



中國人民大學
RENMIN UNIVERSITY OF CHINA

第2讲 认识C与编程准备

余力

buaayuli@ruc.edu.cn



中國人民大學
RENMIN UNIVERSITY OF CHINA



01. 认识C语言

上节课回顾

■ 1.1 计算机怎样理解信息

○ 练习题：写出下面一组数对应的英文字符

• $(76)_{10}$

• $(01001111)_2$

• $(56)_{16}$

• $(105)_8$

○ 答案：

66	B	82	R	98	b	114	r	
67	C	83	S	99	c	115	s	
68	D	84	T	100	d	116	t	
69	E	85	U	101	e	117	u	
70	F	86	V	102	f	118	v	
71	G	87	W	103	g	119	w	
72	H	88	X	104	h	120	x	
73	I	89	Y	105	i	121	y	
74	J	90	Z	106	j	122	z	
75	K	91	[107	k	123	{	
76	L	92	\	108	l	124		
77	M	93]	109	m	125	}	
78	N	94	^	110	n	126	~	
79	O	95	_	111	o	127	△	^Backspace 代码：DEL

上节课回顾

■ 1.2 计算机硬件体系结构

- 冯诺依曼体系结构
- 存储程序 + 程序控制

■ 1.3 程序设计简介

- 机器语言 → 汇编语言 → 高级语言

■ 1.4 教学计划与考核要求

- 程序设计是**练**出来的
- 期中期末考试：上机考试
- 平时考核：上机练习题 + 课程设计

内容提要

- 2.1 认识C语言
- 2.2 如何开始写C程序
- 2.3 C程序的基本结构



中國人民大學
RENMIN UNIVERSITY OF CHINA



01. 认识C语言

C语言发展简史 (1)

- 1972年：美国贝尔实验室的Dennis Ritchie创造了C语言
- 1975年UNIX第6版发布，C优点突出引关注。
- 1977年,Ritchie发表了不依赖于具体机器系统的C语言编译文本《可移植C语言编译程序》
- 1978年Brian W.Kernighan和Dennis M.Ritchie出版影响深远的名著《The C Programming Language》，被称为标准C。
- 之后，C语言先后移植到大、中、小、微型计算机上，已独立于UNIX和具体机器，风靡世界,成为最广泛的几种计算机语言。

C语言发展简史 (2)

■ C的标准化过程:

- 1983年,美国国家标准化协会(ANSI) 制定新的标准ANSI C , 比标准C有了很大的发展
- 1988年K & R按照 ANSI C修改了他们的《The C Programming Language》
- 1987年ANSI公布了新标准——87 ANSI C。
- 1990年国际标准化组织接受了87 ANSI C为ISO C 的标准 (ISO9899—1990)
- 1994年ISO又修订了C语言标准

■ 目前流行的C语言编译系统大多以ANSI C为基础进行开发的

C语言的特点



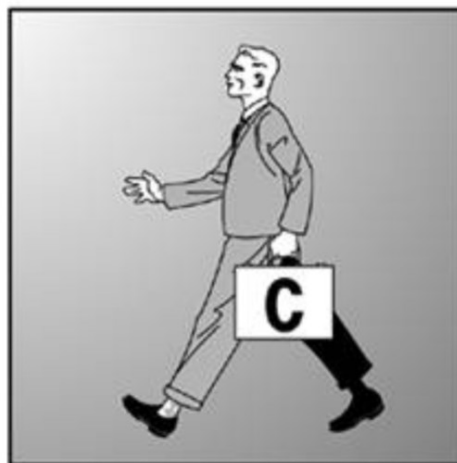
Powerful control structures



Fast



Compact code — small programs



Portable to other computers



中國人民大學
RENMIN UNIVERSITY OF CHINA



02. 开始写C程序

集成开发环境 (IDE)

■ 为什么使用集成开发环境

- 编辑C代码的小助手
- 提供编辑、编译、运行、调试一站式功能

■ 推荐使用哪些IDE

- 教材推荐: Visual Studio 6.0
- Windows系统: Dev C++
- Mac系统: VScode
- 可以自由选择自己喜欢的IDE

案例：Dev C++使用 (1)

■ 下载并安装Dev C++

➤ <https://sourceforge.net/projects/orwelldvcpp/>



Dev-C++

A free, portable, fast and simple C/C++ IDE

Brought to you by: [orwelldvcpp](#)

Summary

Files

Reviews

Support

External Link ▾

Tracker

Code

Forums

★ 4.6 Stars (130)

↓ 86,786 Downloads (This Week)

📅 Last Update: 2015-06-14

sf

Download

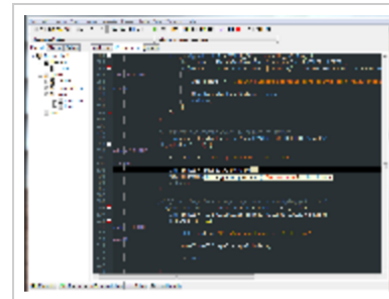
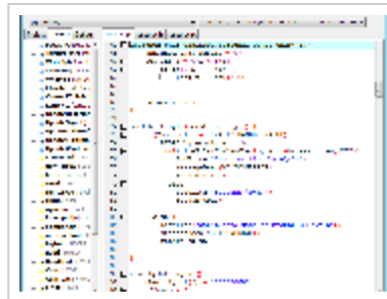
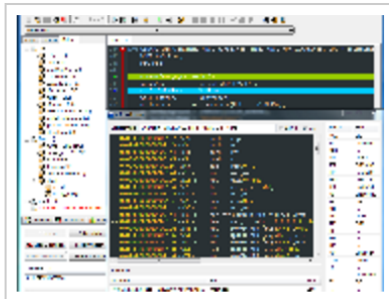
Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup.exe

🐦 Tweet

👍 53

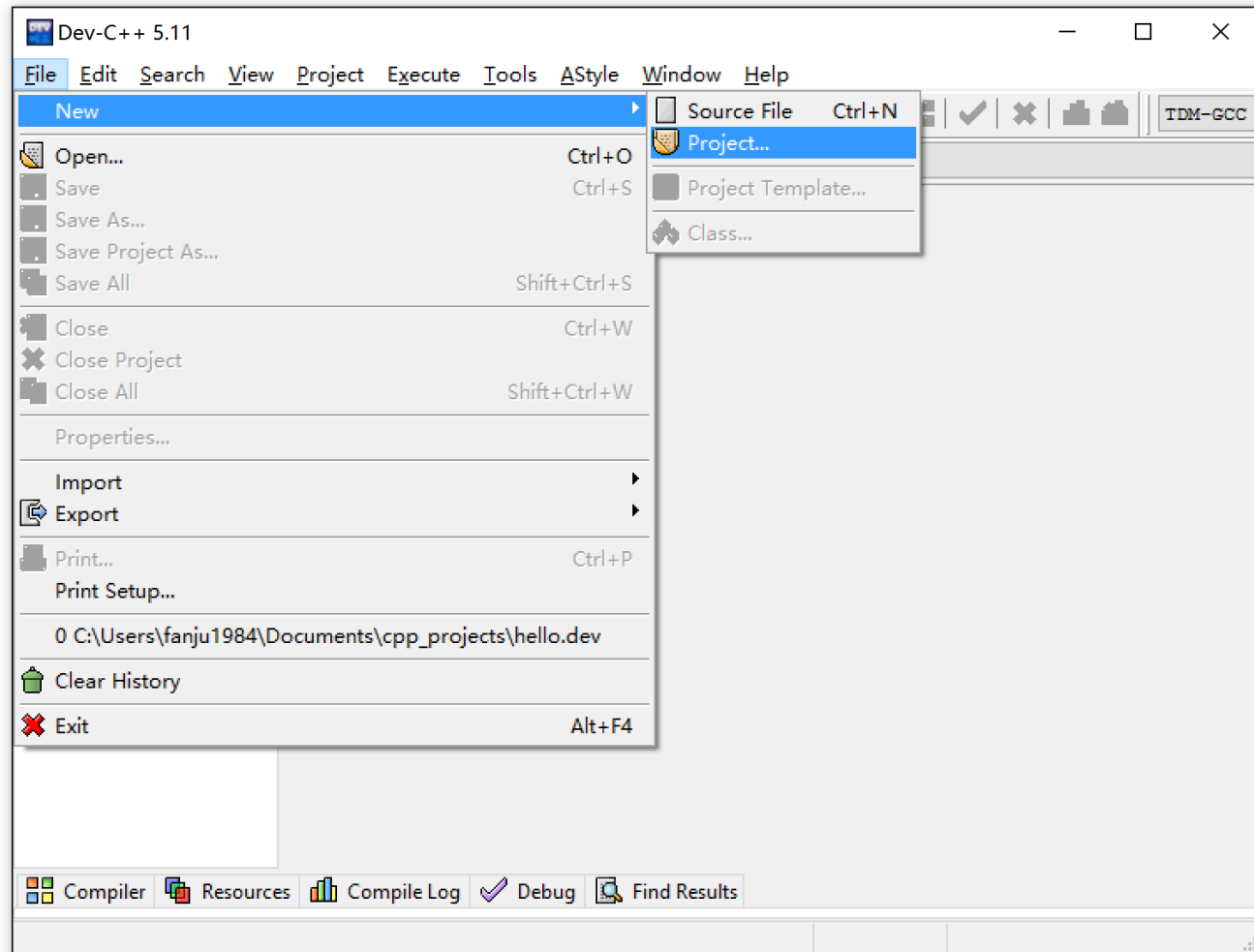


[Browse All Files](#)



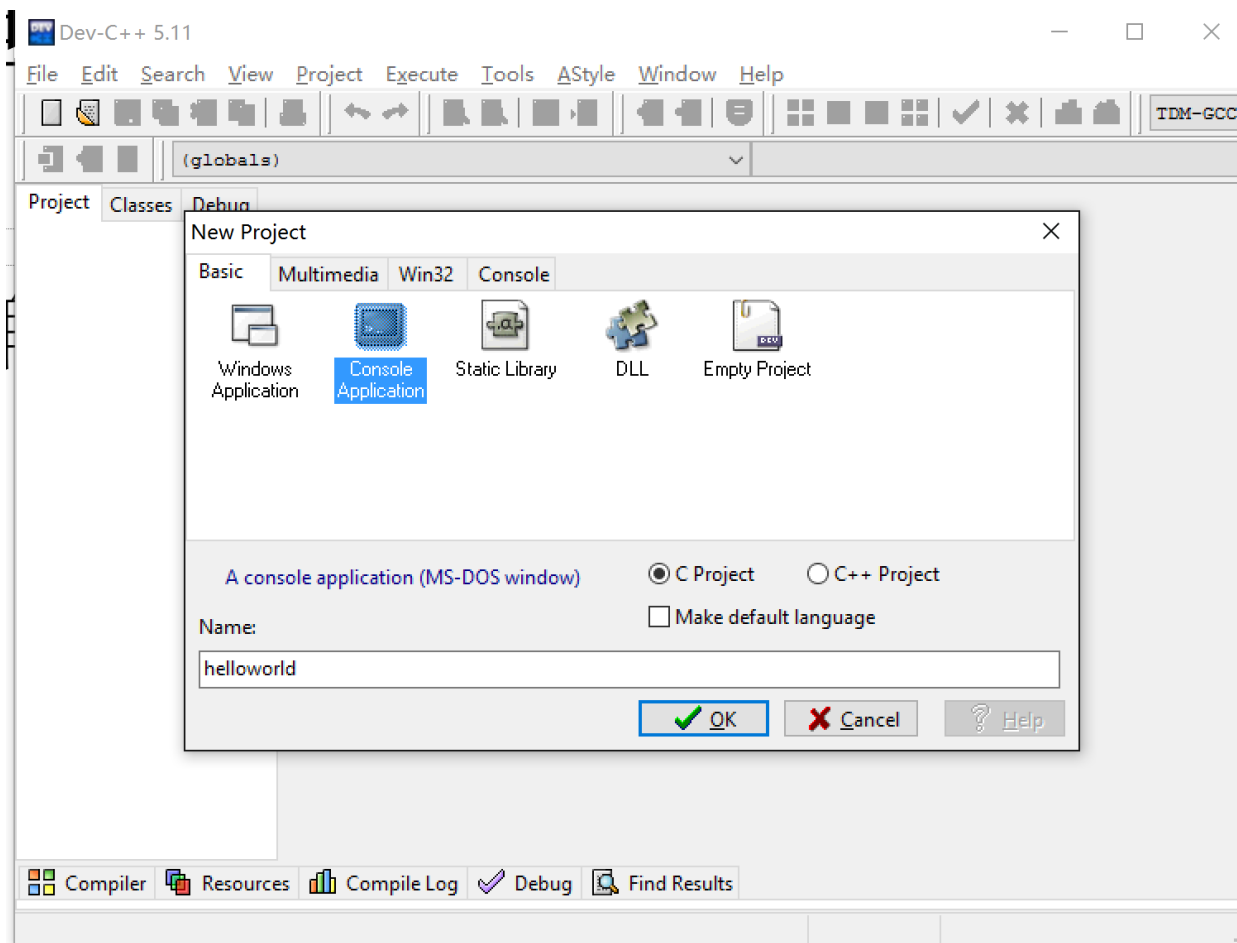
案例：Dev C++使用 (2)

■ 新建工程 (Create Project)



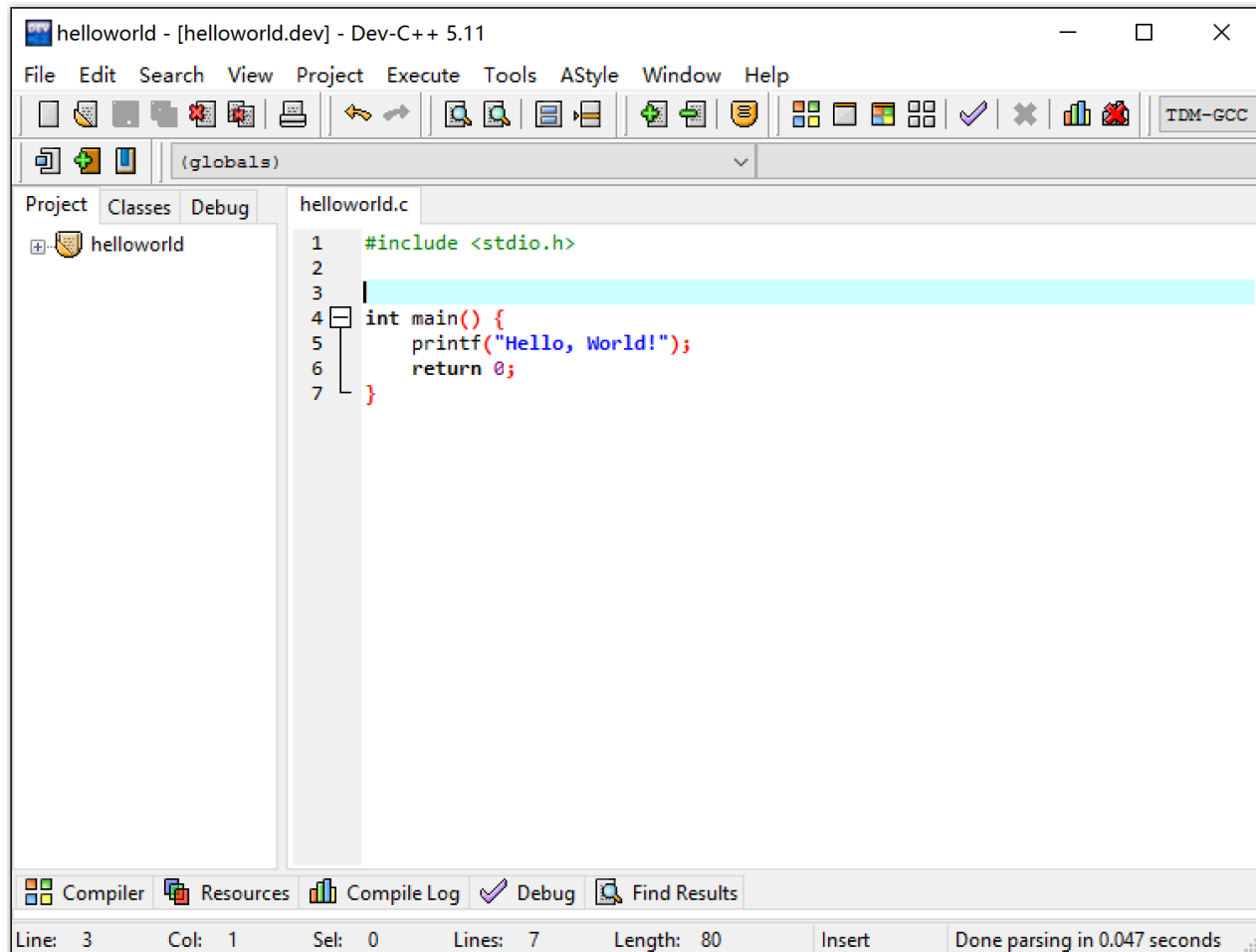
案例：Dev C++使用 (3)

■ 设置工程基本信息



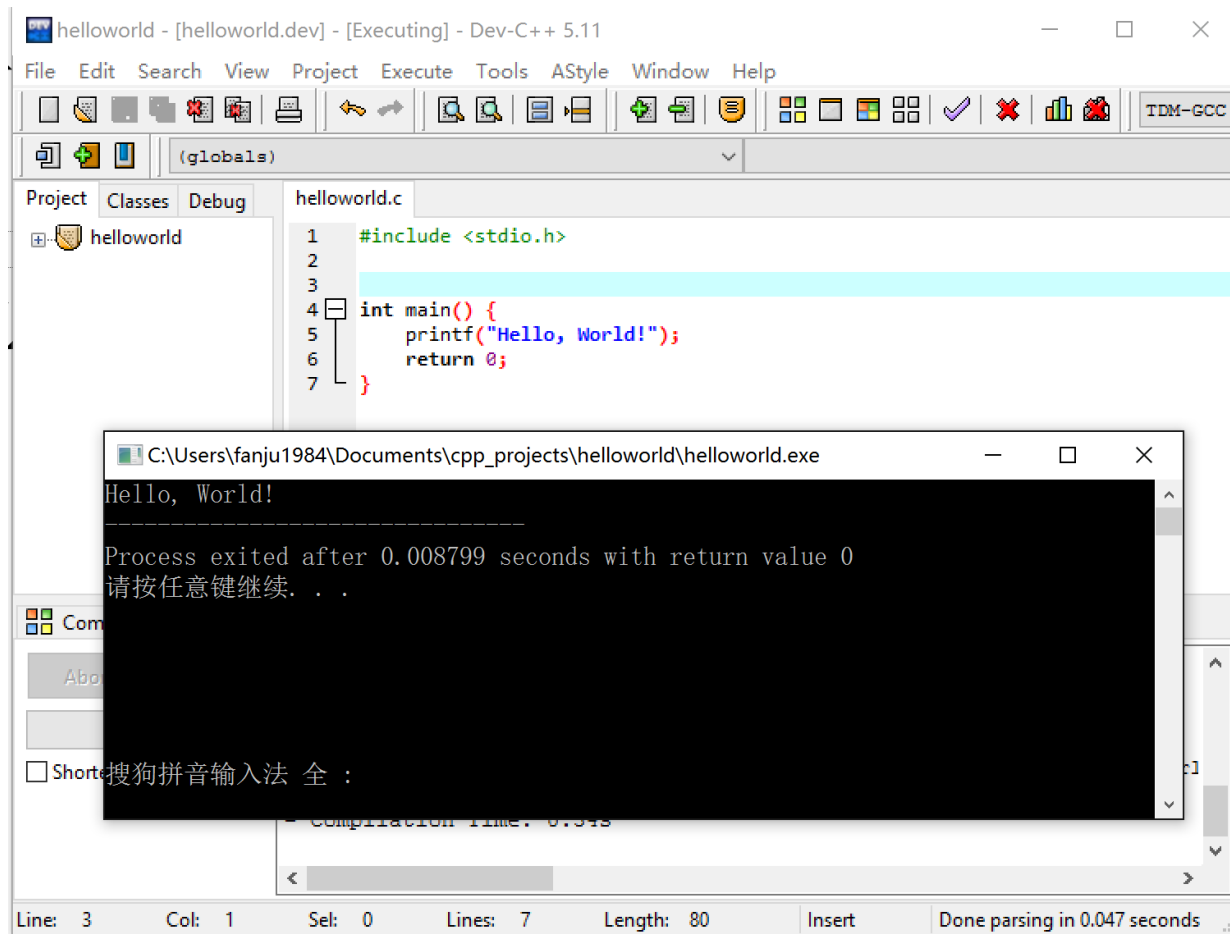
案例：Dev C++使用 (4)

■ 编辑C程序(.c文件)



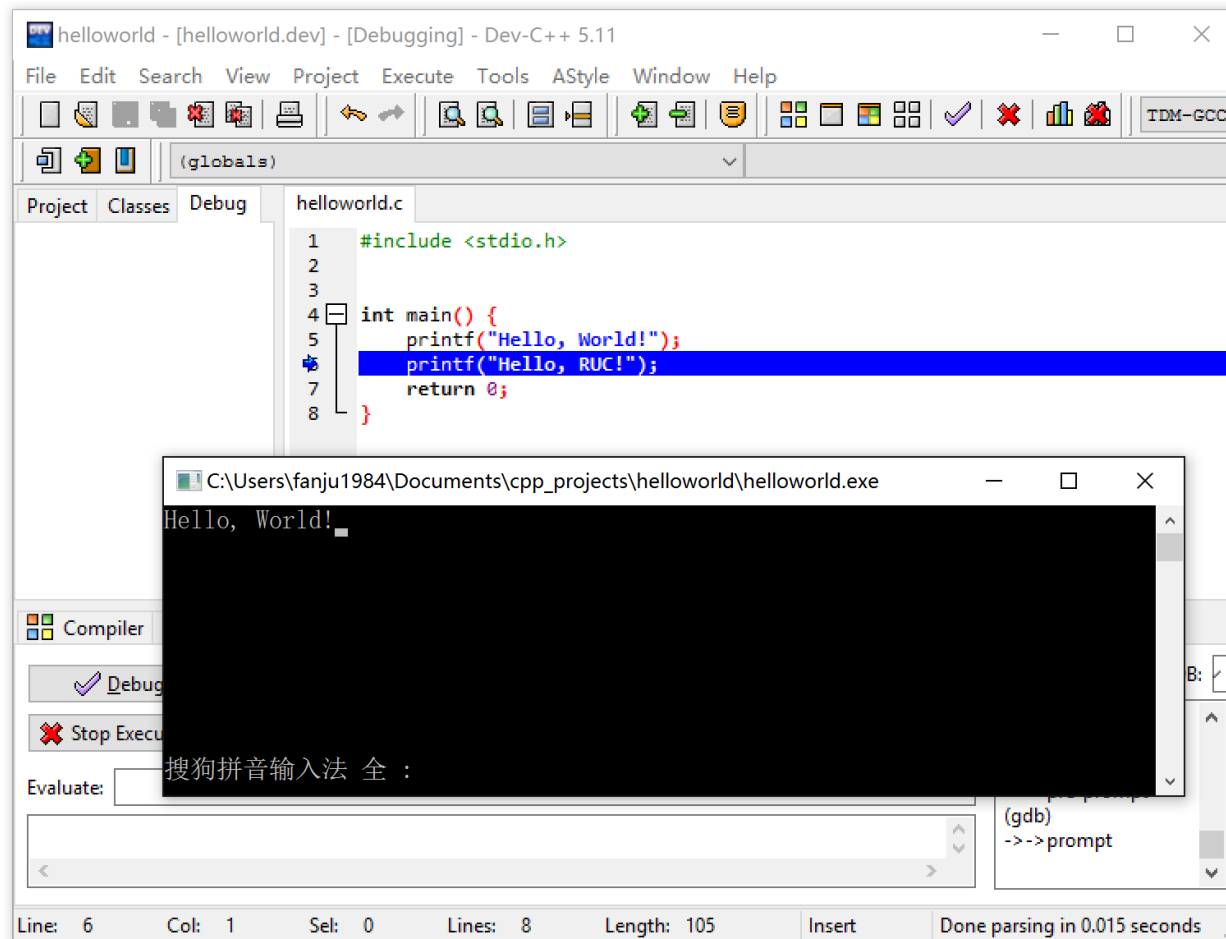
案例：Dev C++使用 (5)

■ 编译 (Compile) & 运行 (Run)



案例：Dev C++使用 (6)

■ 调试 (Debug) – 设置断点 (Breakpoint)



Mac: VSCode

<https://code.visualstudio.com/Download>



↓ Windows

Windows 7, 8, 10

User Installer	64 bit	32 bit	ARM
System Installer	64 bit	32 bit	ARM
.zip	64 bit	32 bit	ARM



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, SUSE

.deb	64 bit	ARM	ARM 64
.rpm	64 bit	ARM	ARM 64
.tar.gz	64 bit	ARM	ARM 64

Snap Store



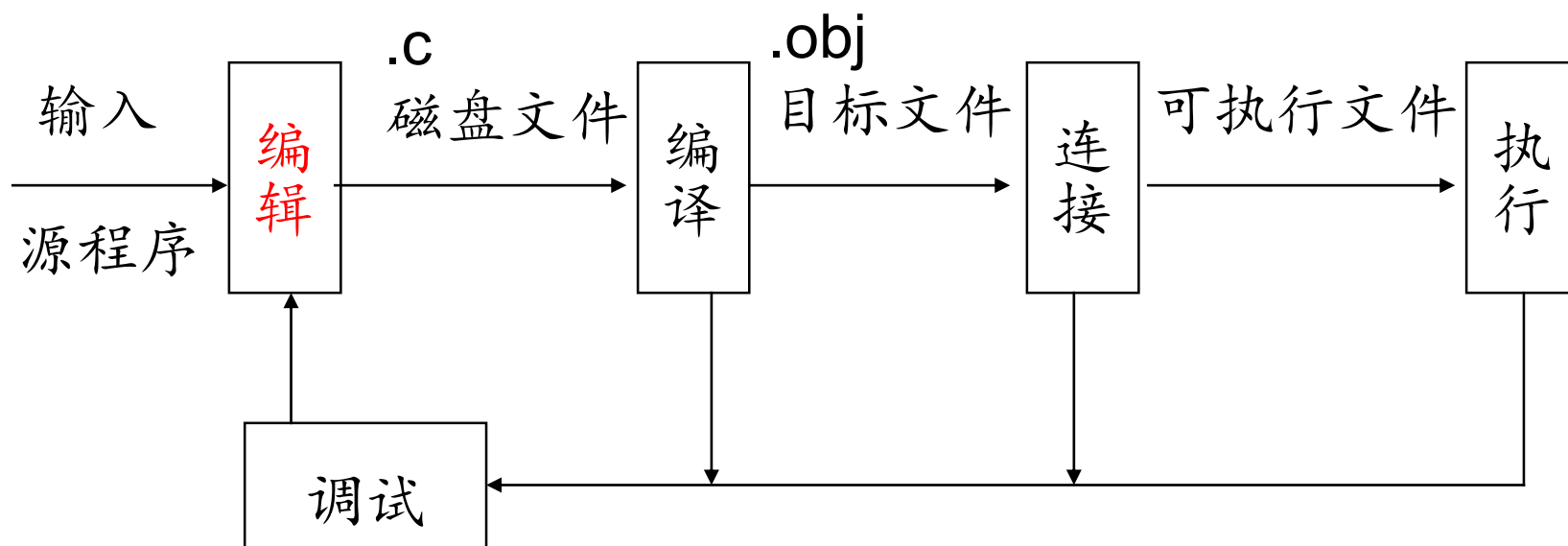
↓ Mac

macOS 10.11+

.zip Universal Intel Chip Apple Silicon

思考：可以不用IDE吗？

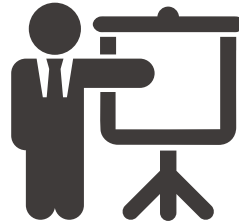
- 原则上讲，一个文本编辑器足矣



- 这种方式适合有一定经验的程序员



中國人民大學
RENMIN UNIVERSITY OF CHINA

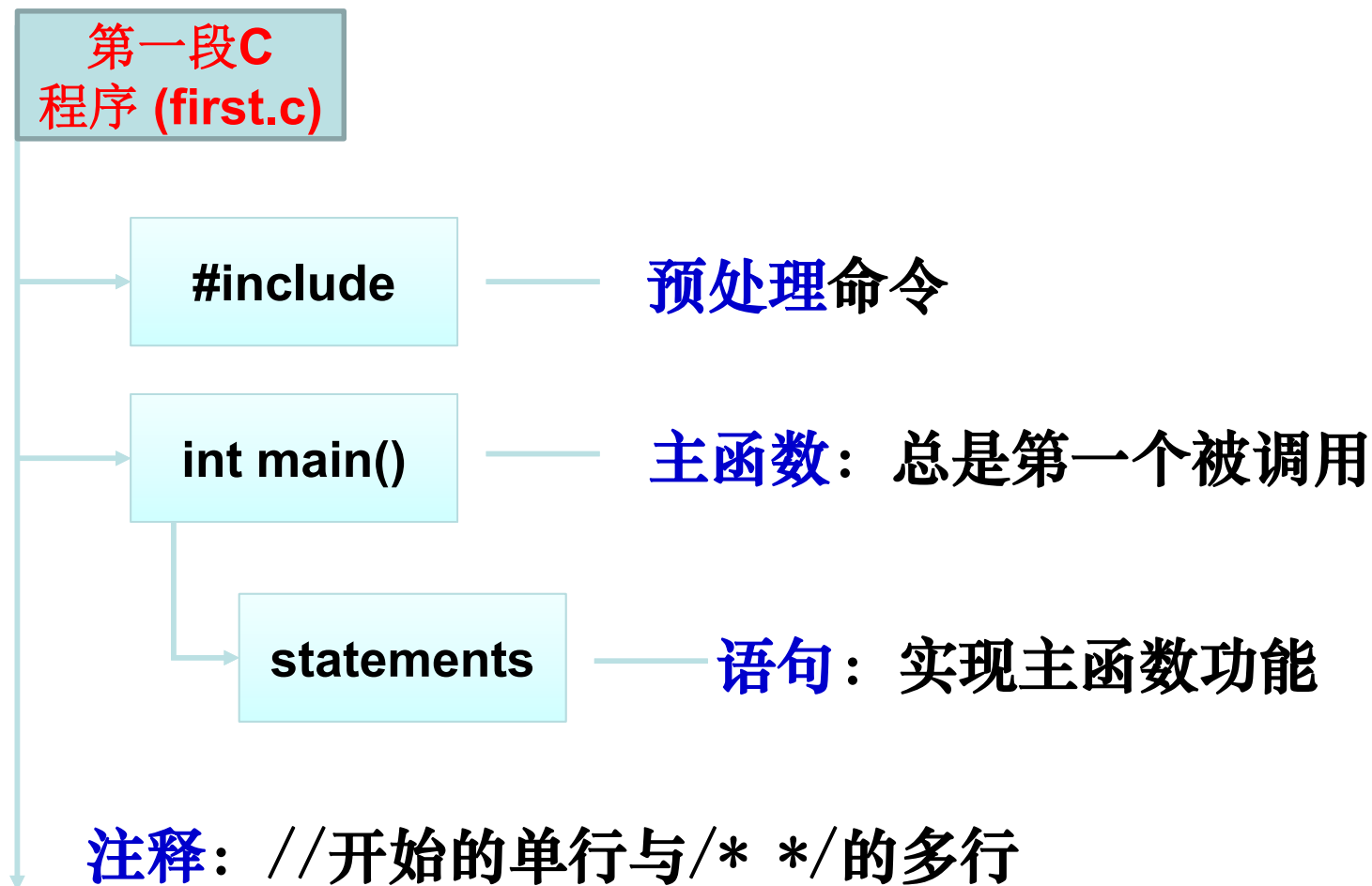


03. 程序基本结构

我们的第一段C程序

```
#include <stdio.h>
int main() /* a simple program */
{
    int num; /* define a variable called num */
    num = 1; // assign a value to the variable
    printf("I am a simple "); // use the printf() function
    printf("computer.\n");
    printf("My favorite number is %d because it is first.\n",num);
    return 0;
}
```

第一遍：分析代码结构



第二遍：分析基本功能 (1)

- `#include <stdio.h>` - 包含另一个文件
 - 基本功能: `stdio.h`, 系统头文件
 - C风格: `# include <stdio.h>`
 - C++风格:
 - `#include <iostream>`
 - `using namespace std;`
- `int main()` – 主函数头
 - 从`main()`函数的第一条语句开始, 直到`main()`运行结束为止
- `{ ... }` - 主函数体开始和结束

第二遍：分析基本功能 (2)

- `int num;` - 声明语句

- 基本功能：告诉系统你现在要使用一个变量，这个变量是个int（整数）类型
- 语句的最后使用分号表示结束

- `num = 1;` - 赋值语句

- 基本功能：将变量num的值设定为1

- `printf("I am a simple ");` - 函数调用语句

- 基本功能：调用printf函数，printf是标准C库函数

第二遍：分析基本功能 (3)

- `printf("computer.\n");` - 函数调用语句
 - 字符' `\n`' 告诉系统另起一新行
- `printf("My favorite number is %d because it is first.\n" , num);` - 函数调用语句
 - 将数字嵌入到文本段落中打印出来
 - 字符`%d`表示了打印出`num`的格式
- `return 0;` - 返回语句
 - 函数返回一个数字给调用方

第二遍：分析基本功能 (3)

- 注释： 注释只是给人看的,对编译和运行不起作用。

➤ 轻松一下：

- 我: 嗨//是什么意思
- 程序员: 嗨
- 我: 我问你//是什么意思
- 程序员: 问吧
- 我: 我问//是什么意思
- 程序员: 你TM倒是问啊
- 我: F\$\$K...

演示环节

第一个first.c程序

我们的第二段C程序

```
#include <stdio.h>

int main() /* a simple program          */
{

    int num1 = 10, num2 = 5; /* declare two variables */
    int sum, max; // declare two variables
    num2 = 15; // assign a value to the variable
    /* Compute the summation */
    sum = num1 + num2;
    printf("Sum of the two numbers: %d\n", sum);
    /* Compute the maximum value */
    if (num1 >= num2)
    {
        max = num1;
    } else {
        max = num2;
    }
    printf("Max of the two numbers: %d", max);
    return 0;
}
```

分析新的基本功能

- `int num1 = 10, num2 = 5;` - 声明且初始化 (多个) 变量
- `sum = num1 + num2;` - 算术操作
 - 试试其它的算术操作
- `if (num1 >= num2) { ... } else { ... }` - 选择控制语句
 - 不同情况需要不同处理时应使用
 - 语句 `num1 >= num2` 是判断条件

演示环节

第二个second. c程序

练习题

- 写出下列语句的输出是什么

```
printf("Baa Baa Black Sheep.");
```

```
printf("Have you any wool?\n");
```

```
printf("Begone!\nO creature of lard!");
```

```
printf("What?\nNo/nBonzo?\n");
```

```
int num;
```

```
num = 2;
```

```
printf("%d + %d = %d", num, num, num + num);
```

程序调试 (Debug)

Photo # NH 96566-KN (Color) First Computer "Bug", 1947

92

9/9

0800 Andam started

1000 " stopped - andam ✓

1300 (032) MP-MC 1.58264000 1.58264000 9.037 847 025

(033) PRO 2 2.130476415 2.130476415 9.037 846 995 correct

concord 2.130676415 4.615925059(-2)

Relays 6-2 in 033 failed special speed test

in relay 11.000 test.

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Multi-Adder Test.

1545

Relay #70 Panel F

(moth) in relay.

First actual case of bug being found.

1630 Andam started.

1700 closed down.

Relay 2145

Relay 9370

Grace Murray Hopper

程序调试 (Debug)

■ 语法错误

- 纠错: `int num1 = 1, int num2 = 2;`
- 可以被编译器发现

■ 语义错误

- 纠错: `area = pi * r * r * r`
- 不能被编译器发现

■ 怎么进行调试

- 逐行扫描定位
- 排除不可能出现错误的地方, 等等

练习题

- 程序在第7行、第8行和第9行后是什么状态（a和b的取值如何）？

```
#include <stdio.h>
int main(void) {
    int a, b;
    a = 5;
    b=2; /*line7*/
    //printf("%d %d\n", a, b)
    b=a; /*line8*/
    //printf("%d %d\n", a, b)
    a=b; /*line9*/
    printf("%d %d\n", a, b);
    return 0;
}
```