Chapter 1
Pedagogical Innovations for the Modern Computer Science Classroom:
50 Teaching Tips and Practical Strategies for Computing Education

Manuel B. Garcia
https://orcid.org/0000-0003-2615-422X
FEU Institute of Technology, Philippines

Ari Happonen
https://orcid.org/0000-0003-0744-1776
LUT University, Finland & Taylor's University, Malaysia

Friday Joseph Agbo
https://orcid.org/0000-0002-9171-7175
Willamette University, USA

Michael Agyemang Adarkwah
Friedrich Schiller University, Germany

Ioannis Kazanidis
Democritus University of Thrace, Greece

Thomas K. F. Chiu
The Chinese University of Hong Kong, Hong Kong

Robertas Damaševičius
https://orcid.org/0000-0001-9990-1084
Vytautas Magnus University, Lithuania

Serhiy O. Semerikov
https://orcid.org/0000-0003-0789-0272
Kryvyi Rih State Pedagogical University, Ukraine

ABSTRACT

Computer science education is undergoing a global transformation as digital technologies continue to redefine how knowledge is created, shared, and applied. Despite this evolution, many teaching practices in computing remain rooted in

DOI: 10.4018/979-8-3373-6546-6.ch001

instructor-centered models that prioritize syntax and correctness over creativity, collaboration, and reflection. This chapter addresses this gap by presenting pedagogical innovations for the modern computer science classroom. Drawing on contemporary research in computing education and learning theories, it identifies ten interrelated themes and fifty practical strategies for effective instruction. Each theme is grounded in theory, contextualized through classroom examples, and supported by current literature. Collectively, these strategies advocate a shift from transmission-based teaching toward inquiry-driven learning that cultivates computational thinkers, creative problem solvers, and socially responsible innovators. The chapter concludes by discussing the implications of these pedagogical approaches for the future of computer science education.

INTRODUCTION

Computer science is the study of how information can be represented, processed, and transformed using algorithms and computational systems. Its principles underpin the development of intelligent systems that augment human decision-making and expand the frontiers of knowledge (Bozkurt et al., 2024; Gantalao et al., 2025). It holds profound significance in modern society because it shapes the infrastructure of communication, commerce, governance, and innovation (Gill et al., 2024; Tegegn, 2024). As societies become increasingly dependent on data and automation, mastery of computational principles ensures that learners remain competent, ethical, and reflective users of technology. Within educational contexts, computer science functions as both a discipline and a catalyst for interdisciplinary integration that connects mathematical reasoning, scientific inquiry, and humanistic understanding through computational logic (Tariq et al., 2025; Yeni et al., 2024). The study of algorithms and programming enhances precision of thought, persistence in experimentation, and innovation through design. With computer science education developing digital literacy comparable in importance to reading, writing, and numeracy, there is a need to strengthen pedagogies that promote conceptual understanding, creativity, and inclusivity (Liu et al., 2024; Srinivasan et al., 2025).

Computer science education has evolved through a dynamic interplay of pedagogical paradigms that reflect shifting views on how learners acquire computational knowledge and problem-solving skills. Early instruction emphasized procedural fluency and syntax mastery, often framed through behaviorist or transmission-based models. Over time, research in cognitive and constructivist learning prompted a transition toward inquiry-driven and project-based pedagogies that highlight the learner's role in constructing understanding through authentic tasks and collaboration (Garcia, 2023; Tubino et al., 2020). Contemporary frameworks increasingly