

面向对象软件开发技术 -课程介绍

李长河

中国地质大学（武汉）自动化学院

lichanghe@cug.edu.cn

教材

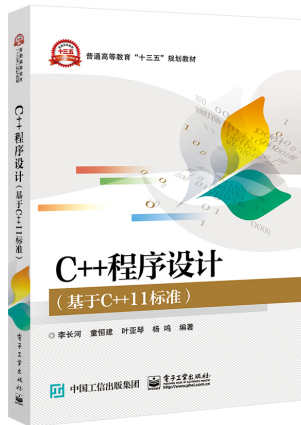
李长河, 童恒建, 叶亚琴等, C++ 程序设计 (基于 C++11 标准) .
电子工业出版社, 2019 年 10 月第 3 次印刷

电子资源

<https://github.com/Changhe160/cplusplus2020-2021-1>

参考书

Stanley B. Lippman, Josée Lajoie, Barbara E. Moo. C++ Primer
(第五版). 王刚等译. 北京: 电子工业出版社, 2013.

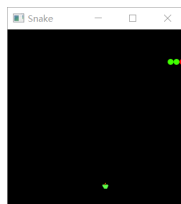
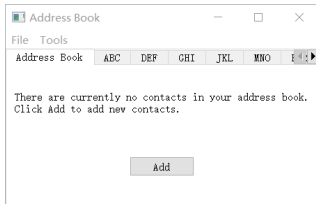


学习目标

- ① 掌握面向对象程序设计方法；
- ② 运用常见数据结构和算法解决实际问题；
- ③ 熟悉可视化程序设计开发工具（Qt）。

学习目标

- 1 掌握**面向对象**程序设计方法；
- 2 运用常见**数据结构和算法**解决实际问题；
- 3 熟悉可视化程序设计开发工具（Qt）。

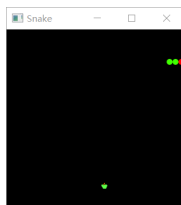
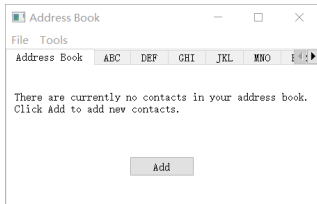


学习目标

- ① 掌握**面向对象**程序设计方法；
- ② 运用常见**数据结构和算法**解决实际问题；
- ③ 熟悉**可视化程序设计开发工具 (Qt)**。

课时安排

讲课学时：28，实验学时：4（上机考试）



学习目标

- 1 掌握**面向对象**程序设计方法；
- 2 运用常见**数据结构和算法**解决实际问题；
- 3 熟悉**可视化程序设计开发工具 (Qt)**。

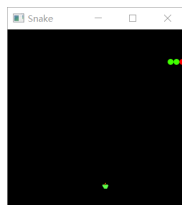
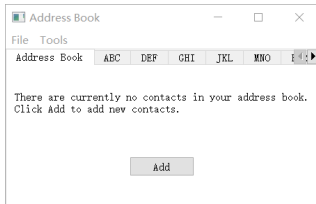
课时安排

讲课学时：28，实验学时：4（上机考试）

课程考核

课程总成绩 = 考勤 *5%+ 四次上机考核 *40%+1 次综合考核 *40%+ 课程报告 *15%

考核方式：线上提交 + 线上验收 <http://course.educg.net/indexcs/simple.jsp?loginErr=0>



课程纪律

- ① 课上严禁看手机；
- ② 上机验收严禁抄袭，一经发现，均记 0 分处理。

上机考试验收评分细则

考察点：诚信、可读性、功能实现、完成数量、性能和效率、创新性

分数	评分依据
违纪	具有抄袭行为，一旦发现并确认，所有涉及的学生均按 0 分处理；
不及格	没有掌握基本概念； 无法编写和调试简单程序；
60-69	至少完成一个题目，测试结果正确； 能够调试代码、解释和回答与代码相关的问题和基本概念；
70-79	代码书写工整，满足可读性要求； 能够解释检查者做出的简单修改所提出的问题； 有少量逻辑错误；
80-89	程序结构合理，合理使用头文件和源文件； 完成所有题目且输出正确测试结果； 没有逻辑错误； 程序的执行效率和安全性强； 能够根据检查者提出的要求修改源代码；
90-100	有较强语法运用和算法设计能力； 鼓励创新，具有独到的见解和想法； 能够在规定时间内（15 分钟）实现检查者提出的新功能；

计算机程序设计课程组

自动化学院

课程纪律

- ① 课上严禁看手机；
- ② 上机验收严禁抄袭，一经发现，均记 0 分处理。

课程群

QQ 群：940393564
助教：杨瑞

上机考试验收评分细则

考察点：诚信、可读性、功能实现、完成数量、性能和效率、创新性

分数	评分依据
违纪	具有抄袭行为，一旦发现并确认，所有涉及的学生均按 0 分处理；
不及格	没有掌握基本概念； 无法编写和调试简单程序；
60-69	至少完成一个题目，测试结果正确； 能够调试代码、解释和回答与代码相关的问题和基本概念；
70-79	代码书写工整，满足可读性要求； 能够解释检查者做出的简单修改所提出的问题； 有少量逻辑错误；
80-89	程序结构合理，合理使用头文件和源文件； 完成所有题目且输出正确测试结果； 没有逻辑错误； 程序的执行效率和安全性强； 能够根据检查者提出的要求修改源代码；
90-100	有较强语法运用和算法设计能力； 鼓励创新，具有独到的见解和想法； 能够在规定时间内（15 分钟）实现检查者提出的新功能；

计算机程序设计课程组

自动化学院

7.1 代码复用

是什么在影响我们的开发效率？

只谈程序员自身因素，06年写过的一段，供参考

熟练人员经过多年的积累加上自己的CodeSnip的总结，基本不用额外再查找资料。而一般的开发人员在开发过程中会花掉10-20%时间去查找资料。

熟练人员注意代码复用，并且时刻注意重构和抽取公用代码。一般开发人员是代码拷来拷去完成功能。

熟练人员非常注意查找，定位，标签等各种快捷键的使用，定位查找方便快捷，IDE环境也根据习惯定义到最方便状态。

熟练人员编码前先思考清楚整个流程，在头脑或纸张上规划好整个实现方式和方法函数的划分。一般人员想到哪里写到哪里。

熟练人员写了50行以上或更多代码才Debug一两次，一般人员写了几行代码就要Debug多次，完全通过Debug来验证代码正确性。

熟练人员注重代码的质量，单元测试和可维护性，注重各种业务逻辑的验证和边界条件的校验。一般人员只注重简单功能的简单完成。

7.1 代码复用

是什么在影响我们的开发效率？

只谈程序员自身因素，06年写过的一段，供参考

熟练人员经过多年的积累加上自己的CodeSnip的总结，基本不用额外再查找资料。而一般的开发人员在开发过程中会花掉10-20%时间去查找资料。

熟练人员注意代码复用，并且时刻注意重构和抽取公用代码。一般开发人员是代码拷来拷去完成功能。

熟练人员非常注意查找，定位，标签等各种快捷键的使用，定位查找方便快捷，IDE环境也根据习惯定义到最方便状态。

熟练人员编码前先思考清楚整个流程，在头脑或纸张上规划好整个实现方式和方法函数的划分。一般人员想到哪里写到哪里。

熟练人员写了50行以上或更多代码才Debug一两次，一般人员写了几行代码就要Debug多次，完全通过Debug来验证代码正确性。

熟练人员注重代码的质量，单元测试和可维护性，注重各种业务逻辑的验证和边界条件的校验。一般人员只注重简单功能的简单完成。

7.1 代码复用

是什么在影响我们的开发效率？

只谈程序员自身因素，06年写过的一段，供参考

熟练人员经过多年的积累加上自己的CodeSnip的总结，基本不用额外再查找资料。而一般的开发人员在开发过程中会花掉10-20%时间去查找资料。

熟练人员注意代码复用，并且时刻注意重构和抽取公用代码。一般开发人员是代码拷来拷去完成功能。

熟练人员非常注意查找，定位，标签等各种快捷键的使用，定位查找方便快捷，IDE环境也根据习惯定义到最方便状态。

熟练人员编码前先思考清楚整个流程，在头脑或纸张上规划好整个实现方式和方法函数的划分。一般人员想到哪里写到哪里。

熟练人员写了50行以上或更多代码才Debug一两次，一般人员写了几行代码就要Debug多次，完全通过Debug来验证代码正确性。

熟练人员注重代码的质量，单元测试和可维护性，注重各种业务逻辑的验证和边界条件的校验。一般人员只注重简单功能的简单完成。