

CSE/CS101: C++ 编程抽象思维

——不一样的编程课

薛浩

2023 年 3 月 14 日

www.stickmind.com

目录

1. StickMind 项目介绍
2. 斯坦福 CS 课程体系
3. CS101 课程介绍
4. C++ 语言介绍
5. Qt Creator 开发环境
6. CMake 构建工具

StickMind 项目介绍

www.stickmind.com

为中国基础工业软件之崛起作贡献!

本项目由武汉理工大学汪洋博士发起，旨在为传统工科领域学生提供编程入门的途径。

非计算机专业的同学在自学过程中常常会遇到很多问题，在一次次尝试一次次失败后，不仅对自己丧失了信心，更浪费了大把的时间。

环境搭建 学习过程中最劝退的就是开发环境问题多多，常常导致无法顺利完成学习

课程答疑 自学往往无法抓住课程的重点和精妙之处，简单的调试问题也会难倒很多人

作业批改 没人帮忙批改作业，不知道自己写的对不对、好不好，只能学个大概

目前规划中的课程如下，力求搭建一套完整的 CS 基础课程体系：

CS101: **C++ 编程抽象思维** 选用优秀的斯坦福 C++ 教学库，重点讲解了递归、经典的数据结构和算法

CS102: **计算机组成与系统** 使用 C 和汇编语言继续提升你的编程技术，带你揭开计算机架构的奥秘

CS103: **计算机系统原理** 学习操作系统、网络等系统的设计思想，并尝试应用到其他大型系统的设计中



CSE/CS101: C++ 编程抽象思维, Spring 2023

最新通知

- 新的一年, 祝大家学业有成, 工作顺利, 健康平安每一天。
- 一期于 2022 年 11 月中旬圆满结束, 累计报名人数 211 人, 广受学生好评。公开课最重要的部分并不在于视频而是作业! 复现课重要的是能够在老师的帮助下, 按照斯坦福的课程进度, 一步一个脚印完成每一份作业。只需要短短的 10 周, 你就能够实现一次蜕变, 真正意义上学会编程!
- 二期报名已开始, 如需报名请点击: [报名](#)

(a) cs101.stickmind.com



CSE/CS102: 计算机组成与系统, Spring 2023

最新通知

- 最新作业发布: [[作业 2: C Strings](#)]
- 最新实验练习: [[实验 2: C Strings](#)]
- 最新课程更新: 第 5 次课: More Strings [[Notes](#)] [[Video](#)] [[PDF](#)]
- 最新课程更新: 第 4 次课: C Chars and Strings [[Notes](#)] [[Video](#)] [[PDF](#)]
- 新的一年, 祝大家学业有成, 工作顺利, 健康平安每一天。
- 最新作业发布: [[作业 1: A Bit of Fun](#)]
- 最新实验练习: [[实验 1: Bits Bytes and Integers](#)]

(b) cs102.stickmind.com

不同于市面上培训类课程，本项目力求小而美，帮助一小部分真正希望学会编程做一些实事的朋友。

课程体系 完美复现斯坦福 CS 课程，让您得到原汁原味的学习体验

教学资源 完善的课程体系，详尽的教学文档，为您的学习体验保驾护航

教学服务 贴心的授课讲师提供及时的答疑，并对每份作业一对一代码评审

为什么学习计算机编程？

Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes or chemistry is about beakers and test tubes. Science is not about tools, it is about how we use them and what we find out when we do.

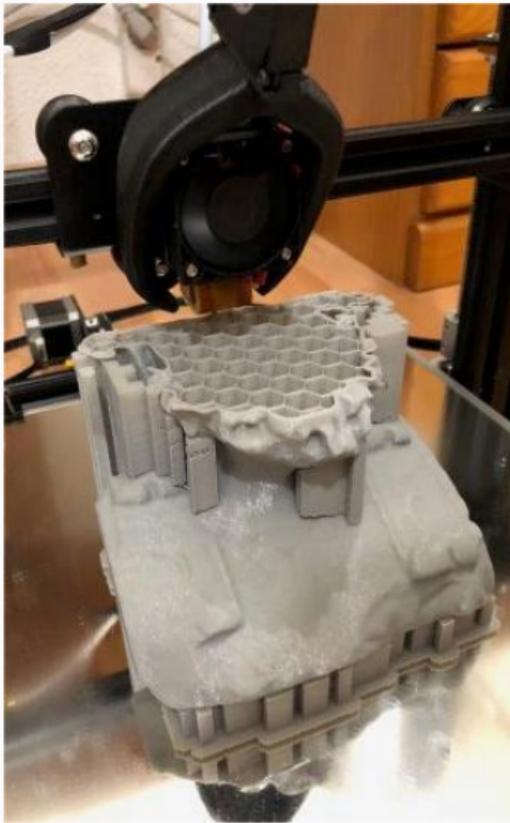
—Michael Fellows and Ian Parberry



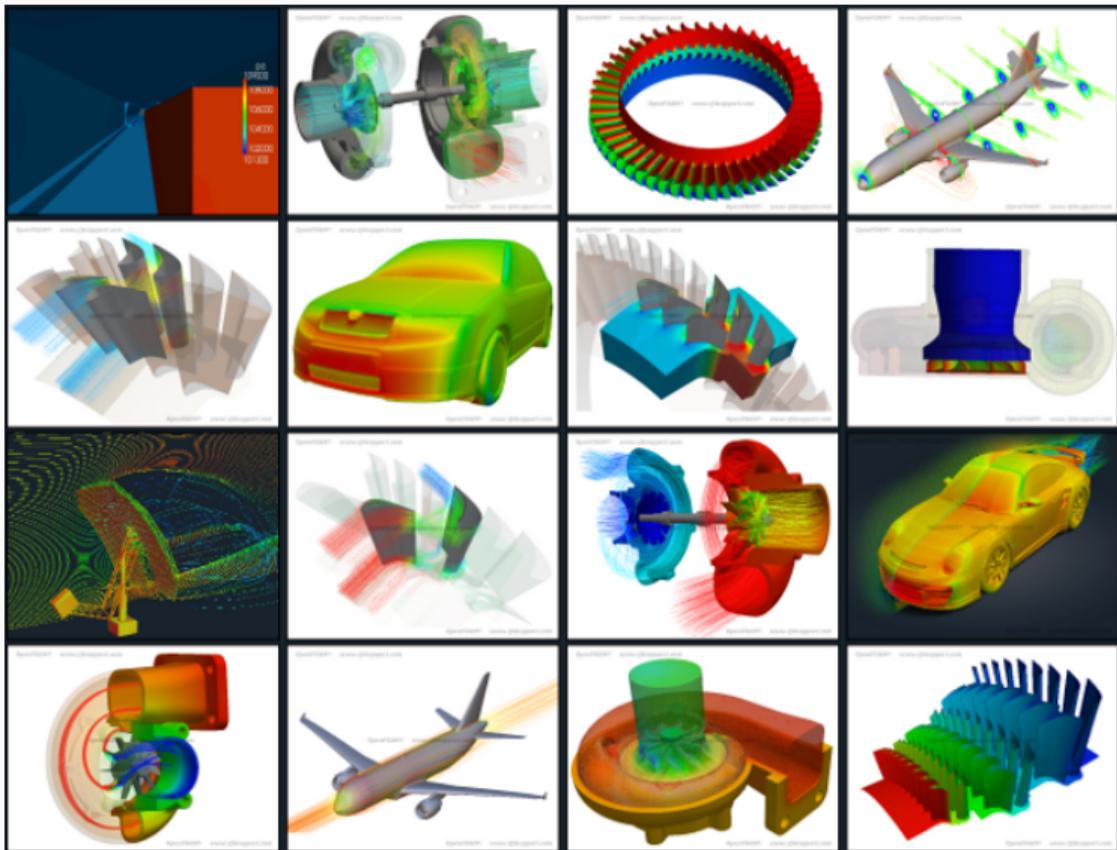
courseera













**所有这些项目都发生在
最近一二十年！**

斯坦福 CS 课程体系

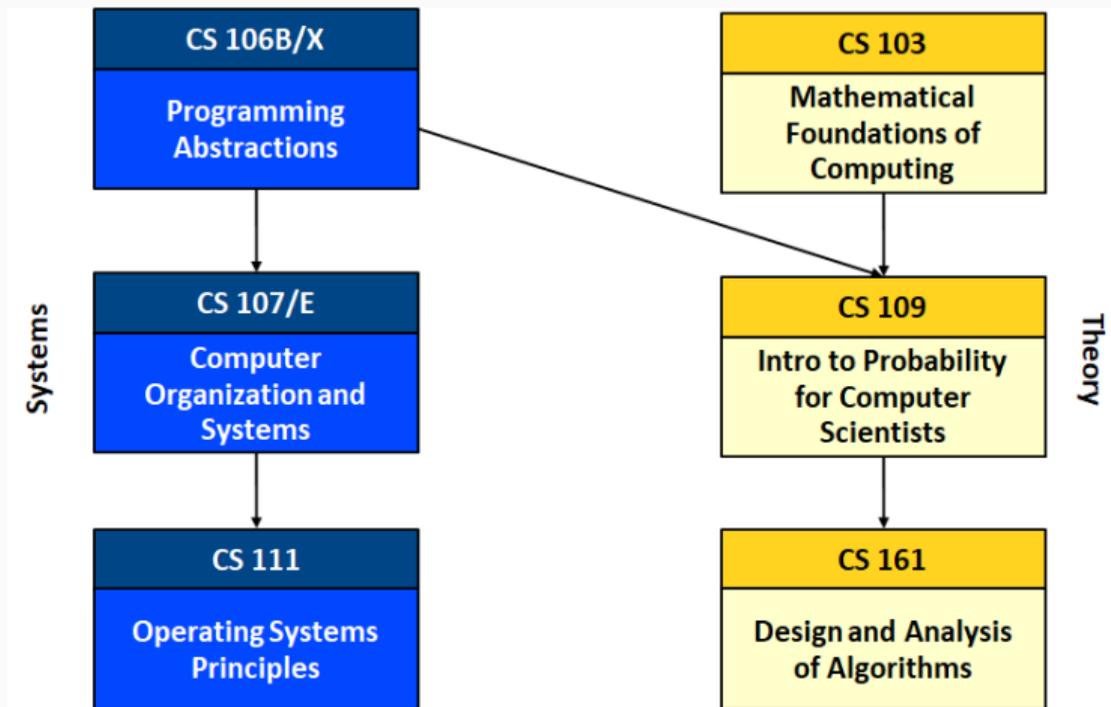


Figure 2: CS Minor [1]

CS101 课程介绍

学习如何利用计算（Computing）的力量来解决问题。

递归思想 递归是一个极其强大的技术，可以让你轻松地解决那些看起来无解的复杂问题。

数据抽象 整个课程会介绍多种抽象数据类型，在你编写程序过程中，这些抽象类型将发挥巨大作用。

算法分析 针对特定的问题，使用不同的算法在时间消耗方面会有巨大差异，本课程将学习如何评估算法效率以便开发出更高效的算法。

- 话题 1: 编程基础** 初学编程的新手，一般应该熟练使用函数和库处理字符串相关的编程任务。
- 话题 2: 抽象数据类型的使用** 在尝试实现抽象数据类型之前，应该先熟练使用这些工具解决问题。
- 话题 3: 递归和算法分析** 递归是一种强有力的思想，一旦掌握就可以解决很多看起来非常难的问题。
- 话题 4: 类和内存管理** 使用 C++ 实现数据抽象之前，应先学习 C++ 的内存机制。
- 话题 5: 常见数据结构和算法** 在熟练使用抽象数据类型解决常见问题之后，学习如何实现它们是一件很自然的事情。

话题 1: 编程基础

初学编程的新手，一般应该熟练使用函数和库处理字符串相关的编程任务。

- C++ 基础
- 函数和库
- 字符串和流

American Soundex		Daitch-Mokotoff Soundex	Phonetic Matching
Waagenasz	Wegonge	Bassington	Bassington
Wachenhausen	Weismowsky	Bazunachden	Vasington
Wacknocty	Weuckunas	Bechington	Washincton
Waczinjac	Wiggins	Bussington	Washington
Wagenasue	Woigemast	Fissington	
Waikmishy	Wozniak	Washington	
Washington	Wugensmid	Vasington	4 names
Washincton	...	Washincton	
Wassingtom	+ 3,900 more names	Wassington	
...			
		9 names	

Figure 3: 语音算法

话题 2: 抽象数据类型的使用

在尝试实现抽象数据类型之前，应该先熟练使用这些工具解决问题。

- Vecotr、Grid、Stack、Queue
- Map、Set、Lexicon

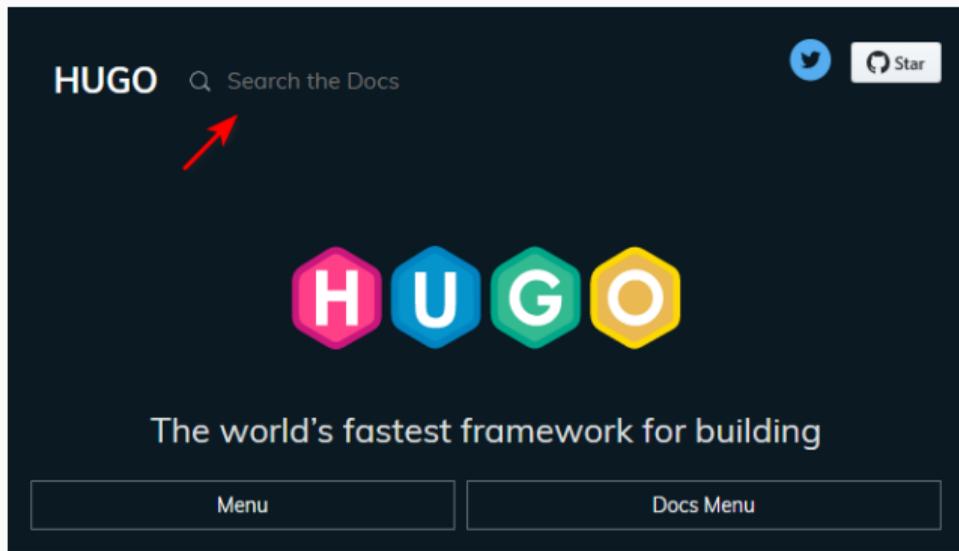


Figure 4: 搜索引擎

话题 3：递归和算法分析

递归是一种强有力的思想，一旦掌握就可以解决很多看起来非常难的问题。

- 递归过程
- 回溯算法
- 算法分析

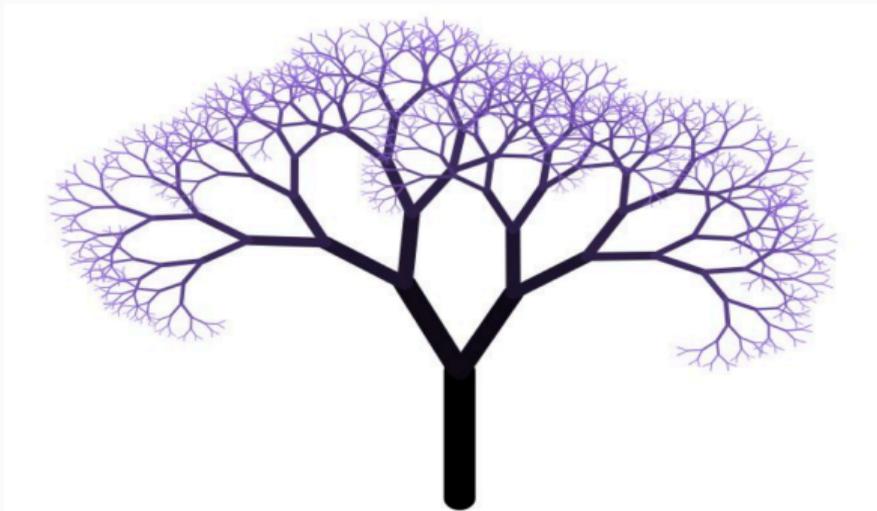


Figure 5: 递归艺术

话题 4: 类和内存管理

学习使用 C++ 实现数据抽象之前, 应先了解 C++ 的内存机制。

- 类的设计
- 指针和数组
- 动态内存管理

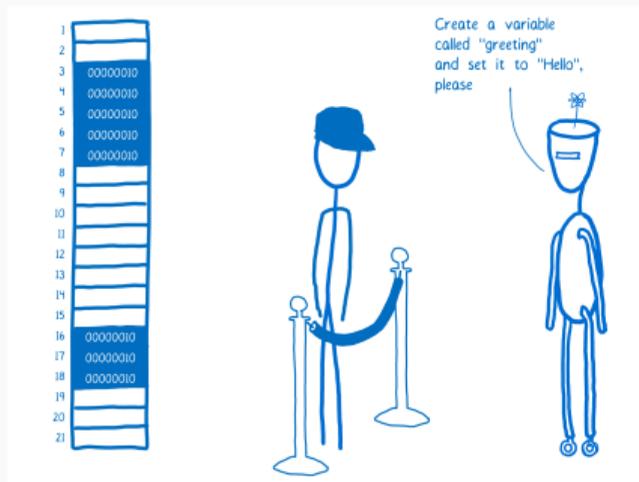


Figure 6: 数据封装和内存管理

话题 5: 常见数据结构和算法

在熟练使用抽象数据类型解决常见问题之后, 学习如何实现它们是一件很自然的事情。

- 链表
- 树
- 排序

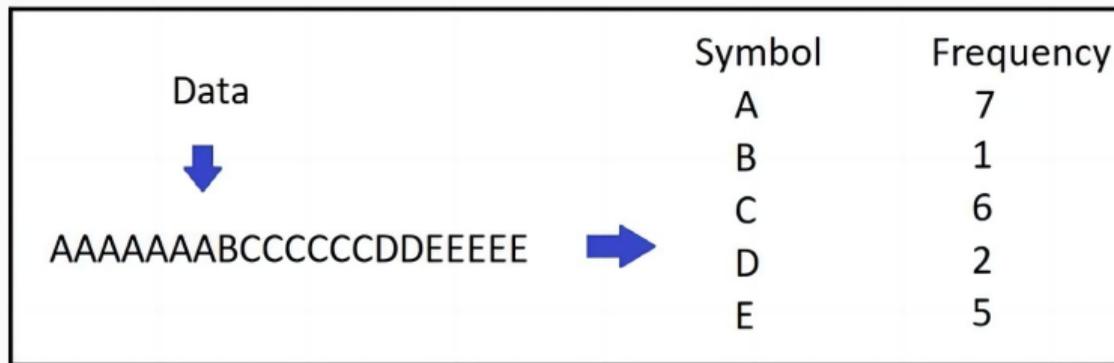


Figure 7: 数据结构和算法

C++ 语言介绍

C++ is a general purpose programming language with a bias towards systems programming that

- is a better C.*
- supports data abstraction.*
- supports object-oriented programming.*
- supports generic programming*

—Bjarne Stroustrup, inventor of C++ [2]



Hello, World!

```
/**
 * File: HelloWorld.cpp
 * -----
 * This file is adapted from the example
 * on page 1 of Kernighan and Ritchie's
 * book The C Programming Language.
 */

#include <iostream>

int main() {
    std::cout << "Hello, World" << std::endl;
    return 0;
}
```

年份	C++ 标准	非正式名称
1998	ISO/IEC 14882:1998	C++98
2003	ISO/IEC 14882:2003	C++03
2011	ISO/IEC 14882:2011	C++11, C++0x
2014	ISO/IEC 14882:2014	C++14, C++1y
2017	ISO/IEC 14882:2017	C++17, C++1z
2020	ISO/IEC 14882:2020	C++20, C++2a

Qt Creator 开发环境

Qt Creator 是一个跨平台集成开发环境 (IDE)，旨在为开发人员提供最佳体验。

Qt Creator 可以在 Windows、Linux 和 macOS 桌面操作系统上运行，允许开发人员跨桌面、移动和嵌入式平台创建软件。

下载：[Install Qt](#)

FIGURE 1-2 The compilation process

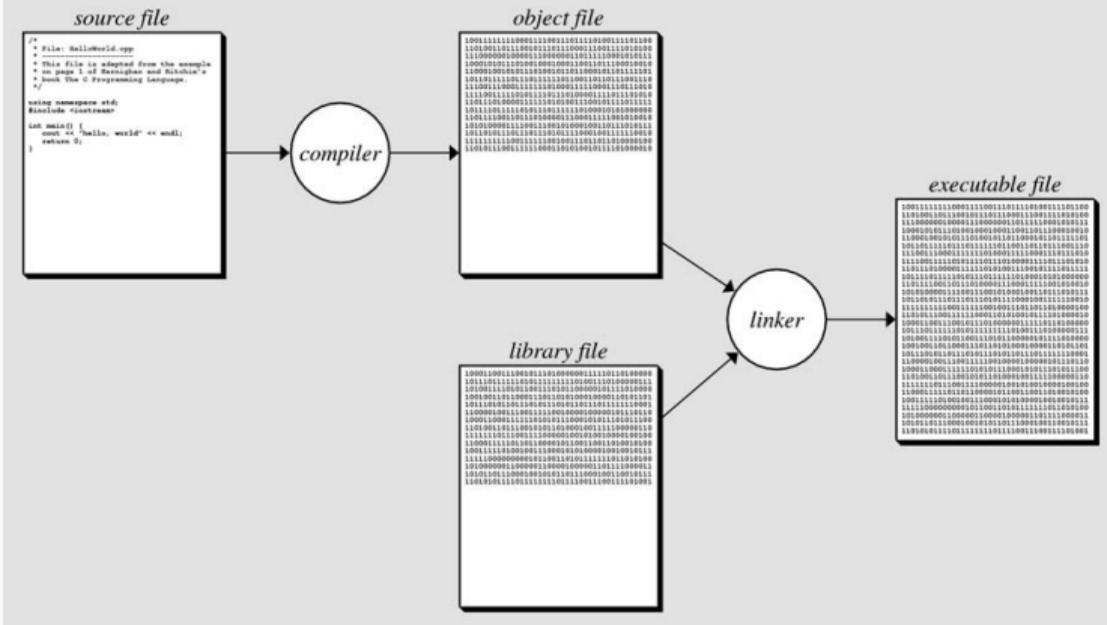


Figure 8: C++ 编译过程

CMake 构建工具

CMake 是一个开源的跨平台工具集，旨在构建、测试和打包软件。

CMake 用于使用简单的平台和编译器独立配置文件来控制软件编译过程，并生成可在您选择的编译器环境中使用的本机 Makefile 和工作区。

CMake 工具套件是由 Kitware 创建的，旨在响应 ITK 和 VTK 等开源项目对强大的跨平台构建环境的需求。

```
cmake_minimum_required(VERSION 3.20)

# Set project information
project(
  HelloWorld
  VERSION 1.0
  DESCRIPTION "The Hello World program"
  LANGUAGES CXX)

# Add an executable target
add_executable(${PROJECT_NAME})
```

```
# Define Cxx standard
set_target_properties(
  ${PROJECT_NAME}
  PROPERTIES CXX_STANDARD 17
             CXX_STANDARD_REQUIRED ON
             CXX_EXTENSIONS OFF)

# Add source files
target_sources(${PROJECT_NAME} PRIVATE
  ↪ src/HelloWorld.cpp)
```

- 课程介绍
- 编程语言
- 开发工具

问题?

- [1] Stanford. *CS Minor*. URL:
<https://www.cs.stanford.edu/degrees/undergrad/Minor.shtml>.
- [2] Bjarne Stroustrup. *The design and evolution of C++*. Pearson Education India, 1994.