UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Computing Education

2627-2-F1802Q119

Aims

Knowledge and understanding

Students acquire advanced knowledge of the epistemological foundations of computer science as a school subject, of the main learning theories applied to CS teaching (constructivism, constructionism, peer instruction), and of national and international curriculum frameworks (Italian National Guidelines, CSTA Standards, DigComp).

They understand the distinction between computer science as a discipline and the mere use of digital tools, and can situate computational thinking within the international CS Education research debate.

Applying knowledge and understanding

Students are able to apply theoretical knowledge to the concrete design of instructional sequences for secondary school. They can use operational frameworks such as backward design (Wiggins & McTighe) and the revised Bloom's taxonomy for computing to build coherent, assessable teaching units.

They can select and use appropriate learning tools and environments (unplugged activities, Scratch, Python, PRIMM, robotics) according to the school level and learning objectives.

Making judgements

Students are able to critically evaluate teaching methodologies, tools, and resources for CS education, justifying their choices in relation to the school context, student level, and curriculum objectives.

They can identify and address common misconceptions in the learning of programming and data structures, adapting teaching strategies accordingly. They develop reflective practice through learning journals and structured post-lesson feedback.

Communication skills

Students can communicate computing content clearly, effectively, and in a manner appropriate to the audience (lower and upper secondary school students). Through micro-teaching, they demonstrate the ability to deliver a structured lesson, manage time, respond to student questions, and adapt their communication to the context.

They can give and receive structured peer feedback using shared observation grids, and can write reflective reports on their own teaching practice.

Learning skills

Students develop the capacity to independently keep up to date with CS Education, a rapidly evolving field, through the critical reading of research articles, participation in professional communities, and field observation in secondary schools.

They acquire the conceptual and methodological tools to pursue continuous professional development as computer science teachers, including in relation to Italian teacher qualification pathways (TFA and D.Lgs. 36/2022).

Contents

Knowledge and Understanding
• Epistemological foundations of computer science as a school subject
• Key learning theories applied to computer science (constructivism, constructionism)
• National and international curriculum frameworks
Skills and Abilities
• Design teaching units with coherent objectives, methods, and assessment criteria
• Select and use tools and methodologies (unplugged, Scratch, Python, PRIMM) according to the level
• Conduct micro-lessons and receive structured peer feedback

Detailed program

Computer Science as a School Subject
What is computer science? Why teach it? History and identity of the discipline, the difference between computer science and digital literacy, computational thinking and its limits as an umbrella concept.
Keywords: Epistemology of CS, Computational thinking, CS vs. digital literacy

Learning Theories
Constructivism (Piaget, Vygotsky), constructionism (Papert), cognitivism. Applications to computer science: zones of proximal development in coding, scaffolding.
Keywords: Constructivism, Constructionism, Papert & Logo, Scaffolding, ZPD

Active Teaching Methodologies
Peer instruction, flipped classroom, pair programming, PRIMM, problem-based learning. Typical misconceptions in programming learning and how to address them.
Keywords: Peer instruction, Flipped classroom, Pair programming, PRIMM, Misconceptions

Assessment of Learning
Formative vs. summative assessment, rubrics, digital portfolios. Self-assessment and peer assessment. Introduction to CS Education as a research field.
Keywords: Formative assessment, Rubrics, Portfolio, CS Education research

Curriculum Design and Teaching Units
Backward design (Wiggins & McTighe), Bloom's taxonomy revised for computer science, writing operational objectives, choosing content by school cycle.
Keywords: Backward design, Bloom's taxonomy, National Guidelines, Teaching unit design, Differentiation

Tools and Learning Environments
Unplugged activities (CS Unplugged), block-based environments (Scratch, App Inventor), Python for education, educational robotics, artificial intelligence in the classroom.
Keywords: CS Unplugged, Scratch, Python for education, Robotics, AI in schools, Digital accessibility

Micro-teaching and Feedback
Each student teaches a mini-lesson (15–20 min) on a topic of their choice. Structured peer observation, feedback using an agreed grid, post-lesson reflection.
Keywords: Micro-teaching, Structured feedback, Peer observation, Reflection

Prerequisites

Knowledge of Python is welcome but not required.

Teaching form

Blended learning. 20 hours of the 52 total amount are delivered online. Exchange activities with secondary school teachers are planned, possibly including field observation visits.

Textbook and teaching resource

Orit Hazzan, Noa Ragonis, Tami Lapidot. Guide to Teaching Computer Science - An activity based approach. Third edition, Springer Nature Switzerland, 2020, ISBN 978-3-030-39359-5.

Course slides.

Papers suggested by the teacher(s).

Other resources:

Guzdial, M. (2015). Learner-Centered Design of Computing Education. Morgan & Claypool.
Wing, J. M. (2006). Computational thinking. CACM, 49(3), 33–35.
Wiggins, G. & McTighe, J. (2005). Understanding by Design. ASCD.
CSTA K–12 CS Standards (2017, aggiornamento 2022). csteachers.org
CS Unplugged. csunplugged.org — attività unplugged pronte all'uso.
MIUR Indicazioni Nazionali per il Liceo Scientifico e gli Istituti Tecnici (Informatica).

Semester

Second semester - A.Y. 2026/2027

Assessment method

Oral examination based on a complete instructional design exercise. Assessment breakdown:

- Teaching Unit Project (50%) — The student designs a complete teaching unit on a computer science topic for secondary school, including operational objectives, activity sequence, chosen tools, and student assessment criteria. The project is discussed orally during the exam, where the candidate justifies their didactic choices and responds to critical questions from the instructor.
- Micro-teaching (30%): The student delivers a 15–20 minute mini-lesson in class on a topic of their choice, addressed to the instructor who simulates the role of a secondary school class. The lesson is followed by a reflection in which the candidate analyses the strengths and weaknesses of their own performance, taking into account the feedback received.
- Participation and reflective journals (20%): Throughout the weeks of the course, the student writes short reflective journals (approximately half a page) after each laboratory session, documenting what they have learned, what surprised them, and how they would connect the content to their future teaching practice. The assessment takes into account both the regularity of submission and the depth of reflection.

Office hours

By appointment.

Sustainable Development Goals

QUALITY EDUCATION