小马加编信息学教案(一)

C++ 第一课

一. 课程内容

1. Dev-C++ 的使用
2. 简单代码的编译与运行

二. 知识讲解

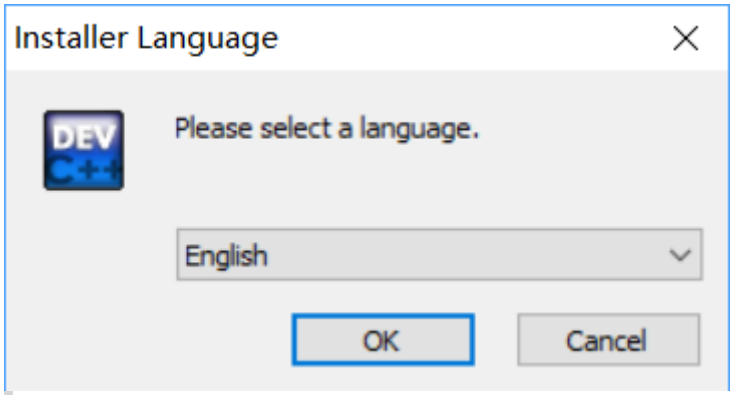
1. Dev-C++ 的使用

1.1 什么是 Dev-C++ ?

> Dev-C++ 是一款 Windows 环境下的 C++ 集成开发环境。
> 我们在以后的很长一段时间内都需要使用 Dev-C++ 进行 coding

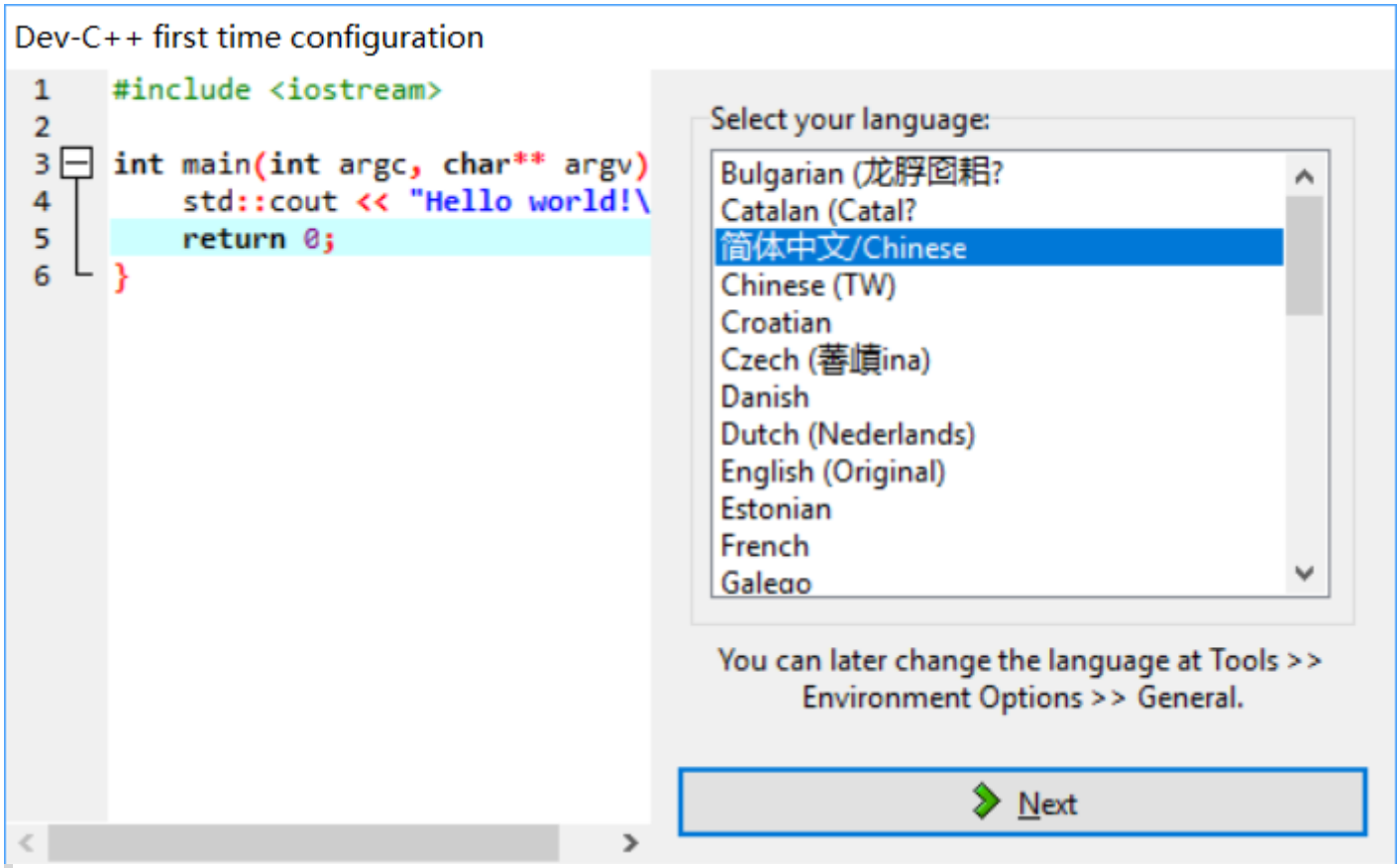
1.2 安装过程注意事项

> 下载 Dev-Cpp.5.11 安装包打开即可进行安装!



> 第一个界面中只能选择 English ， 继续一直按正常推荐点击即可

> 安装完成后运行 Dev-C++ 在第一个界面中选择简体中文



> 最后打开即可开始使用 Dev-C++

1.3 需要掌握的基础操作

1. 编辑程序
2. **保存程序(Ctrl + S)**
3. **打开程序(Ctrl + O)**
4. 编译程序(F9)

> 5. 运行程序(F10)
> 6. **编译、运行程序(F11)**
> 7. 调试程序(F5)

2. 简单代码的编译和运行

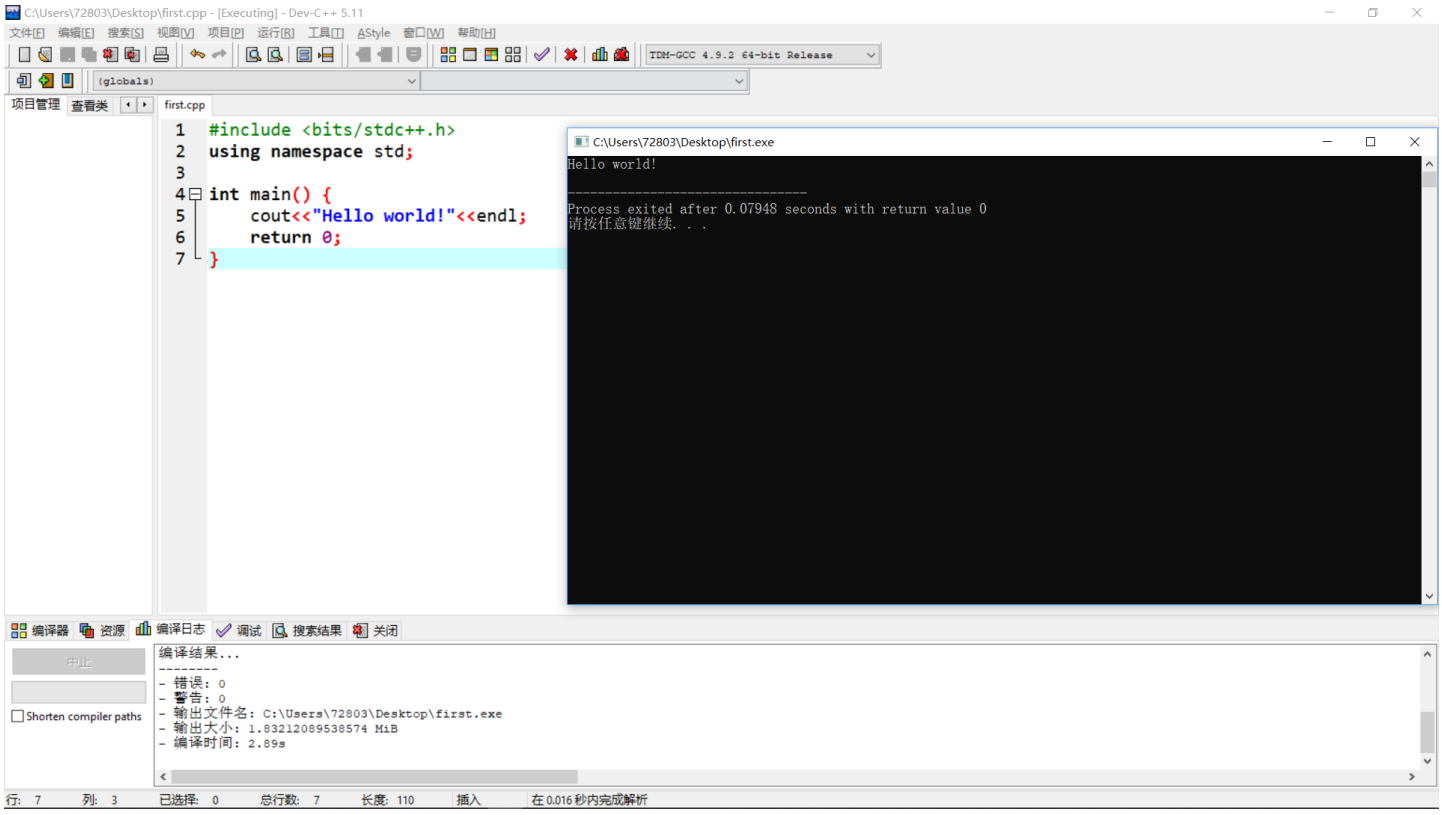
2.1 什么是 C++ ?

> C++ 是一种面向对象的程序设计语言，同时又兼备了结构化程序设计语言的一些特点。
> **C++ 即将成为 NOI 系列比赛唯一支持的程序设计语言。**

2.2 人生第一段代码： Hello world !

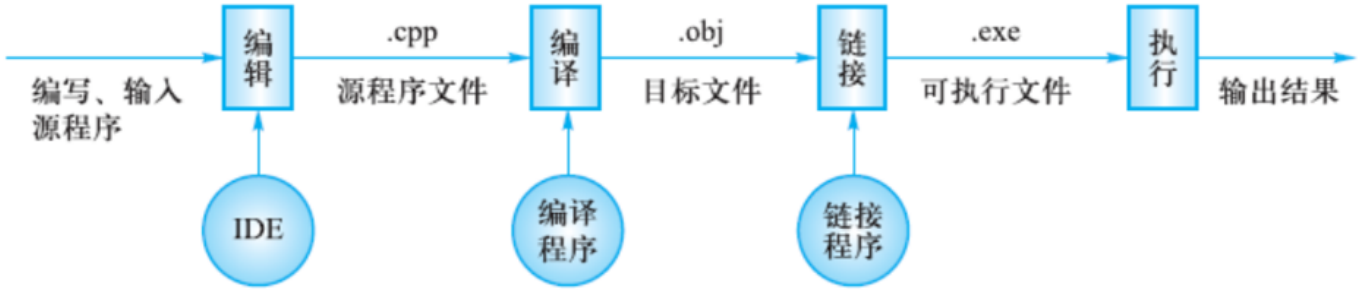
```
#include<bits/stdc++.h>
using namespace std;

int main() {
    cout<<"Hello world!"<<endl;
    return 0;
}
```

> 在Dev中编辑输入上面这段代码。并尝试按 F11 进行编译运行，得到你的第一个成功运行的输出 Hello world! 的程序，如下图。



2.3 从代码到程序

> 你是否想过，从一段英文字母组成的代码是如何变成一个可以运行的电脑程序的呢?



2.4 程序解释

> **C++ 有很多固定的语法规则，需要我们在理解中去记忆并且掌握**

```
#include<bits/stdc++.h>

//#include是C++语句中的一个指令，可以将C++语言库中头文件的内容添加到程序中

//包含头文件可以让你的程序可以使用C++的库函数以及一些标准

//尖括号中包含的是头文件的名称，表明程序要包含哪一个头文件

using namespace std;
//using表示程序正在使用,namespace的含义是名称空间

//std是标准(standard)的缩写

//可以简单的理解记忆为使用C++的标准

//C语言所有运行语句的后面都一定要加上分号!

//int main是主函数

//int main之后跟着一对圆括号，紧跟着是一对花括号，花括号中是函数的主体部分

int main() {
    cout<<"Hello world!"<<endl;//输出"Hello world"(不带双引号)
    return 0;//返回一个0值表示程序正常结束，并结束程序
}
```

3. C++简单概念初步了解

3.1 关键字

> 关键字 （Keywords） 是**由 C++ 语言规定的具有特定意义的字符串，通常也称为保留字**，例如 `int`、`char`、`long`、`float`、`unsigned` 等。
> 我们定义的标识符不能与关键字相同，否则会出现错误。
> 你也可以将关键字理解为**具有特殊含义的标识符，它们已经被系统使用，我们不能再使用了**

> 以下是 C++ 关键字表

保留字(关键字)						
C++系统中预定义的、在语言或编译系统的实现中具有特殊含义的单词：						
if	else	while	signed	throw	union	this
int	char	double	unsigned	const	goto	virtual
for	float	break	auto	class	operator	case
do	long	typedef	static	friend	template	default
new	void	register	extern	return	enum	inline
try	short	continue	sizeof	switch	private	protected
asm	while	catch	delete	public	volatile	struct

3.2 注释

> 注释 （Comments） 可以出现在代码中的任何位置，用来向用户提示或解释代码的含义。
> **程序编译时，会忽略注释，不做任何处理，就好像它不存在一样。**
> C++ 语言支持**单行注释和多行注释**:
> 单行注释以 `//` 开头，直到本行末尾（不能换行）；
> 多行注释以 `/*` 开头，以 `*/` 结尾，注释内容可以有一行或多行。

3.3 表达式和语句

> **表达式可以看做一个计算的公式，往往由数据、变量、运算符等组成**，例如 `3*4+5`、`a = c = d` 等，**表达式的结果必定是一个值**

> 语句的范围更加广泛，**不一定是计算，不一定有值，可以是某个操作、某个函数、选择结构、循环等**。

- 以分号 ; 结束的往往称为语句，而不是表达式

三. 经典例题

1. 举一反三,试着独立编写一个输出 `I love progroming` 的C++程序

2. 将下列代码编辑输入至 Dev-C++ 中，并成功编译运行得到结果

> 输出一棵星星树

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    cout << "     *\n";   //"\n"也表示换行
    cout << "    ***\n";
    cout << "   *****\n";
    cout << "  *******\n";
    cout << "    ***\n";
    cout << "    ***\n";
    cout << "    ***\n";
    cout << "    ***\n";
    cout << "    ***\n";
    return 0;
}
```

3. 将下列代码编辑输入至 Dev-C++ 中，并成功编译运行得到结果

> 输出直角三角形

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    for(int i = 1; i <= 10; i++) {
        for(int j = 1; j <= i; j++)
            cout<<'*';
        cout<<endl;
    }
    return 0;
}
```

四. 提高巩固

1. 独立编写一个输出 `I like C++` 的C++程序
2. 尝试回忆并指出你写出的程序的各个部分的内涵与基本概念
3. 将下列代码编辑输入至 Dev-C++ 中，并成功编译运行得到结果

> **A+B 计算器**

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    int a, b;
    cin>>a>>b;
    int c = a + b;
    cout<<c<<endl;
    return 0;
}
```

> 该程序成功编译运行跳出窗口后，你可以通过键盘输入两个数字，并按下回车，即可得到输出结果。