

计算机类课程螺旋式能力进阶探索与教学实践

胡亚慧 胡俊 刘亮 瞿师 肖蕾

空军预警学院, 武汉 430014

摘要 计算机课程兼具严谨的理论体系与极强的实践属性,其核心目标是让学生掌握用计算机解决实际问题的能力,这既需理解逻辑严密的理论知识,也需完成一系列实操训练并实现知识的落地应用。然而学生在学习过程中,面对大量实践任务时常陷入无从下手的困境:不仅难以将课堂所学迁移到真实应用场景,面对复杂问题时更是无从下手。本文采用螺旋式能力进阶教学模式,设计三轮螺旋循环的教学架构:通过递进式训练,让学生在知识的循环迭代中夯实理论基础,在实操演练中提升动手能力,最终形成“理论扎实、动手娴熟、思维活跃”的综合素养。经实践效果证明,该模式显著提升了教学质量与效率,可为计算机类课程的教学改革提供可复用的教学范式与实践支撑。

关键字 螺旋式能力进阶, 渐进式学习, 教学改革, 能力培养

Exploration and Teaching Practice of Spiral Ability Progression in Computer Courses

Hu Yahui Hu Jun Liu Liang Shi Qu Lei Xiao

Air Force Early Warning Academy, Wuhan 430014, China

Abstract—Computer courses possess both a rigorous theoretical system and a strong practical component. Their core objective is to equip students with the ability to solve practical problems using computers. This requires not only understanding rigorous theoretical knowledge but also completing a series of practical training exercises and implementing the knowledge into practice. However, during the learning process, students often find themselves at a loss when faced with a large number of practical tasks: not only is it difficult to transfer classroom knowledge to real-world application scenarios, but they are also at a loss when confronted with complex problems. This paper adopts a spiral-type ability progression teaching model and designs a three-round spiral cycle teaching framework: through progressive training, students can consolidate their theoretical foundation through iterative cycles of knowledge, enhance their practical skills through hands-on exercises, and ultimately develop comprehensive qualities of "solid theory, proficient hands-on skills, and active thinking". Practical results have shown that this model significantly improves teaching quality and efficiency, providing a reusable teaching paradigm and practical support for teaching reforms in computer-related courses.

Key words—Spiral progression of abilities, Progressive learning, Teaching reform, Ability cultivation

1 引言

当前计算机类课程教学中普遍面临多重困境:其一,实践操作难度与学生能力脱节,课程设置中实践任务往往直接对接复杂应用场景,缺乏梯度引导,导致学生面对代码编写、算法实现等任务时无从下手;其二,学生问题分析能力薄弱,遇到程序报错、功能异常等问题时,仅能停留在“复制报错信息查询”的浅层操作,无法拆解问题本质、定位核心矛盾;其三,理论与实践严重割裂,学生虽能记忆编程语言语法、算法原理等知识点,但面对复杂实际需求时,无法将抽象理论转化为具体的代码实现与系统构建。

本文针对上述核心痛点,以布鲁纳螺旋式课程理论和皮亚杰认知发展理论为支撑,围绕“问题抽象-模型构建-计算机求解”的核心能力,构建“基础认知

-实践应用-创新拓展”的递进式培养体系。通过三轮螺旋循环,让学生在重复中深化理解,在应用中提升能力,最终形成“理论扎实、动手娴熟、思维活跃”的综合素养,适配计算机领域对工程实践与创新能力的双重需求。

2 螺旋式能力进阶培养教学目标

螺旋式递进培养模式以学生认知规律和能力发展路径为核心,通过“基础奠基—迁移应用—创新突破”三个阶段的循环迭代,实现知识与能力的螺旋上升。该模式贴合计算机课程“理论与实践并重”的特性,各阶段既相互独立又层层衔接,最终达成从“知识吸收”到“能力输出”的完整培养闭环,具体阶段划分与核心要求如下:

阶段一:基础认知奠基期——筑牢知识根基,掌握核心入门能力

* 通讯作者: 胡亚慧 hyh5800@163.com。

本阶段以“夯实基础、建立认知”为核心目标，聚焦计算机学科的底层知识与基本应用，为后续能力进阶搭建稳固框架。教学内容围绕“够用、实用”原则设计，避免复杂理论堆砌，重点引导学生建立对知识的初步理解与应用意识。

该阶段学生能够独立完成基础实操任务，如编写100-500行的简单应用程序（计算器、学生信息登记表等）、完成数据库单表操作、配置基础网络连接等，形成对计算机知识“是什么、怎么用”的基础认知。

阶段二：知识迁移提升期——聚焦复杂场景，强化问题解决能力

本阶段以“知识迁移、能力深化”为核心目标，打破基础阶段的单一知识点应用局限，引导学生将零散知识整合为“解决问题的能力模块”，重点提升在复杂场景中的分析与实践能力。教学内容以“真实应用场景”为载体，倒逼学生主动关联、灵活运用前期知识。

该阶段学生能够独立应对中等复杂度的实践任务，如开发包含登录、数据查询、统计功能的小型管理系统，完成多网段网络配置与故障排查，针对具体需求设计并优化简单解决方案，形成“问题—知识—方案”的初步关联思维。

阶段三：创新突破拓展期——对接前沿场景，培

育创新实践能力

本阶段以“创新应用、能力拔高”为核心目标，跳出既有场景的局限，引导学生面向未来技术趋势与更复杂的实际需求，实现从“解决问题”到“创造方案”的跨越，培养符合计算机领域发展需求的创新素养。

该阶段学生能够独立或协作完成复杂创新项目，如开发适配移动端与PC端的跨平台应用、设计基于物联网的环境监测系统、利用人工智能工具实现数据智能分析等，具备针对新场景提出创新解决方案的能力，形成“理论扎实、动手娴熟、思维活跃”的综合素养。

3 教学内容设计

以“无人车路径规划”为贯穿始终的核心应用场景，设置“城市救援-野外道路-复杂地形”三关场景，实现从基础建模到综合应用的能力螺旋提升，各关卡聚焦不同核心能力，层层递进关联。

采用螺旋式培养模式，借助虚拟仿真环境，将Python编程基础、路径规划算法及复杂约束问题解决能力分层递进融入教学。通过“认知筑基—实践应用—反思提升”的闭环循环，使学生在每一轮螺旋中都能基于前序积累实现能力跃迁，最终形成“问题建模—算法选型—代码实现—优化创新”的完整能力链。各阶段既独立聚焦又层层衔接，如表1所示。

表 1 分层目标体系

能力层级	知识目标	能力目标	素养目标
基础层 (螺旋1)	掌握编程语言语法、基本数据结构(数组、链表)、简单算法(排序、查找)	能识别明确的计算机可解问题，用基础语法实现指定功能，完成50-100行代码任务	养成规范编码习惯，建立“输入-处理-输出”的基本思维
进阶层 (螺旋2)	掌握复杂数据结构(树、图)、算法优化策略、模块化开发思想	能将模糊问题转化为明确模型，拆分复杂任务并实现，完成200-500行代码项目	培养问题拆解能力，形成“测试-调试-优化”的迭代意识
创新层 (螺旋3)	掌握跨模块整合、第三方库应用、系统性能分析方法	能结合实际需求设计解决方案，完成1000行以上代码的创新项目，具备问题优化能力	树立工程创新思维，培养团队协作与成果迭代素养

3.1 第一螺旋：基础层

(1) 阶段核心目标

这阶段目的是能将无人车“简单路径场景”（如静态无障碍物的两点之间路径）转化为数据结构模型，用基础遍历算法实现路径搜索。如图1所示。

(2) 教学内容与关键技术

城市救援无人车（基础建模关）：场景设定为“城市道路无拥堵，需从救援站到事故点执行紧急救援”。

核心任务：将城市主干道、交叉路口分别抽象为

“边”和“节点”，忽略复杂支路，构建简易城市道路图模型。



图 1 无人车路径规划（第一关）

技术实现：用Python列表与字典组合实现图的邻

接表存储,定义“道路”“路口”基础类,明确节点与边的关联关系,掌握数据结构与现实场景的基础映射方法。

3.2 第二螺旋:进阶层

(1) 阶段核心目标

这阶段目的是野外道路无人车(遍历算法关):场景升级为“野外简易道路,部分路段狭窄仅容单向通行,需探索所有可达区域确认救援范围”。如图2所示。



图2 无人车路径规划(第二关)

(2) 教学内容与关键技术

核心任务:基于城市道路图模型扩展,增加“道路通行方向”属性,通过遍历算法完成区域探索。

技术实现:在第一关图结构基础上,为边添加方向标识;分别实现DFS(深度优先搜索,适合全面探索未知区域)与BFS(广度优先搜索,适合快速定位邻近可达点)两种算法,对比分析“单向道路”对两种遍历结果的影响,输出野外救援的可达区域清单。

算法1:深度优先

深度优先算法的实现

```
def DFS(neighbor, root):
    # 首先判断根节点是否为空节点
    if root != None: #根节点不为空节点,开始搜索
        search_queue = deque() # 创建一个队列
        search_queue.append(root) # 先将根节点添加到队列中
        visited = [] # 搜索过的节点列表
    else:
        print('root is None')
        return -1
    while search_queue: # 循环搜索该队列
        person = search_queue.popleft() # 移除队列最左边的元素
        # 如果该节点没有被搜索过且该节点还有子节点
        if (not person in visited) and (person in neighbor.keys()):
            tmp = neighbor[person] # 获取该节点的所有子节点
            tmp.reverse() # 将获取到的子节点列表反转
            for index in tmp: # 遍历列表,将这些子节点添加到队列最左边
```

```
search_queue.appendleft(index) # 将该节点添加到搜索过的列表内
        visited.append(person)
    return visited
```

算法2:宽度优先

```
def BFS(neighbor, root):
    # 根节点为空的边界处理
    if root is None:
        print('root is None')
        return -1

    search_queue = deque() # 初始化队列(BFS核心数据结构)
    search_queue.append(root) # 根节点入队
    visited = [root] # 记录已访问节点(避免重复遍历),初始包含根节点

    # 队列非空时循环遍历
    while search_queue:
        person = search_queue.popleft() # 取出队首节点(先进先出)

        # 若当前节点有邻接节点,处理其子节点
        if person in neighbor.keys():
            # 遍历当前节点的所有邻接节点(无需反转,按原顺序处理)
            for child_node in neighbor[person]:
                # 未访问过的节点才入队并标记已访问
                if child_node not in visited:
                    visited.append(child_node)

            search_queue.append(child_node) # 子节点追加到队尾(BFS核心区别)

    return visited
```

3.3 第三螺旋:创新层



图3 无人车路径规划(第三关)

(1) 阶段核心目标

这阶段目的是复杂地形物资运送(约束建模关):场景进阶为“山地、沟壑等复杂地形,存在不可通行区域,物资运送需考虑路径坡度限制(坡度>30°不可通行)”。如图3所示。

(2) 教学内容与关键技术

核心任务：构建含约束条件的地形模型，实现带筛选功能的路径搜索。

技术实现：扩展图结构，为节点添加海拔属性，

通过海拔差计算道路坡度；在 BFS/DFS 算法中嵌入“坡度判断”逻辑，自动过滤不符合条件的路径；输出从物资点到受困点的所有可行路径，标注各路径坡度最大值。

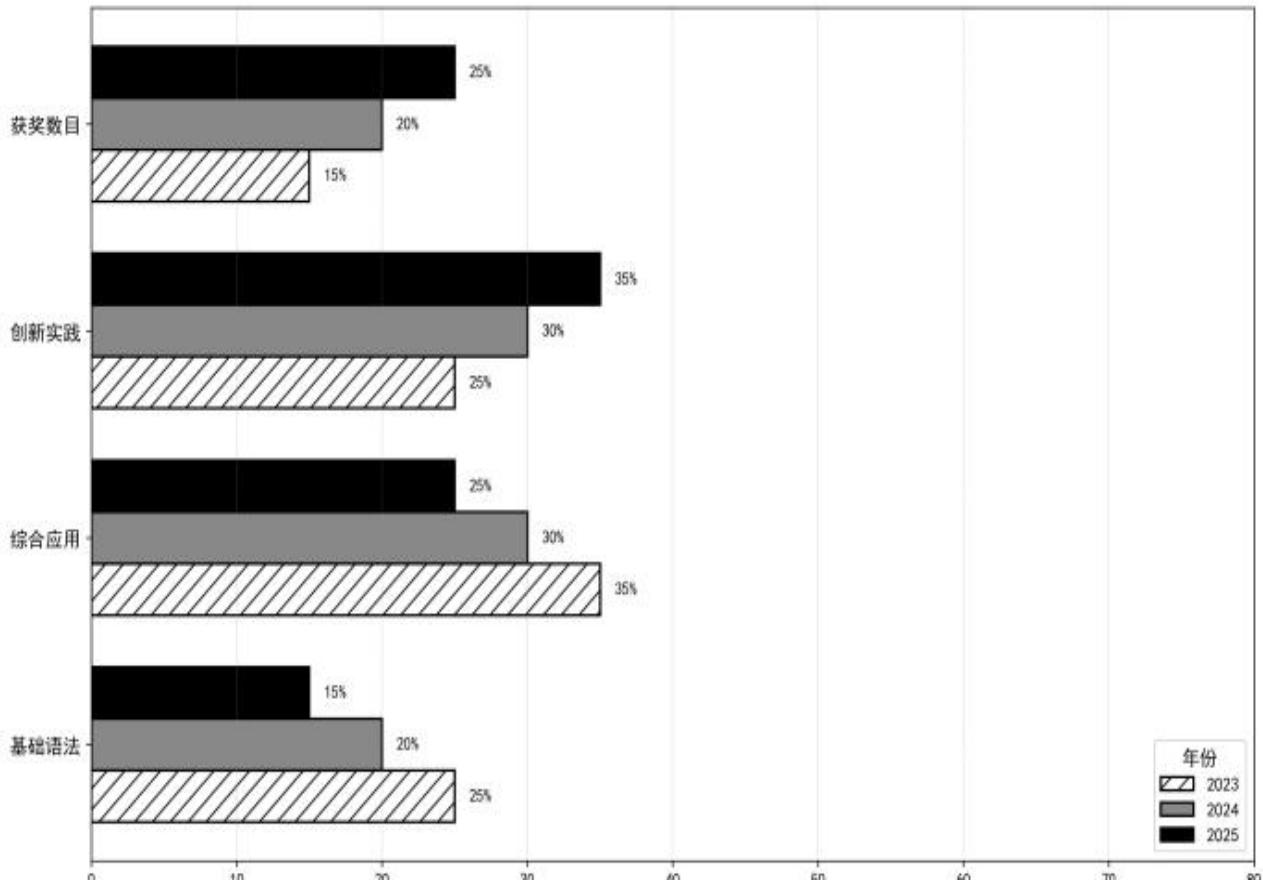


图 4 近 3 年各项成绩占比

4 效果评估

经过近三年的教学实践改革，学生编程能力实现大幅提升：在军队院校计算机能力大赛中，获奖数量位居兄弟院校前列；斩获全国计算机设计大赛、蓝桥杯等国家级赛事奖项 20 余项。与此同时，学生对计算机学科的学习兴趣持续增强，在程序设计基础语法、综合实践及创新实践等维度的表现均取得显著进步。如图 4 所示。

5 结束语

为破解计算机类课程“重理论轻实践”的教学痛点，本文创新采用螺旋式能力进阶教学模式，搭建三轮螺旋循环的递进式教学架构。依托该模式，学生在“理论学习—实操应用—反思迭代”的闭环中夯实知识基础、提升动手能力，最终形成“理论扎实、动手

娴熟、思维活跃”的核心素养。实践验证显示，该教学模式不仅使学生实践能力考核通过率显著提升，更优化了教学流程、提高了课堂效能，为计算机类课程教学改革提供了兼具操作性与创新性的实践方案。

参考文献

[1] 杨阳,郝玉婷,陶丽,等.大数据时代背景下数据分析类课程教学改革与实践探索[J].高等工程教育研究,2023,(05):54-59+116..

[2] 裴壮,田秀霞,李冰雪.知识图谱赋能的面向对象程序设计C++教学改革与实践[J].华东师范大学学报(自然科学版),2024,(05):104-113.

[3] 李晶.工程教育理念下《游戏程序设计》教学改革与实践[J].计算机工程与科学,2019,41(S1):26-30.

[4] 徐涛.基于多维协同融合的计算机网络课程教学模式改革[J].计算机技术与教育学报,2023,11(04):133-137.

[5] 胡文龙.基于CDIO的工科探究式教学改革研究[J].高等工程教育研究,2014,(01):163-168..