Chapter 3
Common Challenges in Coding Education: A Cognitive Load Theory Perspective

Fırat Hayyam Sabuncu
https://orcid.org/0000-0002-6324-9386
Ege University, Turkey

Onur Dönmez
https://orcid.org/0000-0001-5200-1468
Ege University, Turkey

ABSTRACT

This section analyzes coding education challenges through Cognitive Load Theory (CLT), which identifies working memory limitations in processing complex coding concepts. We examine three load types: intrinsic (task complexity), extraneous (poor instructional design), and germane (schema construction). CLT-based solutions include sequenced instruction, worked examples, and multimodal learning to optimize cognitive load distribution. Findings demonstrate these strategies improve knowledge retention and problem-solving transfer. The research highlights educators' role as cognitive designers and emphasizes adaptive scaffolding for diverse learners. By implementing CLT principles, coding education can become more effective and inclusive while reducing cognitive overload.

DOI: 10.4018/979-8-3373-2660-3.ch003

INTRODUCTION

Coding (more broadly programming) is the process of writing instructions in a programming language that a computer can understand and execute. It involves translating human logic into algorithms which are precise, step-by-step commands to create software, apps, websites, games, and more. Coding has been a key competency in the current digital age, empowering learners with problem-solving skills, logical reasoning, and computation skills (Cakir et al., 2021; Garcia et al., 2022; Wing, 2006). Recognizing its importance, educational institutions and online platforms worldwide have increasingly integrated computer science and coding into curricula to prepare learners for a rapidly evolving, technology-driven future (Garcia et al., 2022; Hu, 2024; Oyetade et al., 2025).

At this point, it is helpful to clarify a terminological distinction that is often blurred in both educational discourse and everyday usage. Within the realm of professional business, coding and programming are conceptually similar yet not synonymous. While they are often used interchangeably in informal settings, they differ in scope (Girardin & Courtney, 2023; "indeed", 2025). Coding typically refers to the act of writing code in a programming language, with an emphasis on syntax (Menon, 2025). In contrast, programming encompasses the entire software development process, including problem-solving, design, testing, and optimization (Felleisen et al., 2018). Thus, coding is a part of programming, not its entirety (see Figure 1). Although the distinction is significant in technical contexts, it is often overlooked in popular or educational discourse (Corradini et al., 2018). Given the interchangeable use of the terms in educational discourse, this section includes studies that employ either or both coding and programming for terminological inclusivity and comprehensiveness purposes.