

# C/C++ 程序设计基础

魏恒峰

hfwei@nju.edu.cn

2017 年 10 月 20 日

# C/C++ 程序设计基础

## 1 课程简介

## 2 IO

警告：不要在周五的晚上写代码！



# 辅导课程介绍: C or C++

## C 语言是基础:

- ▶ IO
- ▶ Control Flow
- ▶ Function
- ▶ Array (String) & Pointers
- ▶ Struct

## C++ 语言的特性:

- ▶ OO (Object-oriented)
- ▶ Templates
- ▶ STL (Standard Template Library)
- ▶ FP (Functional Programming)

先学习 C 语言; 是否学习 C++ 视情况而定。

# 辅导课程介绍：课程形式

**讲解** 语言知识点、常见的“坑”（点到为止）

**练习** 指定 OJ 题目：互助练习、从旁辅导

**分享** 得意之处、Debug 之痛

# 辅导课程介绍：课程形式

**讲解** 语言知识点、常见的“坑”（点到为止）

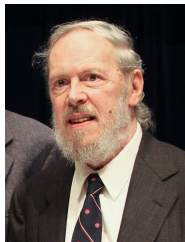
**练习** 指定 OJ 题目：互助练习、从旁辅导

**分享** 得意之处、Debug 之痛

**课后** 继续完成 OJ 剩余题目

# C 语言的历史、标准与实现

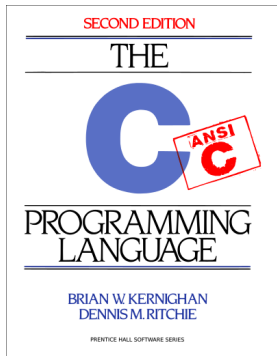
1972 年, Dennis Ritchie 创造了 C 语言:



K&R C (1978) → ANSI C/ISO C (C89/C90) → C99 → C11

GCC, Clang, Microsoft Visual C++, ...

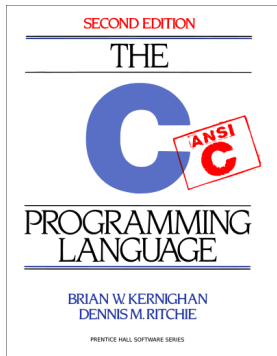
# 参考资料



Hello World!



# 参考资料



Hello World!

<http://www.cplusplus.com/>

<http://en.cppreference.com/w/>

# 听说你还没有注册过 ...



# C/C++ 程序设计基础

1 课程简介

2 IO

# IO 基本概念

IO 模型：字符流 (character stream)

- ▶ a sequence of characters divided into lines
- ▶ each line consists of  $\geq 0$  characters followed by a newline character

# Standard Input and Output: 以 getchar, putchar 为例

代码示例: `file-copy.c`

# Standard Input and Output: 以 getchar, putchar 为例

代码示例: `file-copy.c`

EOF (文件结束符): 由宏定义的整型数

- ▶ EOF 是整型数 (`int c; != EOF`)
- ▶ EOF 不是实际字符, 不同于行结束符 “\n” (`== '\n'`)
- ▶ EOF 不仅仅指示“文件”的结束, 而是指示字符流的结束

# Standard Input and Output: 以 getchar, putchar 为例

代码示例: `file-copy.c`

EOF (文件结束符): 由宏定义的整型数

- ▶ EOF 是整型数 (`int c; != EOF`)
- ▶ EOF 不是实际字符, 不同于行结束符 “\n” (`== '\n'`)
- ▶ EOF 不仅仅指示 “文件” 的结束, 而是指示字符流的结束

代码运行: `./file-copy`

# Standard Input and Output: 以 getchar, putchar 为例

代码示例: file-copy.c

EOF (文件结束符): 由宏定义的整型数

- ▶ EOF 是整型数 (`int c; != EOF`)
- ▶ EOF 不是实际字符, 不同于行结束符 “\n” (`== '\n'`)
- ▶ EOF 不仅仅指示“文件”的结束, 而是指示字符流的结束

代码运行: ./file-copy

Windows Ctrl + Z

Linux Ctrl + D

Mac Ctrl + D

请仔细观察并思考:

1. Enter 的作用
2. EOF 的作用



# IO on OJ: input redirection (输入重定向)

代码运行: `./file-copy <file-copy-file`

## 格式化输出: printf

```
int printf(const char* format, ...);
```

format: %[flags][width][.precision][length]specifier

## 格式化输出: printf

```
int printf(const char* format, ...);
```

format: %[flags][width][.precision][length]specifier

常用的输出格式:

Decimal integer %d, %ld

Decimal float %f, %.2f

Character %c

String %s, %.5s

代码示例: printf.c

使用 `printf('%s', s)`, 不要使用 `printf(s)`。

## 其它输出函数: fprintf, sprintf

```
int fprintf(FILE *stream, const char *format, ...);
```

```
int sprintf(char *str, const char *format, ...);
```

## 格式化输入: scanf

```
int scanf(const char *format, ...)
```

1. Read characters from the standard input
2. Interpret according to format
3. Store in arguments (pointers)

## 格式化输入: scanf

```
int scanf(const char *format, ...)
```

1. Read characters from the standard input
2. Interpret according to format
3. Store in arguments (pointers)

format:

- ▶ Whitespace character
- ▶ Non-whitespace character, except format specifier (%)
- ▶ Format specifiers: %[\*] [width] [length] specifier

# 格式化输入: scanf

代码示例: scanf.c

Return value:

**Success** # of items of the argument list successfully filled

- ▶  $\leq$  # of arguments

**Failure** EOF

- ▶ reading error (ferror), end-of-file (feof)

## 其它输入函数: fscanf, sscanf

```
int fscanf(FILE *stream, const char *format, ...);
```

```
int sscanf(const char *s, const char *format, ...);
```



# Line Input and Output

```
char *fgets(char *str, int num, FILE *stream);
```

```
char *gets(char *str);
```

```
int puts(const char *str);
```

# OJ 练习之 IO

## OJ 练习示例 (1.1.1)

# OJ 练习之 IO

OJ 练习示例 (1.1.1)

1.1.1 ~ 1.1.6      1.2.2

# OJ 常见输入模式

IO 练习之总结与分享

## Problem 1.1.1

**Problem** Your task is to Calculate  $a + b$ .

**Input** The input will consist of a series of pairs of integers  $a$  and  $b$ , separated by a space, one pair of integers per line.

**Sample Input**

1	5
10	20

**Output** For each pair of input integers  $a$  and  $b$  you should output the sum of  $a$  and  $b$  in one line, and with one line of output for each line in input.

**Sample Output**

6
30