

基于 UbD 理论的单元逆向教学设计——以《编程计算》为例

孙芳轩

聊城大学 传媒技术学院, 中国·山东 聊城 252059

摘要: UbD (Understanding by Design) 单元教学设计方法通过融合大概念、基本问题以及评价与反馈机制等要素,旨在设计高效的教学方案。文章基于 UbD 理论,对高中信息技术《数据与计算》单元中《编程计算》进行了教学设计。UbD 强调以“理解”为核心,以“逆向设计”为路径,通过“确立预期结果—确定评估证据—设计学习体验”三个阶段实现教学目标。本文呈现了逆向设计视角下的教学探索与实践,旨在将具体知识背后的大概念外显,并推动信息课程核心素养的落实。

关键词: UbD 理论; 教学设计; 编程计算

Reverse Unit Instructional Design Based on the UbD Theory——Taking Programming and Computing as an Example

Sun Fangxuan

School of Media Technology, Liaocheng University, China Shandong Liaocheng 252059

Abstract: The Understanding by Design (UbD) unit instructional design method integrates elements such as big ideas, essential questions, and evaluation and feedback mechanisms, aiming to design effective teaching plans. Based on the UbD theory, this paper conducts instructional design for Programming and Computing (a section in the "Data and Computing" unit of Senior High School Information Technology curriculum). UbD emphasizes "understanding" as the core and "reverse design" as the approach, and achieves teaching goals through three stages: identifying desired results, determining assessment evidence, and designing learning experiences. This paper presents the teaching exploration and practice from the perspective of reverse design, with the purpose of externalizing the big ideas behind specific knowledge and promoting the implementation of core competencies in information courses.

Keywords: UbD Theory; Instructional design; Programming and computing

1 UbD 理论概述

传统课堂中,学生更多依赖记忆和模仿来学习知识,这样的教学方式存在两大问题:一是记忆型知识容易遗忘,二是学生缺乏深入理解,难以将所学知识融会贯通,进而无法将其有效应用于现实生活中的问题解决^[1]。

追求“理解”的逆向教学设计 (Understanding by Design, UbD) 要求教师在教学设计前,首先关注预期的学习结果,强调教学评价应优先于教学活动,并从学习结果出发进行“以终为始”的逆向思考^[2]。逆向教学设计强调以核心大概念和关键表现性任务为基础,帮助学生更有效地掌握知识,进而实现预定的学习目标。在教学实践中,教师应根据课程标准的具体要求,实施单元逆向教学设计,从而提升学生在信息科技课程中的核心素养,并激发他们

的深度学习。

UbD 逆向教学设计的核心特征是“理解”和“逆向”^[3]。“理解”包括解释、阐明、应用、洞察、神入和自知六个方面,其关键在于深入理解和构建大概念,将其转化为一系列启发性基本问题,并通过结构化引导,使得“理解”过程变得系统化,从而促进大概念的形成与内化。“逆向”设计的核心特征在于构建“结果”“证据”和“活动”之间的逆向因果关系,这三者相互作用,形成了一个从结果出发的设计流程。

根据 UbD 理论,教学设计者需在开始时明确预期学习成果,并根据学习目标的表现性要求规划课程内容。该设计流程可分为三个阶段:确立预期学习结果、确定评估标准、设计教学活动与学习体验。

2 前端分析

2.1 选题

编程能力,简单来说,就是编写、测试、调试和维护计算机程序的能力。它不仅包括算法设计、数据结构、编程语言等基础知识,还涵盖了问题分析、创新思维和团队合作等软技能。在信息时代,编程能力已成为衡量一个人综合素质的重要指标。编程不仅能提高学习者的逻辑思维和解决问题的能力,还能培养他们的计算思维和创新能力。

要系统地培养学习者的编程能力,必须从基础知识入手。因此,本设计选择了 2019 版教育科学出版社出版的信息技术必修 1《数据与计算》第二单元《编程计算》的教学内容,设计了相关编程项目进行学习。该项目旨在帮助学习者掌握编程计算的基本知识,理解顺序、分支、循环等基本结构及其应用场景,并学习函数的定义与调用,为编写更复杂的程序打下坚实的基础。

2.2 教材内容分析

《编程计算》是信息技术课程中的一个重要单元,属于高中信息技术必修课程。本单元的教材旨在帮助学生建立编程的基本认知,理解算法的重要性,并掌握至少一种程序设计语言的基础结构。教材内容涵盖算法设计、程序设计语言的基本概念及程序结构。基于这些基本内容,本单元通过“手机评分”项目展开教学,帮助学生理解并应用所学知识。

2.3 学习者分析

高一年级的学生正处于青春期,认知和逻辑思维能力相对成熟,但对编程概念和技能可能感到陌生。此阶段的学生通常对新事物充满好奇,愿意接受挑战,并具备一定的自学能力。虽然他们已有一定的逻辑思维基础,能够理解基本的算法概念和程序设计原理,但部分学生可能没有编程经验,需要从基础开始教学,逐步建立编程技能。此外,学生可能对编程存在恐惧感,认为编程难以掌握,因此教师需要通过鼓励和正面反馈帮助他们建立信心。

学习者需求分析:学生需要掌握算法的基本概念、流程图的绘制方法,以及至少一种程序设计语言的基本语法和编程技巧。

2.4 学习目标

本单元学习目标如下:①理解算法的概念,并能够用自然语言和流程图描述算法。②掌握一种程序设计语言的基本语法和编程技巧。③完善程序的部分代码,并能够理解其含义。④能够运用程序设计语言解决实际问题。

2.5 大概念

单元大概念包括通过学习程序设计语言,理解计算机解决问题的基本方法,培养学生的计算思维和数字化学习与创新能力。

核心概念如下:①算法,解决问题的步骤和方法,是计算机程序设计的核心。②编程语言,人与计算机交流的工具,用于编写计算机程序。③计算思维,一种解决问题的思维方式,强调逻辑性、抽象性和系统性。④数字化学习与创新,利用数字化工具进行学习和创新的能力。

3 基于 UbD 理论下的教学设计

3.1 阶段一:确定预期结果

预期结果指的是常提到的预期成果、成就目标或表现标准,强调将教学重点从输入转向输出:学生在学习结束后,应该能够知道、理解和应用相关内容,并在表现和作品中体现出来。

在阶段一,教师需要参考已发布的内容标准,检查课程预期结果,并明确学生应掌握的知识、理解的内容和能做的事情。重点是确定哪些内容值得理解,哪些是期望的持久理解。因此,文章将从课程标准、预期理解、基本问题和知识与技能四个方面进行阐述。

3.1.1 课程标准

《普通高中信息技术课程标准(2017 年版 2020 年修订)》^[4]中提出:通过本模块的学习,学生应认识到数据在信息社会中的重要价值,合理处理与应用数据,掌握算法与程序设计的基本知识,运用数字化工具解决生活与学习中的问题,认识到人工智能在信息社会中日益重要的作用,逐步成为信息社会的积极参与者。根据课程标准,结合教材内容,确定本节课的整体学习目标。

课标中学业要求如下:①依据解决问题的需要,设计并表示简单算法;②掌握一种程序设计语言的基本知识,利用它实现简单算法,解决实际问题。

预期的学习结果包括:①理解算法,能够根据算法绘制流程图;②了解不同结构的应用场景;③阐明函数定义;④理解并应用相应代码。

3.1.2 预期理解

《编程计算》模块是必修一的第二模块,是后续模块学习的基础。高一年级的学生处于对编程入门的状态,对于达到高阶编程能力仍有一定的距离。

根据对学习者和学习内容的分析,确定预期理解:①学生将理解单元的基础知识内容(算法、流程图、能够用自然语言和流程图描述算法);②使用流程图的三种基本

结构解决算法中的需求；③了解循环结构的两种类型的执行方式和特点不同；④阐明函数的定义、应用函数的调用，也可以使用标准库的程序模块。

3.1.3 基本问题

基本问题的提出是为了强调大概念，它是教学设计的核心，学生将围绕大概念展开学习。通过提出基本问题，教师鼓励教学设计者避免知识灌输，转向真实的探究——讨论、反思、解决问题、研究和辩论，这些是深入理解基本概念的必要条件。围绕教学目标和预期理解，结合教学内容，确立本节课的基本问题。

学生将要理解的内容有：①单元的基础知识（算法、流程图、用自然语言和流程图描述算法）；②流程图的三种基本结构及其应用；③循环结构的两种执行方式及特点；④函数的定义及应用。

本节课的基本问题有：①什么是算法、流程图，如何用自然语言和流程图描述算法？②日常生活中是否可以用流程图表达的问题？③如何选择适合情境的流程图结构？④在给定程序中需要填补哪些语句？⑤如何定义和应用函数解决更复杂问题？

3.1.4 知识与技能

通过本节课设计的项目探究与学习，学习者将获得以下重要知识和技能：

学习者将了解：①关键术语：算法、流程图、分支、循环、函数；②用自然语言和流程图描述算法；③流程图的基本结构及应用场景；④循环的两种形式；⑤函数的定义。

学习者将能够：①使用语句填补程序；②调用函数；③将知识迁移到其他场景。通过以上分析，学习者将掌握

表1 确定合适的评估证据

1.表现性任务
(1) 学生绘制三种基本结构的流程图，并写出其特点；
(2) 学生补全并完善代码部分，运行并实现基本功能；
(3) 学生理解每部分代码的功能，并能解释其含义；
(4) 代码结构清晰、完整；
(5) 定义和调用函数，尝试用更简洁的方式完成程序；
(6) 能够将代码迁移到其他场景。
2.其他证据
小测验：给出案例，学习者判断使用哪种类型流程图、完善代码。
项目展示：解释代码的含义以及迁移应用。
3.学生的自我评价和反馈：
(1) 学生对自己在编程能力、问题解决能力、合作能力等方面进行评价；
(2) 反思该项目如何应用到生活中的实际场景；
(3) 同伴互评：学生相互评价程序功能、界面设计、代码规范等；
(4) 教师评价：教师对学生的学习成果进行评估，包括程序功能、界面设计、代码规范及学生表现。

表2 所设计的学习体验

课时1：项目介绍和需求分析
导入：展示手机评分的案例，引起学习者兴趣
项目介绍：介绍手机评分项目的侧重的方面和功能。从五个方面打分，利用排序算法，根据流程图的三种结构对应的代码部分完善项目。
需求分析：引导学生分析完善手机评分项目的需求，解释其功能如何实现，需要课本单元内的哪些知识
小组讨论：学生分组讨论，并确定小组完善手机评分项目所需功能
教师小结：教师讲解项目所需要的基础知识。根据流程图相应的特点，学习者画出相应的结构。学习编程语言的基础部分，赋值语句等
课时2：Python编程语言学习
编程语言：根据学习者的基础和学习目标以及教材内容，在课本的案例中选用代码
基础知识学习：学习流程图对用的代码的基本语法，尤其是循环部分
进阶知识学习：学习函数定义、使用；函数调用；函数库等。同时在课上下发小测试题，学习者独立完成后，教师讲解，同学互评，教师回收测试题，作为形成性评价的证据之一
课时3：项目开发
项目规划：学习者根据功能要求，小组合作寻找相应代码
编码实现：学习者使用编程语言完善手机评分项目。并尝试使用不同的代码来表达
测试与调试：学习者测试自己的项目，并修复发现的问题
课时4：项目展示与反思
项目展示：学习者向老师和同学演示自己的项目，并解释相应代码功能和原理
代码审查：教师对学习者的代码进行审查，评估其代码质量和技术水平，作为评估的一部分
反思与总结：学习者反思自己的学习过程和项目开发过程，并总结经验教训

编程计算的基本知识与技能。

3.2 阶段二：确定评估证据

评估，即衡量预期结果将要达到的程度以及预期结果已经达到的程度^[6]。评估是一个综合性术语，需运用多种方法收集相关证据。

教师如何判断学生是否实现预期学习成果？哪些证据能够反映学生对知识的理解与掌握？在逆向设计理念指导下，教师应基于收集的评估证据进行设计，而非单纯围绕教学内容或活动安排。这种方法鼓励教师在规划教学单元之前，采取“评估者视角”，思考如何确认学生是否达到了预期理解。

为全面收集证据，确保学生达到目标，教师将设计表现性任务，并通过其他证据形式及学生的自我评估与反馈等方式，评估学生的学习成果（见表1）。

3.3 阶段三：设计学习体验和教学

为了让学生有效学习并实现预期成果，他们需要掌握哪些知识（如事实、概念、原理）和技能（如过程、步骤、策略）？哪些活动能够帮助学生获取这些知识和技能？基于表现性目标，教师应教授哪些知识点，指导学生进行哪些实践，如何以最适当的方式教学？为了达成这些目标，哪些材料和资源最为适宜？

学习者对编程的理解是逐步深入的，从基础概念、结构及基础代码开始，逐步扩展到编写程序、定义与调用函数。基于此，设计了四个课时的教学活动（见表2）。

4 UbD 理论下的教学设计反思

4.1 教学设计应以目标为导向

教学设计应以具体的学习结果为目标。例如，所选择的教学案例在第一阶段明确教学目标，以项目式教学为主，结合小组合作交流，始终以完善项目为目标，同时深入学习单元的基本问题和大概概念。目标驱动的设计决定了评估证据的性质，并为教学方法和学习体验提供指导。

4.2 教学设计要明确评估内容

教学设计过程中，应预先明确评估内容，这些内容是

表现性任务和证据的关键。例如，通过观察学习者的表现、收集数据，以及评估小组项目中的代码清晰度等，教师可以评估学生的学习成果。这不仅明确教师期望的学习目标，也有助于区分教学内容的优先级，提升课堂教学效率。

4.3 教学设计应筛选教学资源

教学设计过程中，应灵活挑选教学资源，利用这些资源设计任务和体验活动，促进学生的协作学习。教师还可借助资源和技术，帮助学生克服学习障碍，深化解理解。

4.4 教学设计要创设学习活动

在制定教学设计时，教师应关注根据预期的教学目标和评估内容来设计相应的学习体验活动。这是因为精心设计的学习活动不仅让学生享受学习过程，还能促使他们参与有深度的智力挑战。尤其是那些具有挑战性的学习活动，更能有效地培养学生的探究技能，促进他们的计算思维和科学探究精神的发展。

参考文献：

- [1] 加德纳·霍华德. 多元智能[M]. 北京：新华出版社，1999:10.
 - [2] Wiggins, Grant,McTighe, Jay.追求理解的教学设计[M].华东师范大学出版社，2017.
 - [3] 葛丽婷，施梦媛，于国文. 基于 UbD 理论的单元教学设计——以平面解析几何为例[J]. 数学教育学报，2020,29(05):25-31.
 - [4] 中华人民共和国教育部. 普通高中信息技术课程标准（2017 年版 2020 年修订）. 北京：人民教育出版社，2018.
 - [5] 刘徽.“大概概念”视角下的单元整体教学构型——兼论素养导向的课堂变革[J]. 教育研究，2020,41(06):64-77.
 - [6] 倪胜军，付绍武，艾进达. 逆向、整体、可操作：UbD 理论视角下化学单元教学设计——以九年级“化学方程式”为例[J]. 化学教学，2021 (12): 48 - 51 +64.
- 作者简介：孙芳轩（1999-），女，山东人，聊城大学传媒技术学院硕士研究生。