

# Introduction to C world

王慧妍

[why@nju.edu.cn](mailto:why@nju.edu.cn)

南京大学



软件学院



计算机软件研究所



# 本讲概述

---

- 计算机世界和编程语言
- 为什么要学习C语言
- 课程设计和安排
- 选讲示例

# To CS beginners,

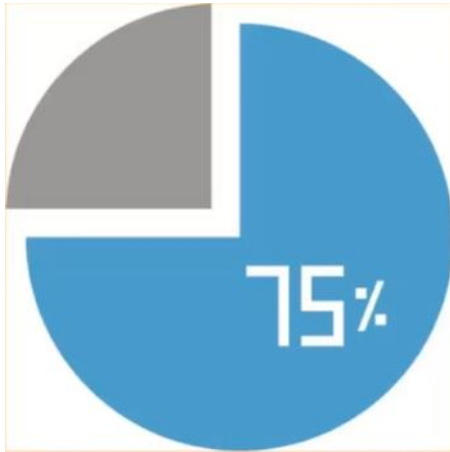
- Don't panic!

**DON'T  
PANIC!**

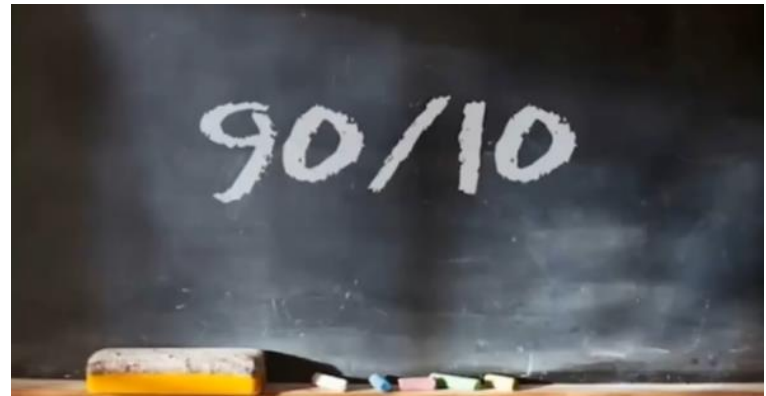


# Questionnaire from the past

---



75% of students are new to programming.



10% of students attended in some programming contests.

# Learn to program, not (only) learn C

---

- Programming
  - C is a computer *programming language* (PL).
  - You **express** your **ideas** to computers via PLs.

# 计算机怎么做事

- “请给我一杯水。”
  - 转身走进厨房
  - 找到一个杯子
  - 找到一个水壶
  - 在杯子里倒入一些水
  - 拿着杯子走向餐桌

和计算机交谈？

告诉计算机如何做！

程序是用特殊语言编写，表达描述要计算机做事的流程和方法

# 重复是计算机最会的事情

- “请帮我算一个数学题 $2x+6=20$ 。”

1? 2? 3? 4?

→while()/for()

5? 7?

→二分法

$$2x+6=20$$

$$2x = 14$$

$$x = 7$$

# 怎么告诉计算机要做什么？



知乎 @汇智动力IT学院



# 有点庆幸，我们不是几十年前的程序员

- Hello.c演示

```
1  #include <stdio.h>
2
3  int main(){
4      printf("Hello world!\n");
5  }
```

```
0000000000401550 <main>:
401550:    55                push    %rbp
401551:    48 89 e5          mov     %rsp,%rbp
401554:    48 83 ec 20       sub     $0x20,%rsp
401558:    e8 d3 00 00 00    callq  401630 <__main>
40155d:    48 8d 0d 9c 2a 00 lea     0x2a9c(%rip),%rcx
401564:    e8 f7 14 00 00    callq  402a60 <puts>
401569:    b8 00 00 00 00    mov     $0x0,%eax
40156e:    48 83 c4 20       add     $0x20,%rsp
401572:    5d                pop     %rbp
401573:    c3                retq
```

```
00000000: 4d5a 9000 0300 0000 0400 0000 ffff 0000  MZ.....
00000010: b800 0000 0000 0000 4000 0000 0000 0000  ....@.....
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ....
00000030: 0000 0000 0000 0000 0000 0000 8000 0000  ....
00000040: 0e1f ba0e 00b4 09cd 21b8 014c cd21 5468  ....!..L.!Th
00000050: 6973 2070 726f 6772 616d 2063 616e 6e6f  is program canno
00000060: 7420 6265 2072 756e 2069 6e20 444f 5320  t be run in DOS
00000070: 6d6f 6465 2e0d 0d0a 2400 0000 0000 0000  mode....$.
00000080: 5045 0000 6486 0f00 9277 1063 0068 0000  PE..d...w.c.h..
00000090: 9304 0000 f000 2700 0b02 021e 001e 0000  ....'
000000a0: 0038 0000 000a 0000 e014 0000 0010 0000  .8.....
000000b0: 0000 4000 0000 0000 0010 0000 0002 0000  ..@.....
000000c0: 0400 0000 0000 0000 0500 0200 0000 0000  ....
000000d0: 0020 0100 0004 0000 ea05 0100 0300 0000  ....
000000e0: 0000 2000 0000 0000 0010 0000 0000 0000  ..
000000f0: 0000 1000 0000 0000 0010 0000 0000 0000  .....
00000100: 0000 0000 1000 0000 0000 0000 0000 0000  .....
00000110: 0080 0000 6c07 0000 0000 0000 0000 0000  ....1.....
00000120: 0050 0000 7002 0000 0000 0000 0000 0000  .P..p.....
00000130: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000140: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000150: 4040 0000 2800 0000 0000 0000 0000 0000  @@..(.....
00000160: 0000 0000 0000 0000 d481 0000 9801 0000  .....
00000170: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000180: 0000 0000 0000 0000 2e74 6578 7400 0000  ....text...
00000190: c81c 0000 0010 0000 001e 0000 0004 0000  .....
000001a0: 0000 0000 0000 0000 0000 0000 6000 5060  .....`P`
000001b0: 2e64 6174 6100 0000 d000 0000 0030 0000  .data.....0..
000001c0: 0002 0000 0022 0000 0000 0000 0000 0000  ...."
000001d0: 0000 0000 4000 50c0 2e72 6461 7461 0000  ....@.P..rdata..
000001e0: d004 0000 0040 0000 0006 0000 0024 0000  ....@.....$.
000001f0: 0000 0000 0000 0000 0000 0000 4000 6040  ....@.....@
00000200: 2e70 6461 7461 0000 7002 0000 0050 0000  .pdata..p....P..
00000210: 0004 0000 002a 0000 0000 0000 0000 0000  ....*
00000220: 0000 0000 4000 3040 2e78 6461 7461 0000  ....@.0@.xdata..
00000230: f401 0000 0060 0000 0002 0000 002e 0000  ....`.....
00000240: 0000 0000 0000 0000 0000 0000 4000 3040  ....@.....@
00000250: 2e62 7373 0000 0000 8009 0000 0070 0000  .bss.....p..
00000260: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000270: 0000 0000 8000 60c0 2e69 6461 7461 0000  ....`..idata..
```

# 程序的执行

- 机器语言，汇编语言，高级程序语言
- 解释
  - 借助于一个程序，那个程序可以理解你的程序，按要求执行
- 编译
  - 借助于一个程序，就像一个翻译，将你的程序翻译成机器能懂的语言，然后机器语言的程序就能够直接执行

```
C 0hello.c > ...  
1  #include <stdio.h>  
2  
3  int main(){  
4      printf("Hello world!\n");  
5      return 0;  
6  }
```

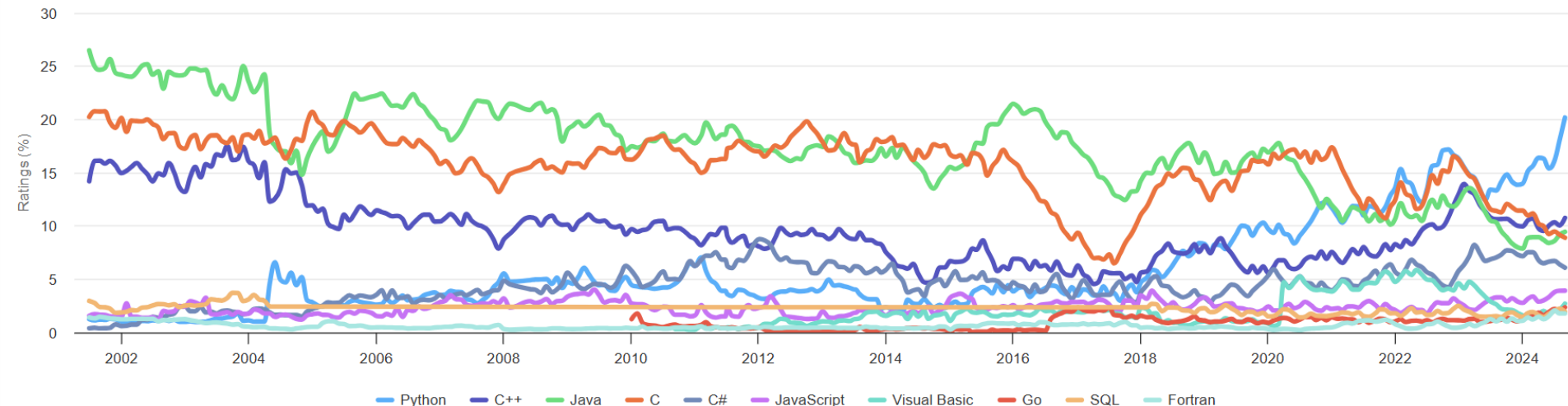






















# Why choose C?

Sep 2024	Sep 2023	Change	Programming Language		Ratings	Change
1	1			Python	20.17%	+6.01%
2	3	▲		C++	10.75%	+0.09%
3	4	▲		Java	9.45%	-0.04%
4	2	▼		C	8.89%	-2.38%
5	5			C#	6.08%	-1.22%

TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)



Sep 2024	Sep 2023	Change	Programming Language		Ratings	Change
1	1			Python	20.17%	+6.01%
2	3	▲		C++	10.75%	+0.09%
3	4	▲		Java	9.45%	-0.04%
4	2	▼		C	8.89%	-2.38%
5	5			C#	6.08%	-1.22%
6	6			JavaScript	3.92%	+0.62%
7	7			Visual Basic	2.70%	+0.48%
8	12	▲▲		Go	2.35%	+1.16%
9	10	▲		SQL	1.94%	+0.50%
10	11	▲		Fortran	1.78%	+0.49%
11	15	▲▲		Delphi/Object Pascal	1.77%	+0.75%
12	13	▲		MATLAB	1.47%	+0.28%
13	8	▼▼		PHP	1.46%	-0.09%
14	17	▲		Rust	1.32%	+0.35%
15	18	▲		R	1.20%	+0.23%
16	19	▲		Ruby	1.13%	+0.18%
17	14	▼		Scratch	1.11%	+0.03%
18	20	▲		Kotlin	1.10%	+0.20%
19	21	▲		COBOL	1.09%	+0.22%
20	16	▼▼		Swift	1.08%	+0.09%

# Why choose C?

```
J hello.java > hello > main(String[])  
1 public class hello {  
    Run | Debug  
2     public static void main(String[] args) {  
3         System.out.println(x: "Hello world!\n");  
4     }  
5 }  
6
```

```
hello.py  
1 print("Hello world!\n")
```

```
hello.cpp > main()  
1 #include <iostream>  
2 using namespace std;  
3  
4 int main()  
5 {  
6     cout << "Hello, World!";  
7     return 0;  
8 }
```

```
C 0hello.c > ...  
1 #include <stdio.h>  
2  
3 int main(){  
4     printf("Hello world!\n");  
5     return 0;  
6 }
```

```
main.cj ×  
src > main.cj  
1 main(): Int64 {  
2     println("hello world")  
3     return 0  
4 }
```

```
hello.cs  
1 using System;  
2 using System.Collections.Generic;  
3 using System.Linq;  
4 using System.Text;  
5 using System.Threading.Tasks;  
6 namespace HelloWorld  
7 {  
8     class Program  
9     {  
10         static void Main(string[] args)  
11         {  
12             Console.WriteLine("Hello world!\n");  
13             Console.ReadKey();  
14         }  
15     }  
16 }
```

# "It was the summer of '69"



## PDP-7

UNICS->UNIX

- 需要一个爱玩游戏的父亲？



## PDP-11

Thompson  
Ritchie

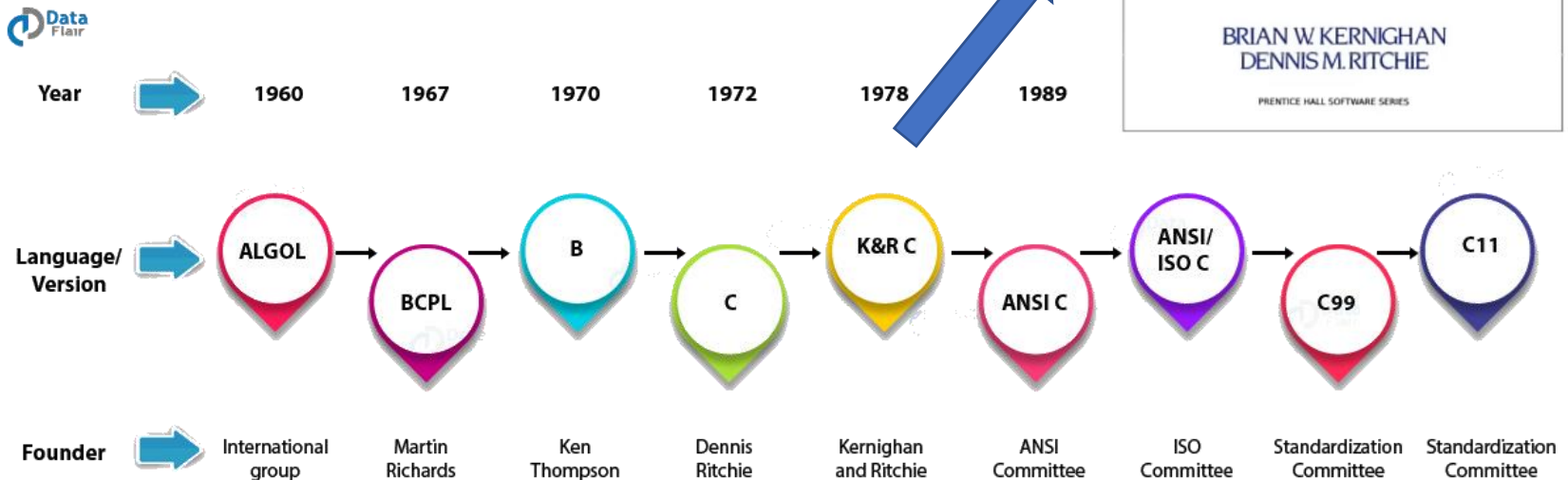
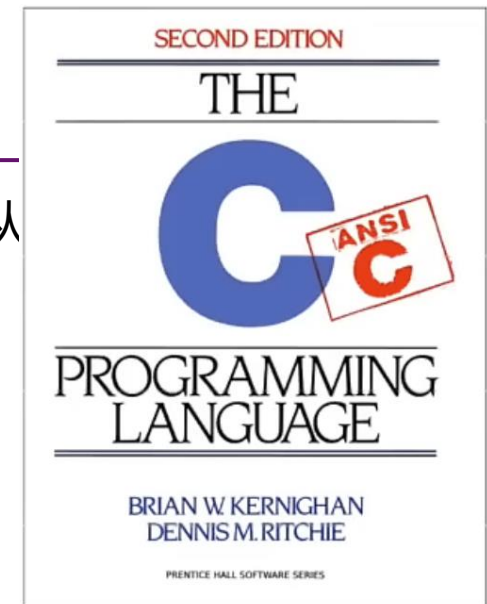
# A long history...

- C语言从B语言发展来，B语言是从BCPL发展来，BCPL从FORTRAN发展来
- BCPL和B都支持指针间接的方式，所以C也支持
- C语言还受到PL/1的影响，贴近表达底层的机器指令

语言	年份	开发者
Algol	1960	International Group
BCPL	1967	Martin Richard
B	1970	Ken Thompson
Traditional C	1972	Dennis Ritchie
K & R C	1978	Kernighan & Dennis Ritchie
ANSI C	1989	ANSI Committee
ANSI/ISO C	1990	ISO Committee
C99	1999	Standardization Committee

# A long history...

- C语言从B语言发展来，B语言是从BCPL发展来，BCPL从
- BCPL和B都支持指针间接的方式，所以C也支持
- C语言还受到PL/1的影响，贴近表达底层的机器指令



- C
  - 1973年3月，第三版unix出现C的编译器
  - 1973年11月，第四版unix完全由C语言编写
- C17; C23 (supported from gcc 14.1.0)



# C语言

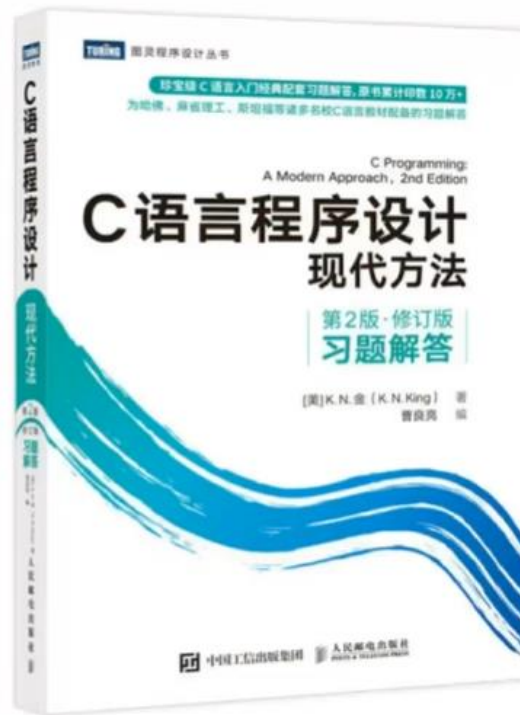
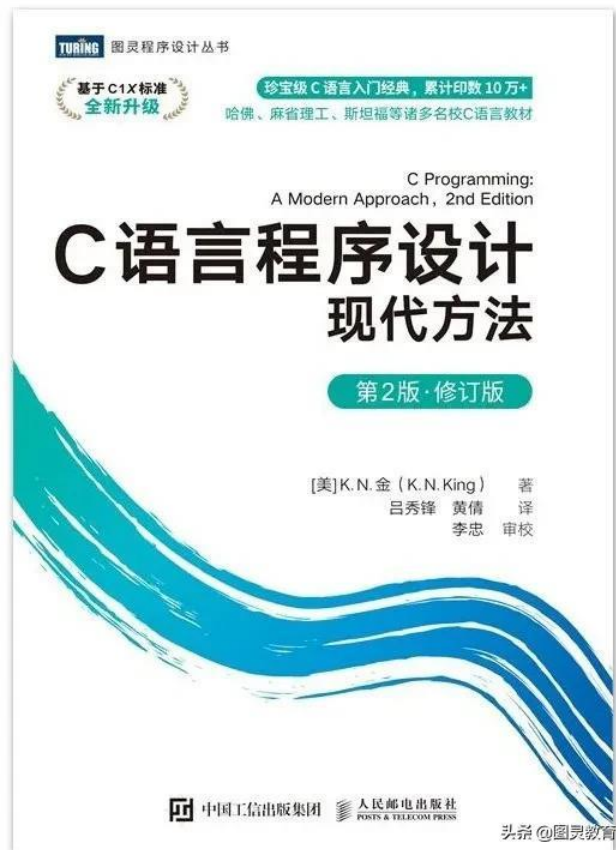
---

- 早期就有的关键特性
  - 结构化编程：支持使用括号、if-else 语句和循环结构
  - 指针：允许对内存位置直接访问
  - 数据类型：定义了诸如 int 和 char 等基本数据类型
  - 编译器：将 C 程序翻译成机器代码
  - 可移植性：编写 C 程序可以很容易地移植到不同的计算机平台
- 最接近底层的高级语言
- 基础领域
  - 操作系统
  - 嵌入式系统
  - 驱动程序

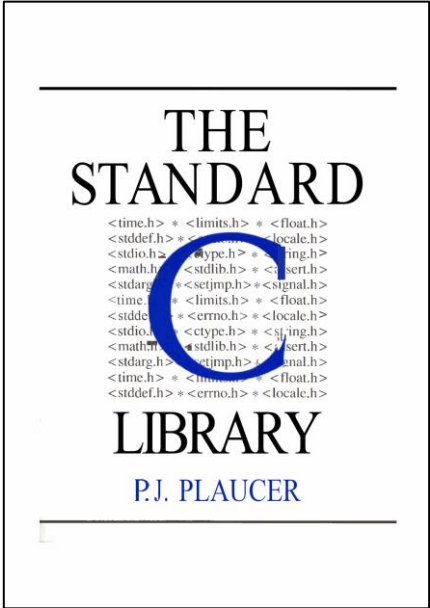
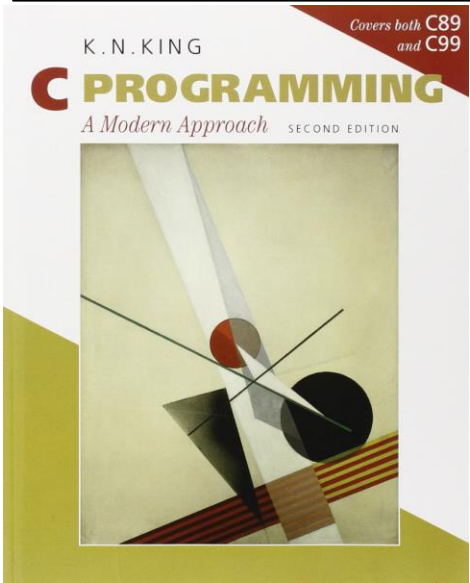
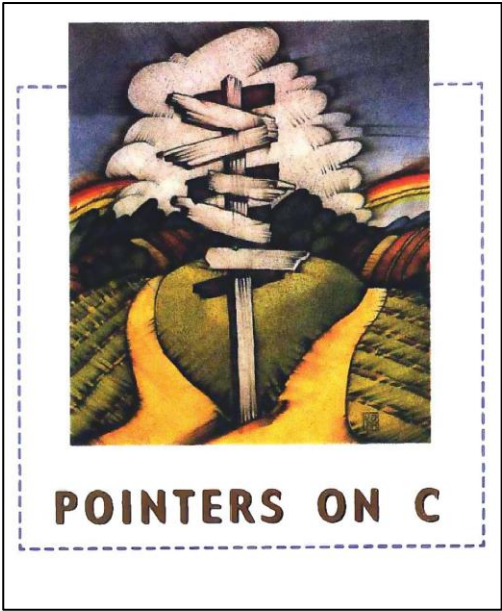
# 关于本课程

<http://www.why.ink:8080/CPL/2024/>

# 课程教材



# 辅助读物



# 老师/助教的使用

- 联系/求助：使用邮件
  - 能使你更好地整理问题。也许在整理问题的过程中你就发现答案了
  - 老师：
    - 王慧妍： [why@nju.edu.cn](mailto:why@nju.edu.cn);
    - Office hour：费楼928，周五下午1-3点
  - 助教：
    - 待定
- Ask
  - 对上课内容没有理解的地方、对课程/学习的疑惑
  - 其他
- Do not ask
  - 安装XXX错了怎么办？程序哪里有bug？
  - 也许助教答疑时间可以有机会，用好助教

# 考核方式

任务		分数占比
平时OJ练习	10%	每周一次
阶段机试	15+20%	学期两次
课程项目	25%	学期一次
期末机试	30%	限时命题



# 重要网站

---

- 本班课程网站:
  - <http://www.why.ink:8080/CPL/2024/>
- CPL-Docs@NJUSE
  - docs.cpl.icu
- OJ系统
  - [dotOJ \(cpl.icu\)](http://dotOJ.cpl.icu)
    - http://oj.cpl.icu/

# 学术诚信 (Academic Integrity)



# 学生手册：不能抄作业

- What is academic integrity?
  - [What is Academic Integrity? | Academic Integrity at MIT](#)

Plagiarism	Cheating		
<b>Do:</b>	<b>Do:</b>	<b>Don't:</b>	
Trust the	Demonstrate your own achievement.	Don't copy answers from another student; don't ask another student to do your work for you. Don't fabricate results. Don't use electronic or other devices during exams.	e a paper
Undertake work.	Accept corrections from the instructor as part of the learning process.	Don't alter graded exams and submit them for re-grading.	out citing
	Do original work for each class.	Don't submit projects or papers that have been done for a previous class.	
Unauthorized	Facilitating Academic Dishonesty		
<b>Do:</b>	<b>Do:</b>	<b>Don't:</b>	
Trust the	Showcase your own abilities.	Don't allow another student to copy your answers on assignments or exams. Don't take an exam or complete an assignment for another student.	d the

# 学生手册：不能抄作业

---

- What is academic integrity?
  - 简单概况：独立完成
- 针对作业的独立完成
  - 自己完成作业
  - 对使用的已有资料作出明确的标示
  - [ACM Policy on Plagiarism, Misrepresentation, and Falsification](#)
- 针对代码的独立完成
  - 自己完成代码的编写
  - 自己完成测试用例
  - 在允许的范围内使用他人的成果

# 知乎贴：作业抄袭中的人生百态



农夫山泉  
NONGFU SPRING

我们不生产~~水~~,我们只是~~大自然的搬运工~~

代码

互联网



代码抄袭：那些让985学

```
1 void P1() {
2   ...
3   puts("Game_Over");
4   ...
5 }
6 void P2() {
7   ...
8   puts("G"); puts("a")
9   puts("e"); puts("_")
10  puts("v"); puts("e")
11  ...
12 }
```

```
1 cur->lineno = temp->lineno;
2 strcpy(cur->type, type);
3 cur->isLexical = 0;
4 cur->children = temp;
```

```
1 $$->is_root=1;
2 $$->no_leaves=1;
3 $$->leaves[0]=(Node*)$1;
4 if(exit_error==0)
5   {print_tree($$,0);}
```

```
1 temp->line = a->line;
2 temp->lChild = a;
3 while(num > 1){
4   a->rChilds = va_arg(list,
5                       node*);
6   a = a->rChilds;
7   num--;
8 }
```

```
1 head->number_signal = 0;
2 head->line = temp->line;
3 strcpy(head->type,type);
4 head->child_left = temp;
```

```
1 $$->final=0;
2 $$->num_children=1;
3 $$->children=(Node**)malloc
4   (sizeof(Node*)*$$->
5    num_children);
6 $$->children[0]=(Node*)$1;
7 if(!wrong)
   printNode($$,0);
```

```
1 p_node->left_child = temp;
2 p_node->line = temp->line;
3 for(int i=0;i < num-1;++i){
4   temp->right_child =
5     va_arg(valist,struct
6             Node*);
7   temp = temp->right_child;
```

知乎 @蒋炎岩7

# 抄袭惩罚机制

- 本课程将对抄袭采取严厉的惩罚措施。
  - 对于平时编程练习，除了抄袭的当次作业分数清零外，每次抄袭从**总评**中扣除 5 分，至平时练习所占总评的 10 分扣完为止。
  - 对于期末项目，抄袭则分数清零，不额外扣总评分数。

## Individual work

- Problem sets in this class are intended to be primarily individual efforts. You are encouraged to discuss approaches with other students but **your code and your write-up must be your own.**
- You **may not use materials produced as course work by other students**, whether in this term or previous terms, **nor may you provide work for other students to use.**
- It's good to help other students. But as a general rule, during the time that you are helping another student, **your own solution should not be visible**, either to you or to them. Make a habit of closing your laptop while you're helping.

*Talk is cheap. Show me the code.*

——*Linus Torvalds*

开始愉快的码代码吧~



# 环境配置 Arm yourself

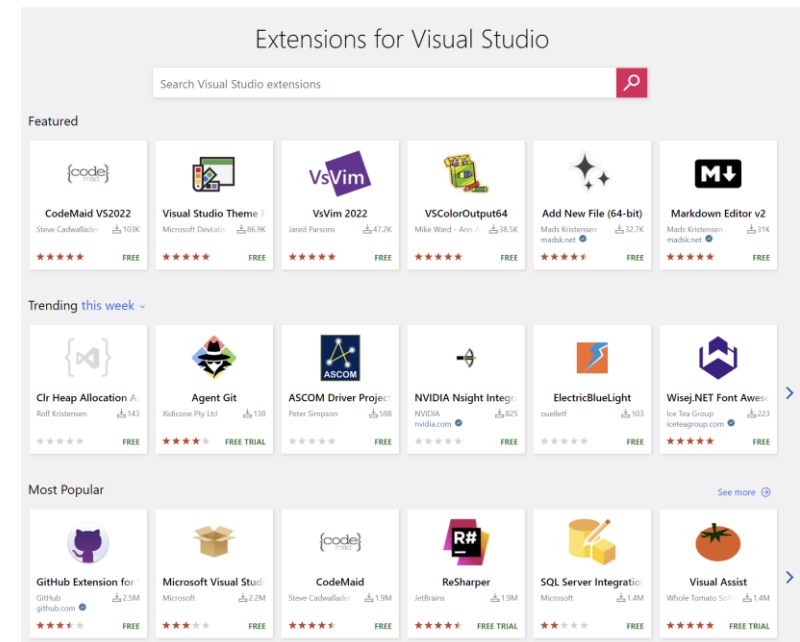
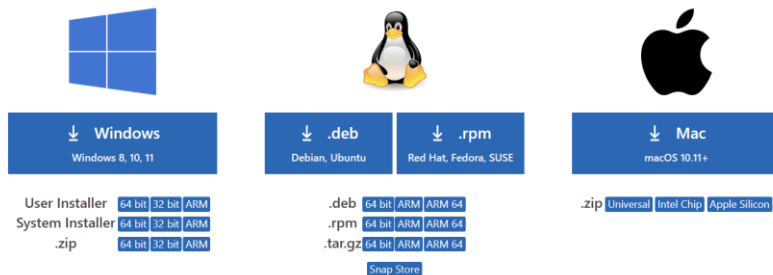
新手村任务：配置合适的C语言编程环境

- 遇到问题（如安装错误）找同学询问/解决 OK
  - 但你可能就失去了这门课原本的训练
    - 尽可能先自己解决
  - 帮其他同学解决问题的人
    - 一起还原解决问题的过程

<http://www.why.ink:8080/CPL/2024/>

# Coding on X

- Linux or Windows or MacOS?
- VSCode, Clion, Vim, Sublime?



# 愉快的编程之旅就这样开始了

- 一个例子的故事



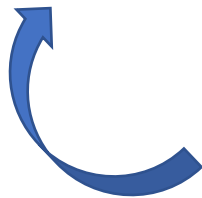
*Talk is cheap. Show me the code.*  
——Linus Torvalds



# 愉快的编程之旅就这样开始了

- 一个例子的故事

```
Hello world!  
12+22=34
```



```
#include <stdio.h>  
  
int main(){  
    printf("Hello world!\n");  
    printf("12+22=%d\n", 12+22);  
    return 0;  
}
```

*Talk is cheap. Show me the code.*  
——Linus Torvalds

# printf

```
int printf(const char *__restrict__ _Format, ...)
```

- <stdio.h>
- 显示格式串的程序
  - `printf("Hello world!\n");`
  - `printf("11+22=%d", 11+22);`
  - `printf("11.05+22.33=%.2f", 110.5+22.33);`
  - `printf("\\");`

# 转义字符

---

- 常见

- `\n`

- `\t`

- `\b`

- `\\`

- `\"`

- `\'`

- `...`

# 愉快的编程之旅就这样开始了

- 程序框架

```
#include <stdio.h>

int main(void){

    return 0;
}
```

← 在未来一段时间(函数前), 大部分你写的代码都在这里

核心准则：编写可读代码

Coding style

# Coding Style

Programs are meant to be read by humans and only incidentally for computers to execute. — D. E. Knuth

(程序首先是拿给人读的，其次才是被机器执行。)

## Google Style

### styleguide

#### Google Style Guides

Every major open-source project has its own style guide: a set of conventions (sometimes arbitrary) about how to write code for that project. It is much easier to understand a large codebase when all the code in it is in a consistent style.

"Style" covers a lot of ground, from "use camelCase for variable names" to "never use global variables" to "never use exceptions." This project ([google/styleguide](https://google.com/styleguide)) links to the style guidelines we use for Google code. If you are modifying a project that originated at Google, you may be pointed to this page to see the style guides that apply to that project.

This project holds the C++ Style Guide, C# Style Guide, Swift Style Guide, Objective-C Style Guide, Java Style Guide, Python Style Guide, R Style Guide, Shell Style Guide, HTML/CSS Style Guide, JavaScript Style Guide, TypeScript Style Guide, AngularJS Style Guide, Common Lisp Style Guide, and Vimscript Style Guide. This project also contains `cpplint`, a tool to assist with style guide compliance, and `google-c-style.el`, an Emacs settings file for Google style.

If your project requires that you create a new XML document format, the [XML Document Format Style Guide](#) may be helpful. In addition to actual style rules, it also contains advice on designing your own vs. adapting an existing format, on XML instance document formatting, and on elements vs. attributes.

The style guides in this project are licensed under the CC-BY 3.0 License, which encourages you to share these documents. See <https://creativecommons.org/licenses/by/3.0/> for more details.

## 华为C语言编程规范

### C语言编程指南 V1.0



# Coding Style

---

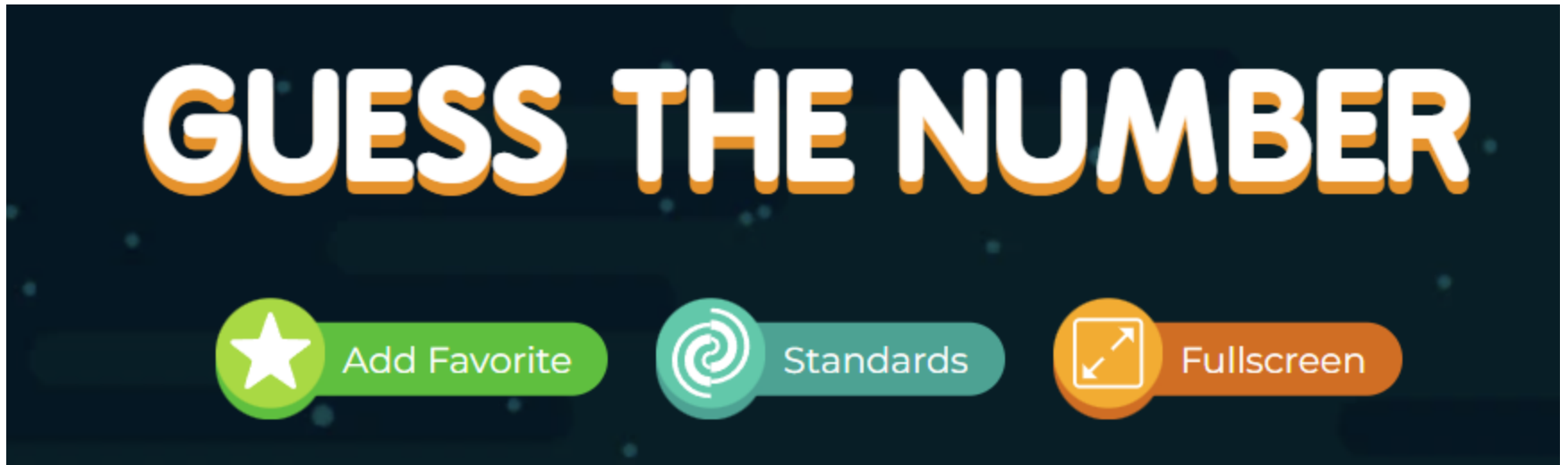


Braces

```
int foo (int Days) {  
    if ((Days % 10) == 3) {  
        return 1;  
    } else {  
        return 0;  
    }  
}
```

# More C programs

- [Guess the number](#)





# How to obtain a random number?

---

- [Pseudo-random number generation - cppreference.com](http://cppreference.com)

$$s \quad f(s) \quad f(f(s)) \quad \dots \dots$$

# More C programs

---

- [Guess.c](#)



# What is next?

---

- Variables and Types (int, double)
- Math: computer arithmetic
- Input/Output (I/O)
  
- Branching: "if/else", "switch/case"
- Looping: "while", "do/while", "for"
- Jumps: "break", "continue", "goto"
  
- Functions & Libraries

机器永远是对的 (and RTFM)

RTFM: Read The Friendly Manual

STFW: Search The Friendly Web

# 本次课程最重要的内容

---

- 静下心来，从头开始
  - [The Missing Semester of Your CS Education](#)
  - RTFM (slides), STFW, RTFSC
- 用好 Github的 “[Awesome](#)” 系列
  - 例如[The art of command line](#)
- 用好 Stack Overflow / Stack Exchange
- 禁用百度和中文关键字