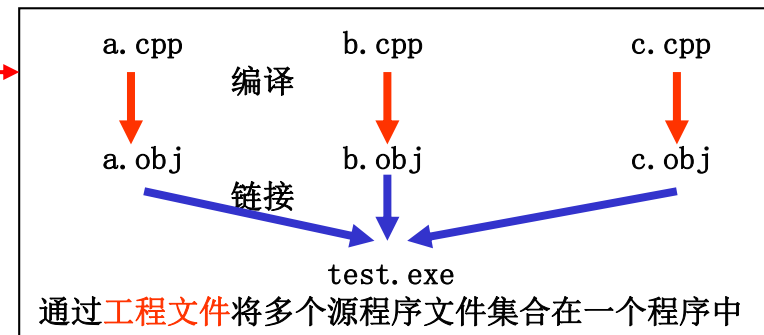


★ C++程序由函数组成，函数是C++程序的基本单位

★ 一个C++程序可由若干源程序文件 (*.cpp) 组成，
每个源程序文件可以包含若干函数

★ 有且仅有一个名为main()的函数，称为主函数，
程序的执行从它开始

★ C++提供许多库函数 (已做好的)



§ 1. C++的初步知识

1. 2. 最简单的C++程序

1. 2. 3. 函数的组成

函数返回类型 函数名（形式参数表）

{

函数体（局部变量的定义、函数体的可执行部分）

}

```
int max(int x, int y)
{
    int z;
    if (x>y) z=x;
    else z=y;
    return (z);
}
```

```
#include <iostream>
using namespace std;
int max(int x, int y)
{ ...
}
int main()
{ ...
}
```

```
#include <iostream>
using namespace std;
int main()
{ ...
}
int max(int x, int y)
{ ...
}
```

★ 从main()开始执行，函数相互间的位置不影响程序的正确性

§ 1. C++的初步知识

1. 2. 最简单的C++程序

1. 2. 3. 函数的组成

★ 从main()开始执行，函数相互间的位置不影响程序的正确性

★ 函数平行定义，嵌套调用

★ 函数由语句组成，一个语句以;结尾（必须有），语句分为定义语句和执行语句，定义语句用于声明某些信息，执行语句用于完成特定的操作

★ 书写格式自由，可以一行多个语句，也可以一个语句多行（以\表示分行）

例 1.3

```
#include <iostream>
using namespace std;
int max(int x, int y)
{
    int z;
    if (x>y) z=x;
    else z=y;
    return (z);
}

int main()
{
    int a,b,m;
    cin >> a >> b;
    m=max(a,b);
    cout<<"max="<<m<<' \n' ;
    return 0;
}
```

定义

调用

<pre>#include <iostream> using namespace std; int main() { cout << "This is a C++ program"; return 0; }</pre>	<pre>#include <iostream> using namespace std; int main() { cout << "This is a C++ program";return 0;}</pre>
<pre>#include <iostream> using namespace std; int main() { cout \ << \ "This is a C++ program"; return 0; }</pre>	<pre>#include <iostream> using namespace std; int main() { cout << "This is a C++ program"; return 0; }</pre>

同一个程序，这些格式编译器都认为是正确的

§ 1. C++的初步知识

1. 2. 最简单的C++程序

1. 2. 3. 函数的组成

★ 可以用/* ... */ (多行) 或 //... (单行) 两种方式加入注释，注释中的内容是为了增加程序的可读性，因此不需要符合C++的语法及规定

```
#include <iostream>
using namespace std;    //命名空间

int main()               /* function main */
{
    cout << "This is a C++ program";
    return 0;            /* fanhui :-^ ^-^ */
}
```

```
#include <iostream>
using namespace std;    //命名空间

/* 下面是主函数，仅用于输出一个
   字符串，输出后程序即运行结束
   P.5 例 1.1 */
int main()               /* function main */
{
    cout << "This is a C++ program";
    return 0;            /* fanhui :-^ ^-^ */
}
```

★ 系统提供的库函数和自己编写的函数调用方法相同

§ 1. C++的初步知识

1. 3. 编译器的安装与使用、编程的基本步骤

1. 3. 1. VS2017等编译器的安装及配置

另行下发文档

1. 3. 2. VS2017等编译器的使用(编译C++程序)

另行下发文档

1. 3. 3. 上机的基本过程

- ★ 建立新的源程序文件 (*.cpp)
- ★ 对源程序文件 (*.cpp) 进行编译，检查其中的语法错误，错误分致命错误(error)及警告错误(warning)，生成编译结果文件 (*.o/*.obj)
- ★ 对编译结果文件 (*.o/*.obj) 进行链接，检查链接错误，形成可执行程序文件 (*.exe)
- ★ 运行可执行程序文件 (*.exe)，检查其中的逻辑错误，验证程序的正确性
- ★ 编译执行过程中会产生很多临时文件（交作业时只有*.cpp是需要的）

1. 4. 作业命名及格式要求

另行下发文档