

高中学生计算机编程能力培养的阶梯式教学设计探究

杨厚福

江西省奉新县冶城职业学校

摘要: 本文针对高中学生计算机编程能力培养问题,提出阶梯式教学设计方案。分析高中计算机编程教学现状,阐述阶梯式教学设计内涵与优势,详细介绍基础、进阶、拓展提升三阶段设计内容,并通过实例说明实施过程与效果,同时提出保障措施。研究表明,该设计能提升学生编程能力,为高中计算机编程教学改革提供参考。

关键词: 高中学生; 计算机编程能力; 阶梯式教学设计

【DOI】 10.12252/j.issn.2096-627X.2025.08.155

引言

在数字化时代,计算机编程能力已成为一项关键技能。对于高中学生而言,掌握计算机编程不仅能提升其科技素养,还能为其未来的学术发展和职业选择奠定基础。然而,高中学生在学习计算机编程过程中面临诸多挑战,传统教学模式难以满足学生多样化的学习需求。因此,探索一种有效的教学方法来培养高中学生的计算机编程能力具有重要的现实意义。阶梯式教学设计作为一种创新的教学理念,强调根据学生的认知水平和学习能力,将教学内容划分为不同层次,逐步引导学生提升编程能力,为解决高中计算机编程教学难题提供了新的思路。

一、高中计算机编程教学现状分析

(一) 学生基础与学习能力差异显著

高中学生来源不同,其计算机基础和学习能力存在明显差异。部分学生在初中阶段已接触过计算机基础知识,对编程有一定的认知和兴趣;而另一部分学生可能对计算机操作还不够熟练,缺乏基本的编程思维。这种较大的差异给统一教学带来了困难,传统的“一刀切”教学方式难以满足不同层次学生的学习需求,导致基础薄弱的学生跟不上教学进度,逐渐失去学习信心,而基础较好的学生又觉得教学内容缺乏挑战性,无法充分发挥其潜力。

(二) 教学内容与方法有待优化

当前高中计算机编程教学内容在一定程度上存在理论与实践结合不紧密的问题。教学中过于注重编程语言的语法讲解,而忽视了编程思维和实际应用能力的培养。学生往往只是机械地记忆语法规则,在面对实际编程问题时,却不知如何运用所学知识进行分析和解决。在教学方法上,多采用传统的讲授式教学,教师在课堂上占据主导地位,学生被动接受知识。这种教学方法缺乏互动性和趣味性,难以激发学生的学习兴趣 and 主动性,学生在学习过程中容易感到枯燥乏味,对编程学习产生抵触情绪。

(三) 实践教学环节有待加强

计算机编程是一门实践性很强的学科,实践教学环节对于学生编程能力的提升至关重要。然而,在实际教学中,高中学校的实践教学环节存在诸多不足。一方面,实践教学设备和软件更新不及时,无法满足当前编程教学的需求。部分学校的机房设备陈旧,运行速度缓慢,一些新的编程软件无法正常安装和运行,限制了学生的实践操作。另一方面,实践教学的项目设计缺乏真实性和综合性,往往只是简单的案例练习,与实际应用场景相差甚远。学生在实践过程中无法真正体验到编程在实际问题中的应用,难以将所学知识融会贯通,解决实际问题的能力得不到有效锻炼。

二、阶梯式教学设计的内涵与优势

(一) 内涵阐述

阶梯式教学设计是一种遵循学生认知规律和学习能力发展的教学模式。它将计算机编程教学内容按照从易到难、从基础到复杂的顺序,划分为多个层次的教学模块,每个模块都有明确的教学目标和任务。学生在完成低层次模块的学习后,逐步上升到更高层次的学习,如同攀登阶梯一般,每一步都建立在已有的知识和技能基础之上。在基础阶段,注重教授编程的基本概念、语法和简单的编程结构,帮助学生建立编程的初步认知;进阶阶段则深入讲解编程的核心知识和技术,培养学生解决中等难度问题的能力;拓展提升阶段引导学生综合运用所学知识,进行复杂项目的开发,提升学生的创新能力和实际应用能力。通过这种分层递进的教学方式,使不同基础和学习能力的学生都能在各自的起点上得到发展和提高。

(二) 优势分析

阶梯式教学设计充分考虑了学生的个体差异,为不同层次的学生提供了适合他们的学习路径。基础薄弱的学生可以在基础阶段扎实掌握基础知识和技能,逐步建立学习信心;基础较好、学习能力较强的学生则可以在更高层次的学习中挑战自我,不断拓展知识和技能的深

度与广度。这种个性化的教学模式能够更好地激发学生的学习兴趣 and 主动性,使每个学生都能在编程学习中获得成就感。同时,将教学内容划分为不同层次的模块,使知识呈现出系统性和逻辑性。学生在逐步攀登“阶梯”的过程中,能够清晰地看到知识之间的联系和递进关系,从而更好地理解和掌握编程知识体系。通过不断地将新知识与已有的知识进行整合,学生能够构建起更加稳固、完整的知识结构,为解决复杂的编程问题奠定坚实的基础。

三、高中学生计算机编程能力培养的阶梯式教学设计

(一) 基础阶段设计

1. 教学目标:使学生了解计算机编程的基本概念和编程环境,掌握编程语言的基本语法、数据类型、运算符和控制结构等基础知识,能够编写简单的程序实现基本功能,培养学生的编程兴趣和初步的编程思维。

2. 教学内容:编程基础概念:介绍计算机编程的定义、作用和发展历程,让学生了解编程在现代社会中的重要地位。讲解编程语言的分类和特点,引导学生选择适合高中阶段学习的编程语言,如 Python 语言。

Python 语法基础:详细讲解 Python 语言的基本语法规则,包括变量的定义与使用、数据类型(整型、浮点型、字符串、布尔型等)、运算符(算术运算符、比较运算符、逻辑运算符等)、表达式和语句的书写规范。通过大量简单的示例代码,让学生熟悉语法的应用,例如计算两个数的和、判断一个数的奇偶性等。

控制结构:教授顺序结构、选择结构(if 语句)和循环结构(for 循环、while 循环)的概念和使用方法。通过实际案例,如根据学生成绩判断等级、计算 1 到 100 的累加和等,让学生掌握不同控制结构在程序中的应用,学会根据实际问题选择合适的控制结构来实现程序功能。

3. 教学方法:采用讲授法与演示法相结合,教师在课堂上详细讲解知识点,并通过多媒体教学软件演示代码的编写和运行过程,让学生直观地感受编程的实现过程。同时,安排大量的课堂练习时间,让学生及时巩固所学知识,教师在学生练习过程中进行巡视指导,及时纠正学生的错误。

(二) 进阶阶段设计

1. 教学目标:深化学生对编程语言的理解,掌握函数、数组、对象等重要编程概念和技术,能够运用所学知识编写具有一定功能复杂度的程序,培养学生的模块化编程思维和问题解决能力。

2. 教学内容:函数:讲解函数的定义、参数传递和返回值的概念,让学生学会如何定义和调用函数。通过实际案例,如编写一个计算圆面积的函数、一个判断素数的函数等,使学生理解函数在代码复用和程序模块化中的重要作用。

数组与数据结构:介绍数组的概念和使用方法,包括一维数组和二维数组的定义、初始化和访问。讲解常见的数据结构,如列表、元组、字典等,让学生了解不同数据结构的特点和适用场景。通过实际项目,如学生成绩管理系统(使用数组存储学生成绩并进行统计分析),让学生掌握数组和数据结构在实际编程中的应用。

3. 教学方法:采用项目驱动教学法,教师给定一些具有一定复杂度的项目任务,如小型管理系统、简单游戏开发等。学生在完成项目的过程中,综合运用所学的函数、数组、面向对象编程等知识,自主分析问题、设计解决方案并编写代码实现。教师在项目实施过程中提供指导和支持,帮助学生解决遇到的问题,引导学生逐步提高编程能力。

(三) 拓展提升阶段设计

1. 教学目标:培养学生的综合编程能力和创新能力,使学生能够独立完成具有一定规模和复杂度的实际项目开发,了解软件开发的流程和规范,提升学生的科技素养和团队协作能力,为学生未来的升学或职业发展打下坚实的基础。

2. 教学内容:综合项目开发:组织学生进行大型项目的开发,项目主题可以结合实际应用场景,如校园信息管理系统、移动应用开发、数据分析与可视化项目等。学生在项目中需要综合运用所学的各种编程知识和技术,进行系统设计、数据库设计、界面设计和代码编写等工作。通过实际项目的开发,让学生体验软件开发的全过程,提高学生解决实际问题的能力和项目管理能力。

软件开发流程与规范:讲解软件开发生命周期,包括需求分析、设计、编码、测试、维护等阶段,让学生了解每个阶段的任务和重要性。介绍代码编写规范、注释规范、文档撰写规范等,培养学生良好的编程习惯和职业素养。在项目开发过程中,要求学生严格按照软件开发流程和规范进行操作,提高项目的质量和可维护性。

3. 教学方法:采用小组合作学习法和自主探究学习法相结合。学生组成项目小组,分工协作完成项目开发任务。在小组合作过程中,培养学生的团队协作能力和沟通能力。同时,鼓励学生自主查阅资料、学习前沿知识,在项目中进行创新实践。教师作为项目导师,为学生提供宏观指导和技术支持,引导学生在实践中不断提升综合能力。

四、阶梯式教学设计的实施过程

（一）学生分层与目标设定

在课程开始前，通过对学生的计算机基础、学习能力和学习兴趣等方面进行测试和调查，将学生分为基础层、提高层和拓展层三个层次。针对不同层次的学生，制定相应的教学目标和学习计划。基础层学生以掌握基础知识和基本技能为主要目标，提高层学生在基础上进一步提升编程能力和解决问题的能力，拓展层学生则注重培养创新能力和综合应用能力。

（二）教学资源准备

根据不同层次的教学内容和教学方法，准备相应的教学资源。包括教材、课件、案例、项目资料、在线学习平台等。为基础层学生提供详细、易懂的基础教程和大量的基础练习案例；为提高层学生准备具有一定难度和综合性的项目案例及相关技术文档；为拓展层学生提供前沿技术资料和行业案例分析。

（三）教学组织与实施

按照阶梯式教学设计的三个阶段，分层次进行教学。在基础阶段，采用集中授课的方式，确保全体学生都能掌握基础知识和技能。随着教学层次的提升，逐渐增加小组合作学习和自主探究学习的比重。在项目实践环节，根据学生层次分配不同难度的项目任务，学生在教师的指导下进行项目开发。定期组织课堂讨论和经验分享活动，让不同层次的学生相互交流心得体会，促进共同进步。

（四）教学评价与反馈

建立多元化的教学评价体系，对学生的学习过程和学习成果进行全面评价。评价内容包括课堂表现、作业完成情况、项目实践成果、考试成绩等。针对不同层次的学生，制定不同的评价标准，注重对学生的进步和努力进行肯定和鼓励。及时收集学生的反馈意见，根据学生的学习情况和反馈，调整教学内容和教学方法，确保教学效果的不不断提升。

五、阶梯式教学设计实施的保障措施

（一）教师专业能力提升

教师是阶梯式教学设计实施的关键因素。学校应加强对计算机编程教师的专业培训，定期组织教师参加各类编程培训课程、学术研讨会和教育教学实践活动，让教师及时了解教育最新动态和技术发展趋势，不断更新教学理念和教学方法，提升自身的编程能力和教学经验。鼓励教师开展教学研究和教学改革实践，探索适合高中学生的编程教学模式和方法，提高教学质量。

（二）教学资源建设

丰富的教学资源是阶梯式教学设计实施的重要保障。学校应加大对教学资源建设的投入，购置先进的教学设

备和软件，确保机房设备能够满足学生的编程实践需求。建设完善的在线学习平台，提供丰富的教学视频、电子教材、在线测试、讨论区等学习资源，方便学生随时随地进行学习和交流。组织教师编写适合阶梯式教学的校本教材和教学案例集，根据不同层次学生的特点和需求，精心设计教学内容和案例，使教材和案例更具针对性和实用性。

（三）教学评价体系完善

建立科学合理的教学评价体系是阶梯式教学设计有效实施的重要支撑。教学评价应注重过程性评价与终结性评价相结合，全面、客观地评价学生的学习过程和学习成果。过程性评价包括课堂表现、作业完成情况、小组合作表现、项目实践过程中的参与度和贡献等方面，及时反馈学生的学习进展和存在的问题，鼓励学生积极参与学习过程。终结性评价采用多元化的考核方式，除了传统的理论考试外，增加实践操作考核、项目成果展示与答辩等环节，综合评价学生的知识掌握程度、编程能力和创新能力。针对不同层次的学生，制定差异化的评价标准，关注学生在原有基础上的进步和发展，充分发挥教学评价的激励和导向作用。

结语

阶梯式教学设计为高中学生计算机编程能力的培养提供了一种行之有效的教学模式。通过将教学内容分层递进，满足了不同基础和学习能力学生的个性化需求，促进了学生知识的系统构建和编程能力的逐步提升。实践证明，阶梯式教学设计在高中计算机编程教学中具有显著的优势和良好的应用前景，能够为培养具有较高科技素养的创新型人才奠定坚实的基础。未来，随着信息技术的不断发展和教育教学改革的深入推进，阶梯式教学设计还需要不断优化和完善，以更好地适应新的教学要求和学生的发展需求。

参考文献

- [1] 蔡兴. 高中计算机编程教学与中学生创新能力培养的几点思考 [J]. 电脑迷, 2018 (20): 118.
- [2] 张雨莹. 高中生如何提高计算机编程能力 [J]. 消费导刊, 2018 (43): 115.
- [3] 王曙亮. 对高中生提高计算机编程能力的体会 [J]. 数码设计 (上), 2019 (1): 25-26.
- [4] 王汝丹. 刍议高中信息技术教学中提升学生编程能力的有效方法 [J]. 软件 (教育现代化) (电子版), 2019 (12): 167.
- [5] 杨芷若. 对高中生提高计算机编程能力的体会 [J]. 数字通信世界, 2018 (3): 270, 282.