

Methods for teaching computer science

Course Description

Familiarizes teachers, grades with pedagogical approaches and innovative teaching techniques needed to convey to a diverse population emerging state and professional standards-based curriculum in computer science. Innovative uses of technology, development of instructional units, individualizing for a diverse population of students, and strategies for managing problem behavior will be emphasized throughout the course.

Schedule

- July 9 - July 30, Monday - Friday 12:30 - 3:00

Grading

- 30% : Demonstration of an understanding of CS teaching methods through class discussion, breakout rooms, small presentations.
- 30% : Use of effective methods in delivery of Topics project
- 20% : Final portfolio
- 20% : participation

By the end of this course, students will be able to:

- develop coherent, meaningful goals, plans and materials.
- Plan and deliver effective computer science instruction.
- assess learning and differentiate instruction for diverse populations.
- Employ a variety of effective pedagogical standards in order to deliver instruction aligned with the proposed CSTA standards.
- Use current technology and CS content and will be able to learn new technologies and content. Teacher-candidates will be able to develop coherent, meaningful goals, plans, and materials.

Texts

N/A

Standards note

Area	Standards Covered
IC	1,2,3,4
CT	4,9,10
DL	1,2,3,4,5,6,7

Topics

1. July 9
 - Why should students study computer science?
 - What are the difficulties in teaching and learning computer science?
 - What tools can we use to explore CS?
 - Describe portfolio requirement (see last topic)
2. July 12
 - The anatomy of a CS lesson
 - content
 - Styles
 - Pacing
 - Assignment: design a sample computer science lesson
1. July 13
 - Existing courses and frameworks
 - overview of existing standards (CSTA, NY)
 - overview of popular courses and materials
 - Assignment: bring in example of a lesson from a popular resources
1. July 14
 - Live coding
 - what is it and why is it effective
 - live coding demo
 - lab: design and demo a live coding experience from a CS1 program
1. July 15
 - worked examples and code tracing
1. July 15
 - Unplugged assignments
 - what and why
 - lab
 - * paper plane lab
 - * graph paper programming
 - Assignment: design an unplugged experience.
 - Standards
 - * IC 1
1. July 16 Active learning
 - worked examples (code reading)
 - making predictions
 - Assignment: design a worked example experience.
 - Standards
 - CT 9

1. July 19 Pair programming lab
 - Standards
 - CT 10
1. July 20
 - Scaffolding assignments
 - Code templates
 - starter code
 - code idioms
 - Assignment: Analyze an existing program. What would you pull out as starter code or code templates for a student assignment.
 - Standards
 - CT 4
1. July 21
 - SE techniques that can be used in the classroom
 - coding standards
 - Subgoal labelling
 - lab/assignment: Break down a CS program with subgoal labels.
 - Standards
 - CT 9
2. July 22
 - Differentiated (layered) assignments
 - Class exercise: Designing a differentiated experience around Conway's game of life
3. July 23
 - Debugging as a class technique part 1
 - Code tracing review
 - Print statement debugging
 - Using a debugger
4. July 23
 - SE techniques that can be used in the classroom part 2
 - Rubber Duck Debugging - code explaining
 - Standards
 - CT 9
5. July 26 Using other people's code.
 - Using library code
 - "thwarting" assignments
 - Ethical issues (copying vs collaborating)
 - Standards
 - IC 1,2

- CT 4
 - Assignment: Design an experience that makes use of an existing library.
- 6. July 27 Choosing the correct tools part 1
 - comparison of NetLogo, Scratch, Scheme, Python and Java
 - Class activity - explorations
- 7. July 28 Choosing the correct tools part 2
 - Debrief and analysis from part 1
- 8. July 29 Setting up student workflows
 - Tools
 - language sets
 - resoruces
 - Standards
 - DL 1-7
- 9. July 30 Using remote tools to support learning
 - Chat and communication tools (Slack, Piazza)
 - Code repositories (GitHub, GitLab)
 - Standards
 - IC 3, 4
- 10. Portfolio assignment due 1 week after last class session.

Students are to create a detailed lesson plan, outline and supporting materials for a computer science lesson. The lesson plan and outline should include a description of the intended student audience and a detailed explanation of how multiple pedagogical techniques can be used in delevering the lesson.

For example, if the lesson includes subgoal labelling, the project should include the finished desired code along with the subgoal labels included within.

The plan must also be aligned to the New York State Standerds for Digital Fluency and Computer Science.

The completed assignment should be such that the portfolio can be given to any computer science teacher and they could deliver the lesson as intendded.

Hunter College Policy on Academic Integrity

Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty.

The College is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures.

ADA Policy

In compliance with the American Disability Act of 1990 (ADA) and with Section 504 of the Rehabilitation Act of 1973, Hunter College is committed to ensuring educational parity and accommodations for all students with documented disabilities and/or medical conditions. It is recommended that all students with documented disabilities (Emotional, Medical, Physical, and/or Learning) consult the Office of AccessABILITY, located in Room E1214B, to secure necessary academic accommodations. For further information and assistance, please call: (212) 772- 4857 or (212) 650-3230.