# Computer Science Supplementary Authorization - Summer 2021



## Syllabus for SCI 5200

### Instructor and Contact Information

Instructor: Melissa Danforth (pronouns: she/her/hers)
Email: mdanforth@csub.edu
Discord: https://discord.gg/h79N87P (Note: When you first connect, let me know you're a CSSA student)
Slack: TBA

### Catalog Description

**SCI 5200 Computational Thinking (3)**
A graduate level course offered in an online format that satisfies the Computational Thinking requirement for an Introductory Supplementary Authorization in Computer Science. Topics include solving problems and designing systems using fundamental computing concepts such as decomposition, data representation, generalization/abstraction and algorithms. Students will learn effective computational thinking and programming techniques for the classroom through study of historically significant works, conducting literature search in methods/case studies, and critical discussion in seminar setting.

### Prerequisites

Admission to the M.A. in Education - Curriculum and Instruction program or to the Introductory Computer Science Supplementary Authorization (CSSA) program.

### Units and Contact Time

3 units lecture. 10 week virtual asynchronous format. Self-paced. Plan to spend about 5-6 hours per week on videos, reading assignments, quizzes, and course assignments to complete the course in 10 weeks.

### Type

Satisfies the Computational Thinking requirement of the Introductory CSSA.

### General Course Notes

This course is organized into an introductory area, then three main sections (also called "strides", a software development term): (1) Foundational Knowledge in Computational Thinking, (2) Sample Lesson Plans for Computational Thinking, and (3) Block Programming with Scratch. Each section has a capstone assignment that will be used to assess your learning in that section.

Individual modules within each section generally contain four parts: (1) the learning objectives for that module, (2) required reading assignments, (3) optional resources that will go further in-depth on the topic, and (4) my slides and video about the module. Some modules are on the honor system, where you check-off Moodle's completion box to indicate that you have completed the module. Some modules have an assessment quiz to gauge your learning. Assessment quizzes allow unlimited attempts, and the highest score is retained in the gradebook.

You will begin the course with modules in the introductory area. You will also need to complete the Ubiquity of Computing module and quiz from SCI 5230 Impacts of Computing before you can start the final introductory module. As you mark modules as complete and get at least a 60% grade on the assessment quizzes, additional modules will be unlocked.

Collaborative and exploratory learning is encouraged in this course, with the below Academic Integrity note about turning in one's own work in one's own words.

### Required Textbook(s)

- Foundational Knowledge section: "Computational Thinking: A Beginner's Guide to Problem Solving and Programming" by Karl Beecher. BCS Learning & Development, 2017. ISBN 978-1-78017-36-41.

- Sample Lesson Plan section: "Computational Thinking {and Coding} for Every Student: The Teacher's Getting Started Guide" by Jane Krauss and Kiki Prottsman. Corwin, 2017. ISBN 978-1-506341286.
- Block Programming section: "Computer Science in K-12: An A to Z Handbook on Teaching Programming" edited by Shuchi Grover. Edfinity, 2020. ISBN 978-1-7346627-0-2.

## Recommended Textbook(s) and Other Supplementary Materials

Recommended textbook for additional applications of computational thinking:

- "The Power of Computational Thinking: Games, Magic and Puzzles to Help You Become a Computational Thinker" by Paul Curzon and Peter W. McOwan. World Scientific, 2017. ISBN 978-1-786341846.

Recommended academic papers on computational thinking:

- Wing, Jeannette M. "Computational thinking." Communications of the ACM 49.3 (2006): 33-35. ACM: https://dl.acm.org/doi/10.1145/1118178.1118215
- Lee, Irene, et al. "Computational thinking for youth in practice." ACM Inroads 2.1 (2011): 32-37. ACM: https://dl.acm.org/doi/10.1145/1929887.1929902
- Brennan, Karen, and Mitchel Resnick. "New frameworks for studying and assessing the development of computational thinking." Proceedings of the 2012 annual meeting of the American Educational Research Association (AERA), Vancouver, Canada. Vol. 1. 2012. MIT: https://www.media.mit.edu/publications/new-frameworks-for-studying-and-assessing-the-development-of-computational-thinking/
- Weintrop, David, et al. "Defining computational thinking for mathematics and science classrooms." Journal of Science Education and Technology 25.1 (2016): 127-147. SpringerLink: https://link.springer.com/article/10.1007/s10956-015-9581-5
- Lye, Sze Yee, and Joyce Hwee Ling Koh. "Review on teaching and learning of computational thinking through programming: What is next for K-12?." Computers in Human Behavior 41 (2014): 51-61. Science Direct: https://www.sciencedirect.com/science/article/abs/pii/S0747563214004634

## Course Topics

| Section | Modules |
|---|---|
| Introduction | Module 0 - Demystifying Computer Science<br>Module 1 - History of Computer Science<br>Module 2 - Overview of Computational Thinking |
| Foundational Knowledge | Module 3 - Algorithmic Thinking<br>Module 4 - Decomposition<br>Module 5 - Pattern Recognition and Generalization<br>Module 6 - Abstraction and Modeling<br>Module 7 - Evaluation<br>Module 8 - Cybersecurity Concepts for Algorithms and Program Design (optional)<br>Module 9 - Foundational Knowledge Capstone |
| Sample Lesson Plans | Module 10 - "Do's" and "Don'ts" of Teaching CS and Computational Thinking<br>Module 11 - Plans for Algorithmic Thinking<br>Module 12 - Plans for Decomposition<br>Module 13 - Plans for Pattern Recognition and Generalization<br>Module 14 - Plans for Abstraction and Modeling<br>Module 15 - Plans for Evaluation<br>Module 16 - Lesson Plan Capstone |
| Block Programming | Module 17 - Block Programming Overview<br>Module 18 - Basic Concepts for Scratch Programming<br>Module 19 - Flow Control in Scratch<br>Module 20 - Graphical and Audio Elements in Scratch<br>Module 21 - Scripts and Subroutines in Scratch<br>Module 22 - Scratch Programming Project<br>Module 23 - Tips for Teaching Scratch<br>Module 24 - Block Programming Capstone |

## Mapping to California K-12 Computer Science Standards

| Grade Level | Foundational Knowledge & Sample Lesson Plans sections | Block Programming section |
|---|---|---|
| K-2 | K-2.AP.10 – Algorithms for tasks<br>K-2.AP.11 – Model how data is stored (variables)<br>K-2.AP.13 – Decomposition<br>K-2.AP.14 – Develop plan<br>K-2.AP.16 – Debug algorithm / program | K-2.AP.12 – Create program w/ loops<br>K-2.AP.14 – Develop plan for program<br>K-2.AP.16 – Debug algorithm / program<br>K-2.AP.17 – Describe programming steps / choices |
| 3-5 | 3-5.AP.10 – Compare / refine algorithms<br>3-5.AP.13 – Decomposition<br>3-5.AP.17 – Test / debug algorithm / program<br>3-5.AP.19 – Describe choices made | 3-5.AP.11 – Create program w/ data (variables)<br>3-5.AP.12 – Create program w/ loops, events, conditions<br>3-5.AP.15 – Plan program w/ input from others<br>3-5.AP.17 – Test / debug algorithm / program<br>3-5.AP.18 – Peer programming / collaboration<br>3-5.AP.19 – Describe choices made<br>3-5.IC.21 – Accessibility concerns for technology |
| 6-8 | 6-8.AP.10 – Flowcharts and pseudocode<br>6-8.AP.13 – Decomposition<br>6-8.AP.15 – Incorporate diverse perspectives / societal need<br>6-8.AP.17 – Systematically test w/ test cases<br>6-8.AP.18 – Work collaboratively<br>6-8.NI.5 – Explain potential security threats (optional module on cybersecurity) | 6-8.DA.9 – Test / analyze effects of changing variables<br>6-8.AP.11 – Create clearly named variables w/ operators<br>6-8.AP.12 – Create program w/ compound flow control<br>6-8.AP.14 – Create procedures w/ parameters<br>6-8.AP.15 – Incorporate diverse perspectives / societal need<br>6-8.AP.17 – Systematically test w/ test cases<br>6-8.AP.18 – Work collaboratively<br>6-8.IC.21 – Issues of bias and accessibility in design<br>6-8.IC.22 – Collaborate w/ many contributors |
| 9-12 | 9-12.CS.1 – Abstraction<br>9-12.DA.11 – Refine computational models<br>9-12.AP.12 – Design algorithms<br>9-12.AP.13 – Create generalized solutions<br>9-12.AP.16 – Decomposition<br>9-12.AP.20 – Evaluate artifact & refine<br>9-12.AP.22 – Document and justify decisions<br>9-12.IC.25 – Application of algorithm to different disciplines | 9-12.AP.14 – Justify using specific control structures<br>9-12.AP.15 – Design project for practical use, personal use, or to address societal issue<br>9-12.AP.17 – Use modular design<br>9-12.AP.18 – Design for broad audience w/ user feedback<br>9-12.AP.21 – Team programming and collaboration<br>9-12.AP.22 – Document and justify decisions<br>9-12.IC.24 – Impacts of bias / equity deficits on design |

## Academic Integrity Policy

You may discuss the assignments with others in the class. A message board is also available on Moodle for discussions. However, all assignments in this course are individual assignments. For individual assignments, each student must turn in their own work in their own words; no direct copying from any source or each other is allowed.

Refer to the Academic Integrity policy in the campus catalog and class schedule for more details. You can also refer to the Academic Integrity policy at the Office of Student Rights and Responsibilities at https://www.csub.edu/osrr/

## Academic Accommodations

To request academic accommodations, please contact the Office of Services for Students with Disabilities (SSD) and email your instructor an accommodations letter from the SSD Office. Policies from the SSD Office relating to accommodations, such as scheduling policies for using their testing center, must also be followed. For more information about the services and policies of the SSD Office, contact their staff by email and/or visit their website at https://www.csub.edu/ssd/

## Grading Categories

| Category | Weight |
|---|---|
| Assessment quizzes | 5% |
| Scratch programming project | 5% |
| Foundational knowledge capstone | 30% |
| Lesson plan capstone | 30% |
| Block programming capstone | 30% |

Changelog:
25 July 2021: Fixed the link errors in the module list, so now all modules here in the syllabus link to the proper Moodle pages

Last modified: Sunday, July 25, 2021, 12:03 PM

◄ CSSA Lesson Plan Rubric

Jump to...

Overview of SCI 5200 ►