

# 面向安全思维的 C 语言程序设计课程 教学改革与实践<sup>\*</sup>

彭聪 王慧 冯琦 何德彪 吴黎兵 罗敏<sup>\*\*</sup>

武汉大学国家网络安全学院, 武汉 430072

**摘要** 随着信息技术的快速发展和网络安全威胁的日益加剧, 社会对安全意识和安全能力的培养越来越重视。本文针对目前高校中程序设计类课程普遍存在的理论与实践脱节、教学方法单一和安全思维培养滞后等问题, 分析 C 语言程序设计课程教学中对代码安全问题关注不足、学生安全意识培养缺失等现象, 提出了以“目标驱动、课堂融合、平台联动、多维评价”为核心的教学改革思路, 将安全思维贯穿于教学全过程。从课前预习、课中学习、课后复习和拓展提升四个环节开展系统性教学改革措施, 着力培养学生的编程实践能力与代码安全意识。实际教学结果表明, 该改革能够有效增强学生的安全思维与实践能力, 项目实践得分率从 81% 提高到 92%, 有效提升了程序设计类课程的教学质量。

**关键字** 安全思维, 计算机程序设计, 工程实践, 教学改革

## Reform and Practice of C Programming Education Oriented Toward Security Awareness

Cong Peng Hui Wang Qi Feng Debiao He Libing Wu Min Luo

School of Cyber Science and Engineering of Wuhan University  
Wuhan 430072, China;

**Abstract**—With the rapid development of information technology and the growing threat of cybersecurity risks, society is increasingly emphasizing the importance of cultivating security awareness and capabilities. This study addresses the common issues in current university-level programming courses, such as the disconnect between theory and practice, limited teaching methods, and the lagging development of security thinking. It analyzes the insufficient attention to code security issues and the lack of security awareness cultivation in C language programming course instruction, and proposes a teaching reform approach centered on “goal-driven, classroom integration, platform collaboration, and multi-dimensional evaluation,” integrating security thinking throughout the entire teaching process. Systematic teaching reform measures are implemented across four phases: pre-class preparation, in-class learning, post-class review, and extension and enhancement. These measures focus on cultivating students' programming practical skills and code security awareness. Actual teaching results demonstrate that this reform effectively enhances students' security thinking and practical abilities, with project practice scores increasing from 81% to 92%, thereby significantly improving the teaching quality of programming courses.

**Keywords**—Security Awareness, Computer Program Design, Engineering Practice, Teaching Reform

## 1 引言

C 语言因其高效性、灵活性和强可移植性等优势, 多年以来被广泛应用于操作系统内核、嵌入式系统和数据库系统等核心领域, 在软件开发领域具有举足轻重的角色<sup>[1]</sup>。但因缺乏内置安全机制及人为操作不当等因素, 使得 C 语言极易引发内存泄漏、越界访问和堆栈溢出等安全问题, 成为许多严重安全漏洞的高发源头<sup>[2]</sup>。2024 年, 美国白宫国家网络主任办公室在《回到基础构件: 通往安全软件之路》报告中呼吁开发者

应尽快采用“内存安全的编程语言”, 停止使用 C/C++。

近年来, 随着国家对软件安全问题重视程度的日益提升, 产业界对高等教育人才培养也提出了新的需求和挑战<sup>[3]</sup>。以 C 语言为代表的程序设计类课程作为计算机类专业的核心课程, 是培养高等教育人才编程实践能力和代码安全意识的关键环节。然而, 目前高校中程序设计类课程普遍存在理论与实践脱节、教学方法单一和安全思维培养滞后等问题<sup>[4]</sup>。因此, 探索面向安全思维的程序设计类课程改革的实践路径, 对培养具有创新精神、实践能力和安全意识的高素质工程技术人才具有重要意义。

## 2 教学发展困境

<sup>\*</sup>基金资助: 本文得到 2024 年蚂蚁智信(杭州)信息技术有限公司教育部产学研合作协同育人项目“隐私计算开发与实践课程建设”资助。

<sup>\*\*</sup>通讯作者: 罗敏 mluo@whu.edu.cn。

在面向安全思维背景下，以C语言程序设计课程为代表的程序设计类课程主要面临着课程内容不全面、实践教学不足和评估方式受限三个方面的教学难题。

## 2.1 课程内容不全面

传统教学模式过于依赖教师讲授，学生往往被动接受知识，课堂互动性较差<sup>[5]</sup>。而且课程内容普遍忽视对代码规范和安全意识的系统培养，导致学生在编程过程中易忽略潜在的漏洞和安全隐患，缺乏应对真实开发环境的能力<sup>[6]</sup>。这种教学方式不仅导致学生学习积极性不高，而且削弱其规范解决问题的能力。例如，学生通常只记住C语言的基本语法，如变量声明、循环结构和条件判断等，但在面对复杂项目时，往往难以将所学的内容与实际问题的有效结合。

## 2.2 实践教学不足

当前课程中的理论部分与实践环节未能有效衔接，导致学生理论知识难以转化为实际操作能力。在课程实践教学中，大部分时间用在概念讲解上，实践时间较少，且实践操作往往停留在“照猫画猫”式的练习上，缺乏解决实际问题的深度与挑战性<sup>[7]</sup>。这种教学设计限制了学生的逻辑思维发展，也削弱了其工程实践能力的培养。

## 2.3 评估方式受限

许多高校的考核方式以期末考试为主，未能充分重视对学生学习过程的动态评价<sup>[8]</sup>。这种单一评估方式难以全面反映学生的综合能力，尤其是在实践能力与创新思维方面。因此，考核结果与学生实际水平之间存在一定差距，学生可能在理论考试中取得较好成绩，但在实践操作中能力较弱。

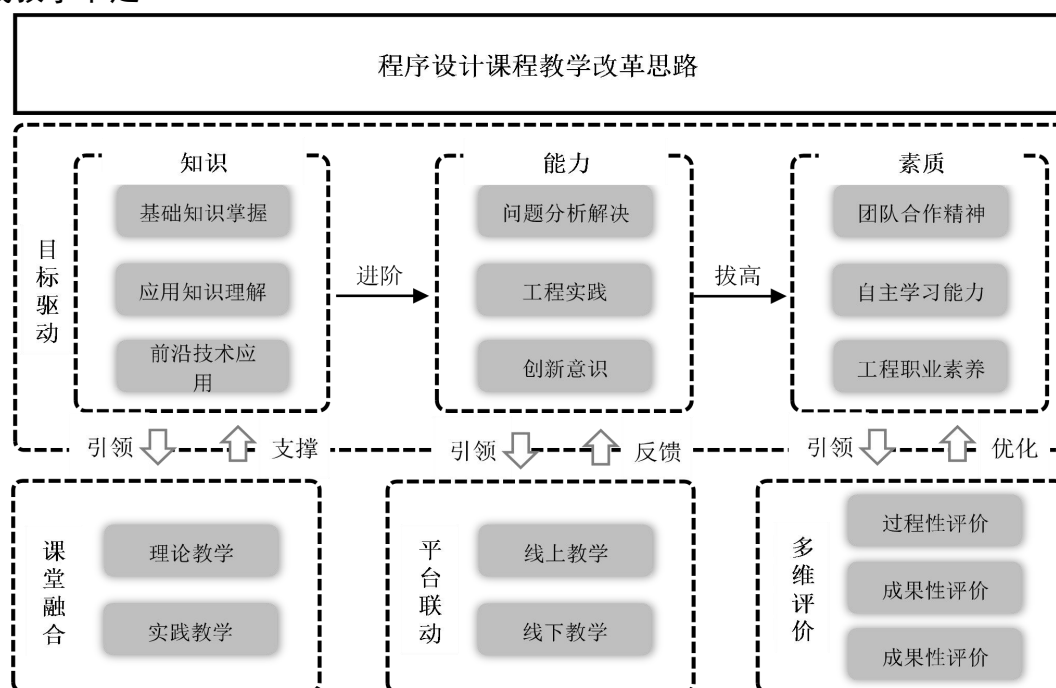


图1 教学改革思路图

## 3 教学改革思路

针对当前程序设计课程的教学中的理论与实践脱节、教学方法单一和安全思维培养滞后发展困境，本文提出了以“目标驱动、课堂融合、平台联动、多维评价”四位一体的教学改革思路，整体改革思路如图1所示。

### 3.1 目标驱动

课程目标的制定应紧密围绕当前行业的实际需求，同时需要充分考虑武汉大学学生的知识基础、学习特点与学校人才培养的整体要求，充分调动学生学

习积极性。

一方面，参照工程教育专业认证标准，结合学生的毕业要求，明确课程对知识吸收、能力培养和素质提升的综合目标。在知识吸收方面，强调程序设计核心知识体系的构建；在能力培养方面，着重提升学生的问题分析与解决能力、工程实践能力与创新意识；在素质提升方面，培养学生的自主学习能力与团队合作精神，为其未来职业发展筑牢根基。

另一方面，立足应用型人才培养定位，将行业前沿技术、岗位能力需求与课程结合。例如将数据结构实践、系统编程等行业典型任务引入课堂，在课程中

增加复杂数据结构实现、多线程编程等项目实践, 让学生了解实际行业开发需求。

### 3.2 课堂融合

课堂融合是指理论教学与实践教学有机结合, 确保学生将所学知识应用于解决实际工程问题。

在理论教学环节, 围绕基础知识点, 结合现实生活情境设计富有趣味的编程任务, 如简单计算器、随机迷宫、人机五子棋等, 激发学生学习兴趣。引入结构化程序设计思想, 培养学生“自顶而下, 逐步求精”的编程能力。引导学生从问题出发, 先总体再细节, 逐步分解细化, 将解决问题的步骤分解为由基本程序结构模块组成的结构化程序框图。

在实践教学环节, 将学科竞赛与课堂内容有机融合。引导学生接触“蓝桥杯”大赛、ACM/ICPC 国际大学生程序设计大赛、中国大学生计算机设计大赛等经典竞赛题型。同时, 结合课程内容设计实践项目, 如开发简单管理系统或校园服务类应用等, 锻炼学生的实践能力。

### 3.3 平台联动

依托信息化教学平台, 充分整合线上与线下资源, 形成优势互补的混合教学模式, 激发学生对程序设计课程的兴趣, 提升自主学习能力。

线上环节利用在线平台发布课程视频、实验作业与讨论话题。课程视频按知识点分段讲解, 每段聚焦一个核心概念或操作, 便于学生随时回顾。实验作业覆盖各阶段重点与难点, 结合经典实例, 引导学生深入理解关键知识。线上讨论鼓励学生积极交流思路、互帮互助, 营造良好学习氛围。

线下环节以小组为单位开展案例实践, 组织研讨交流活动。例如, 安排学生通过小组合作实现大整数运算库项目, 完成多文件项目工程的设计、模块功能划分、代码编写、编译链接与测试等功能, 全面锻炼学生的工程实践与团队协作能力。在研讨交流环节, 鼓励学生分享思路与遇到的问题, 通过思维碰撞, 加深对知识的理解与应用。

### 3.4 多维评价

采用多维度评价体系, 全面、系统地评估学生的学习效果, 根据学生个性特长开展有针对性的因材施教。过程性评价注重学习过程的监控与指导, 涵盖日常作业、课堂表现和实验任务等环节, 评估学生知识点掌握程度和实践能力; 成果性评价通过综合项目、课程设计报告及答辩展示等形式, 检验学生学习成果、沟通能力以及创新能力的综合应用水平; 多元主体评价则结合教师、学生互评与自评, 全面客观的反映学

生综合能力。

## 4 教学改革措施

根据上述教学改革思路, 将从课前预习、课中学习、课后复习及拓展提升四个环节实施教学改革, 构建评价连贯、动态反馈、持续优化的闭环教学流程, 系统提升学生的工程素养与自主学习能力。

### 4.1 课前预习

课前教师通过教学平台发布预习内容, 为课堂学习奠定基础。预习内容包含核心知识框架、基础概念讲解、预习作业。知识框架以思维导图形式呈现, 清晰展示课上所需知识点之间的逻辑关系; 制作基础概念讲解视频, 每个视频聚焦一个核心知识点, 时长控制在 10-15 分钟, 便于学生利用碎片化时间学习; 布置预习作业, 让学生根据初步学习情况进行基础概念填空、简单程序编写, 根据学生理解情况调整课堂教学重点。

### 4.2 中学习

课堂学习是吸收知识重中之重。具体来说从教学内容和教学方式两个维度进行开展和强化。

教学内容方面, 根据行业前沿技术、岗位能力需求进行优化。提供基础模块, 围绕程序设计基础语法与算法进行教学, 确保学生掌握基本程序设计、程序流程控制、函数、数组等基础知识; 提供核心模块, 强化面向过程编程思想, 提升学生抽象建模能力。课程内容涵盖指针、符合数据类型等核心技术。例如要求学生编写函数实现二维整型数组所有正数求和, 考察基本数组知识、嵌套循环知识掌握程度; 提供应用模块, 为学有余力的同学提供发展空间。例如开发一个简易的学生管理系统, 学生需综合运用指针、结构体和文件操作知识实现数据的动态增删改查功能, 重点培养学生的工程实践思维能力; 补充常见安全编码知识, 降低引入错误的可能性。ISO 26262 等安全标准表明, 编码标准可以提高程序的安全性。适当引入 CWE (常见软件缺陷列表) 知识, 提升学生对编程安全问题的认识。

教学方式方面, 为提升学生课堂参与度, 课上采用多元化的教学方法。增加翻转课堂, 让学生课前自主学习、准备基本程序设计的讲解, 课上进行分组展示。期间, 其他学生可随时提出疑问, 教师引导学生共同分析问题; 设置探究性问题: 教师提出开放性问题, 引导学生查阅资料、小组讨论等方式, 自主探索解决方案。不断鼓励学生尝试不同方法, 培养创新思维和问题解决能力; 提供案例教学, 引入华为、小米、京东等公司经典面试题和真实工程案例激发学生兴趣。

4.3 课后复习

课后作业实施线上线下双轨并行模式，让学生及时巩固课堂学习成果。

线上教学环节通过在线评测系统（OJ 平台）构建智能化实验环境。教师通过平台发布实验任务，任务按难度分为基础、提高、挑战三个等级。学生提交代码后，系统自动执行测试，即时反馈运行结果，让学生快速定位问题。教师可通过分析学生代码提交次数、提交时间等信息，了解学生知识掌握程度，优化后续教学策略。此外，平台提供丰富的实验案例库和教学资源，学生可根据自身能力选择进阶任务，满足个性化学习需求。

线下教学环节通过精心设计的课后习题与开放性编程问题等，帮助学生扎实掌握语法、算法等核心知识；同时，依托小米-武汉大学、京东-武汉大学、OPPO-武汉大学等校企联合实验教学研究基地，定期组织学生开展实地观摩与技术交流，引导学生了解行业真实场景与技术需求，拓宽学生技术视野，从而增强工程实践与创新能力。

4.4 拓展提升

结合“蓝桥杯”编程大赛等创新实践项目，设置项目实践环节，让学生以小组为单位完成项目开发，培养学生创新能力、项目管理及工程实践能力。在项目开发过程中，引导学生关注 C 语言编程安全问题，如内存泄漏、溢出等，培养学生的安全编程意识。

结合上述四种方法对学生进行全方位的培养，并构建多元化考核体系，综合考察学生学习情况。例如，

将成绩考核分为平时表现、实验任务和期末项目三个环节，各环节评分比例根据武汉大学人才培养目标进行设定。平时表现包括课堂参与度、作业完成质量（含线上线下作业）和实验任务过程表现。实验任务包括完成度、代码规范性及算法效率。期末项目指以小组为单位完成的综合实践项目。通过现场答辩、学生互评、小组评估等方式，确保考核结果全面、客观反映学生综合能力。具体教学改革措施如表 1 所示。

表 1 教学改革措施示范表

措施	实施过程	培养内容
课前学习	自主预习	课前发课程相关的预习课件，培养学生自学能力
	课前测验	课前通过在线测验平台检测基础知识掌握
课堂学习	要点讲解	回顾关键知识点，帮助学生巩固自学内容，并补充薄弱环节
	探究学习	设置开放性问题，培养学生的团队合作精神与批判性思维能力
	随堂测试	及时反馈学习效果，提升问题解决能力与代码实现能力
课后学习	课后测验	巩固课程核心内容，提升学生解决复杂问题的能力
	辅导答疑	解决学生个性化问题，完善其知识结构并增强应用能力
拓展学习	项目实践	培养学生创新能力、项目管理能力和工程实践能力

2023与2024年学生成绩对比

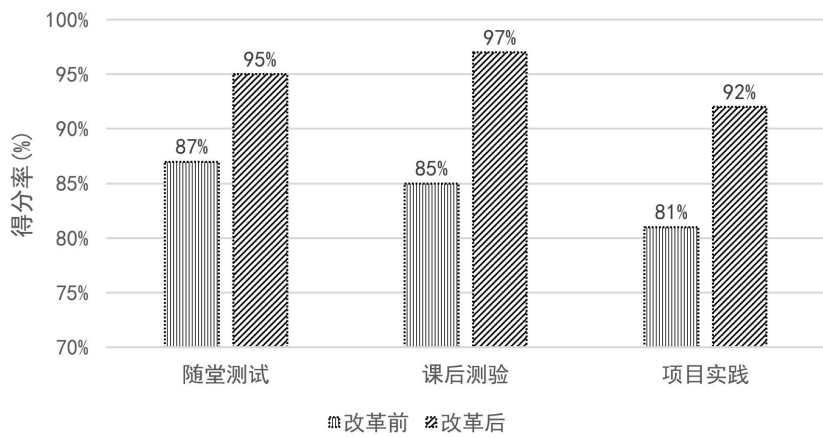


图 2 教学改革效果图

5 教学改革效果

为评估教学改革的实际效果，对武汉大学 2023 级、2024 级计算机科学与技术专业的 C 语言程序设计

课程改革效果进行跟踪评估，具体改革成效如图 2 所示。

（1）学生学习效果提升：通过对比 2023 至 2024 学年度学生的出勤率、作业质量、实验情况及期末成

绩等记录,可看出改革后学生随堂测试得分率由 87% 提升至 95%,优秀率保持在 18%左右;此外,课后测验得分率由 85%提升至 97%,项目实践得分率由 81%提升至 92%。结果表明,改革后学生的各项成绩相较之前均有显著提升。

(2) 实践能力显著提高:学生在省级和国家级程序设计竞赛中累计获得 20 余项奖项,其中一等奖获奖比例相较于改革前增加了 10%。改革期间,学生对竞赛参赛积极性更高,且倾向于团队协作参赛,可看出学生团队合作意识和实践能力有明显提高。

(3) 课程知识点掌握提升:通过分析随堂测试、在线平台实验及期末考试学生错题情况,发现改革后学生对指针正确使用及 C 语言编程漏洞相关题目得分率更高。

## 6 结束语

教育是强国建设、民族复兴之基。为提升学生的编程能力及安全意识,提出“目标驱动、课堂融合、平台联动、多维评价”的改革思路,并通过课前预习、课中学习、课后复习及拓展提升四个环节的系统性改革措施

理论与实践相结合、知识与能力协同发展的育人体系。为行业培养专业基础扎实、符合发展需求的高素质人才,为推动全球教育事业发展贡献更多中国力量。

## 参考文献

- [1] 姚立红,邱卫东. 面向网络空间安全专业的 C/C++ 程序设计教学研究[J]. 计算机教育, 2020(01): 93-95,99.
- [2] 尹良泽,徐建军,李姗姗,等. C 语言程序设计课程中的代码安全意识和能力培养[J]. 计算机教育, 2025(04): 53-57.
- [3] 新工科背景下“大学计算机基础”课程教学改革研究与实践[J]. 计算机技术与教育学报, 2022, 10(1): 77-80.
- [4] 孙彦斌,李默涵,田志宏. 面向工业安全的跨领域网络安全课程建设探索实践[J]. 计算机技术与教育学报, 2025, 13(1): 97-102.
- [5] 廖兴宇,刘海龙. 新工科背景下《数据结构》课程重构与实践研究[J]. 计算机技术与教育学报, 2025, 13(1): 42-46.
- [6] 李永庆,孙丽敏,孙媛媛,等. 国产化背景下 C 语言课程建设与创新实践[J]. 计算机技术与教育学报, 2024, 12(5): 19-23.
- [7] 姜锋. 新工科背景下“C 语言程序设计”课程教学改革探索[J]. 中国新通信, 2024, 26(19): 118-120.
- [8] 曹曼曼. 新工科背景下计算机程序设计类课程实验教学创新改革[J]. 计算机教育, 2024(05): 64-69.