



中国地质大学  
China University of Geosciences

# 绪论——关于课程

主 讲：王红平

Email: wanghp@cug.edu.cn

# 内容提要

---

- 学习动机
- 关于课程
- 学习方式

# 1、学习动机

- 为什么要学习面向对象程序设计？
- 你将学到什么技能？

# 学习动机——为什么要编程？

---

- 编程是人工智能时代必备的技能包
  - 可能你对工作不满意，想换一个高薪水的工作
  - 或者你想把自己从乏味的工作中解放出来，了解隐藏在机器背后的高级应用
  - 总之，如果你想获得一种解决问题的能力，编程是个不错的选择
-

# 学习动机——听听牛人们怎么说？

---

“Everybody in this country should learn how to program a computer... because it teaches you how to think.”

——Steven Jobs

## ■ Watch Video

- ⌘ What Most School don't teach (中文版)
  - ⌘ Inspire Students
-

# 什么是真正的编程能力？

**编程能力是一种解决问题的能力。**如果问题没能被很好地解决，知道再多也没用。

**编程能力是一种运用机器解决问题的能力。**首先是要判断问题在什么程度上可被机器解决，比如理论计算机科学会告诉我们什么可做、什么理论上不可做、什么理论上可做实践上不可做。然后是让机器更好地理解问题，比如计算机都是（图灵 - 冯诺依曼模型）等价，但不同的问题可能会适用不同的编程语言。再后是让机器能更高效地解决问题，比如同样的问题可能会有效率差别巨大的算法。

**编程能力是一种抽象问题的能力。**借用轮子是很好的办法，省力省时间。（程

# 什么才算是真正的编程能力？

**编程能力是一种需要考虑扩展性的能力。** 算法竞赛中的很多算法考虑的是单机的

内存算法，计算模型经过高度抽象，在实践中机器的模型更为复杂。比如单机的多级结构带来的各种时间空间复杂度的取舍平衡，多机网络中如何能在提高单机性能外进一步优化整体性能。除了在机器端的扩展，在程序员一端的扩展也很重要。

**编程能力是一种取舍的能力。** 局部的最优解未必是全局的最优解。如果一个美妙

的解决方案需要将完工时间向后推迟一两个月，需要考虑是否先使用平凡方案解决问题，之后再进行优化。当你的工作延后会阻碍别人的工作时尤其如此。发现一个绝妙的优化方案时先想想这个优化是否真的有价值，如果只是系统中很小的部分，那么不要为了追求心理满足而花很多时间放一个漂亮的轮子上去（参考 Amdahl 定律）。

**编程能力是一种预见未来的能力。** 目前的方案有哪些假设和局限性，在何种情

形下会遇到问题甚至崩溃。在未来出现问题时问题是否需要重新定义，系统是否需要重新设计，代码是否需要重构或优化等等都需要未雨绸缪。

# 什么才算是真正的编程能力？

**编程能力是一种工程能力。**无它，唯手熟尔。

**编程能力是一种解决问题的能力。**如果问题没能被很好地解决，知道再多也没用。

**编程能力是一种解决问题的能力。**如果问题没能被很好地解决，知道再多也没用。

**(重要的事情说三遍，重要的事情说三遍，重要的事情说三遍)**



## 2、关于课程

- 课程体系
- 授课方式
- 参考书
- 教学安排
- 考核办法

# 课程体系

学期	课程名称	学时数
大一上	面向对象程序设计（C++）	56学时
大一上	面向对象程序设计课程设计	1.5周
大一下	面向对象程序设计（C++）与Windows程序开发	64学时
大一下	面向对象程序设计课设	1.5周
大二上	数据结构与算法	60学时
大二上	数据结构与算法课程设计	2周
大三上	数字图像处理课程设计	2周
大三下	多门专业课的程序设计	? 周
大学四年	总计	>180学时+7周

# 授课方式

---

## ■ 授课

- ≡ 36学时

- ≡ 随时提问

## ■ 上机实习

- ≡ 20学时（5次）

# 教学参考书

---

## ■ 教材

- 《C++语言程序设计》（第3版），郑莉等，清华大学出版社

## ■ 参考书

- 《C++ Primer》，Stanley B. Lippman 和 Josée Lajoie著，潘爱民、张丽 译，中国电力出版社
  - 《C++程序设计语言（特别版）》，Bjarne Stroustrup 和 贝尔实验室 著，裘宗燕 译，机械工业出版社
  - 《实用C++调试指南》，於春景 译，华中科技大学出版社
  - 《C++语言程序设计教程》，谭浩强，清华大学出版社
-

# 教学安排

序号	内容
1	第一章 绪论C++语言概述
2	第二章 C++基本数据类型、表达式
3	第三章 算法的基本控制结构：循环、其它控制
4	第三章 自定义数据类型：枚举、结构、联合体
5	C++程序构成与IDE环境介绍
6	第四章 函数（函数的定义与使用）
7	第四章 内联函数、函数重载、标准库函数
8	第五章 类与对象基本概念
9	第五章 构造和析构函数、引用
10	第一~ 五章 习题课

# 教学安排（续）

序号	内容
11	第六章 作用域、生存期
12	第六章 多文件结构+课堂练习
13	第七章 数组
14	第七章 指针
15	第七章 字符串
16	第六~七章 习题课
17	课程复习

# 考核方式

---

考核名称	百分比
到课情况	10%
课堂（后）作业	10%
上机实习	30%
期末考试	50%

---

# 3、学习方式

- 大学学前班
- 程序设计入门
- 学习资源



# 大学学前班——应试学习VS.自主学习

---

## ■ 应试学习

- ㄥ 授课
- ㄥ 作业
- ㄥ 课本
- ㄥ 复习
- ㄥ 考试
- ㄥ 遗忘。。。。

## ■ 自主学习

- ㄥ 阅读参考资料
- ㄥ 自主学习与探索
- ㄥ 研讨
- ㄥ 项目
- ㄥ 过程考核
- ㄥ 印象深刻。。。。

兴趣是最好的老师！

---

# 大学学前班——学习的正确姿势

---

## ■ 学——输入

- ㄥ 看——见识
- ㄥ 听——接收
- ㄥ 读——强化
- ㄥ 写——强化

## ■ 思——加工

- ㄥ 总结规律
- ㄥ 举一反三

## ■ 习——输出

- ㄥ 模仿——熟练（熟能生巧）
  - ㄥ 实践——潜意识（变成习惯）
-

# 学习C++的路线图

---

- 简单程序 -> 熟悉简单程序的框架
  - 标准库 -> 模仿，入门要正，从优秀库开始
  - 小规模的程序 -> 通过写小程序，建立成就感
  - 量变到质变 -> 大量的积累
  - 建立面向对象思想 -> 通过模仿，写程序
  - 设计自己的类 -> 转换问题
  - 迈向大型开发 -> 积累经验
-

# 程序设计入门——几点忠告

## ■ 制定目标

\\ 开发专业系统、搭建网站、实现算法、开发游戏、iOS或Android的App开发

## ■ 不要急躁

Bad programming is *easy*. Even Dummies can learn it in 21 days.

Good programming requires **thought**, but *everyone can do it* and everyone can experience the *extreme satisfaction* that comes with it.

by Matthias Felleisen



## ■ 培养兴趣

Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.

– Linus Torvalds



# 程序设计入门——几点忠告（续）

## ■ 学好英语

- ∟ 大多数的资源都是用英语写成的，尽管机器翻译很厉害，但却容易丢失情感和思想。

## ■ 学会提问

- ∟ 提问要明确、清晰而具体，而不是含糊、宏观的问题

## ■ 不要做一匹孤狼

- ∟ 互联网改变了我们的世界，他的本质就是自由与分享。不要埋头苦干，自我封闭，而应该开放、分享并乐于助人。

## ■ 程序语言入门，不在于多而在于精

- ∟ 世上的编程语言多达100多种，不知如何选择？那就从C++开始。

## ■ 完事开头难，坚持不懈就会上升到新境界

- ∟ 勇敢跳出“舒适区”，会有不一样的体验
-

## 程序设计入门——几点忠告（续）

## ■ 不要纠结，上路就好

## 与其踌躇不前，不如保留好奇之心，就此上路

## ■ 学习程序的秘诀

## 编程、编程、再编程

≡ 要说有的话，那就是上机coding、debug、reading and thinking

## ■ 遇到错误不要惊慌失措

新的技能就是在不断的犯错、改错中获得的，错误没什么大不了

## ■ 入门要正

## 养成良好的编程习惯：细心、遵守编码规范、添加足够的注释

## ■ 请热爱C++!

## 没有真心的付出，是没有资格评价编程语言优劣的

# 电子资源与参考资料

---

## ■ 网络课程

- ⋋ [Learn C++](#) (English)
- ⋋ [面向对象程序设计-C++ - 网易云课堂](#)
- ⋋ [www.code.org](http://www.code.org)

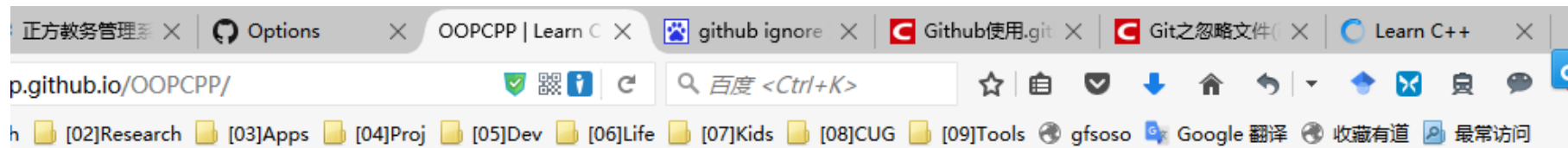
## ■ 查询问题

- ⋋ [中国最大的IT技术社区CSDN](#)
- ⋋ [www.stackoverflow.com](http://www.stackoverflow.com)
- ⋋ [Google](#)

## ■ 国外网站

- ⋋ [www.codeproject.com](http://www.codeproject.com)
  - ⋋ [www.codeguru.com](http://www.codeguru.com)
  - ⋋ [www.sourceforge.net](http://www.sourceforge.net)
  - ⋋ [www.github.com](http://www.github.com)
-

■ <https://cugwhp.github.io/OOPCPP/>



## OOPCPP

This is the resource about the course of Orient Object Programming Language C++. There are course lecture, exercises, example code and reference books.

[TOC]

### 0. Preview

Inspire Students to coding

Bad programming is *easy*. Even Dummies can learn it in 21days.

Good programming requires **thought**, but *everyone can do it* and everyone can experience the *extreme satisfaction* that comes with it.



# Home Work

- 1. 阅读[编程小白学Python](#)节选（[完整版](#)），完成一篇读后感即课程学习目标。
- 2. 登录[电子资源和参考资料](#)提供的网站，了解C++备用
- 3. 选读[大学学前班](#)，好好思考大学规划

