EDG 4430: Computational Thinking and Problem Solving for Teachers.

3 Credits

COURSE SYLLABUS

Semester: Fall 2026
Course Type: 100% Full Distance Learning
Instructor: Dr. Zafer Unal
Office Hours: Online with Appointment
Phone: 727-873-4803
Email: unal@usf.edu

I. Welcome!

This course introduces the foundational principles of computational thinking and algorithmic problem-solving, specifically designed for K-12 educational settings. We will explore how to break down complex problems, design step-by-step solutions, and apply a structured development process that you can use in your own classroom. Along the way, we'll master strategies for designing algorithms, predicting their behavior, and thoroughly testing solutions. We'll also cover the complete Software Development Life Cycle (SDLC), providing you with a robust framework for tackling challenges systematically. By the end of this course, you'll have a powerful toolkit of strategies to enhance your students' critical thinking and problem-solving skills. Let's embark on this journey to unlock the potential of computational thinking in education!

II. University Course Description

This course introduces foundational principles of computational thinking and algorithmic problem-solving for K-12 educators. Students learn to decompose complex problems, design algorithms using flowcharts and pseudocode, and apply the Software Development Life Cycle. Topics include algorithm selection, execution prediction, and test data design for validating solutions.

III. Course Prerequisites

None

IV. Course Purpose

This course serves as a foundational introduction to computational thinking and problem-solving. As these skills become increasingly vital in educational contexts, this course equips K-12 teachers with the theoretical knowledge and practical skills needed to decompose problems, design and analyze algorithms, and apply a structured development process. The course focuses on exploring and practicing these strategies, making it essential for educators seeking to foster critical thinking and systematic reasoning in their students.

V. Course Format

This fully online course is structured through weekly modules that combine asynchronous recorded lectures with hands-on technical demonstrations and projects. Students engage through discussion boards for peer interaction and complete practical assignments applying AI tools to educational scenarios. The course offers flexibility through asynchronous learning while maintaining support through individual consultation sessions with the instructor as needed.

VI. Student Learning Outcomes

Upon completion of this course, students will be able to:

- Decompose complex problems into manageable components and apply appropriate solution strategies, as measured by graded problem analysis assignments and exams using a scoring rubric (minimum 70% proficiency).
- Design and represent algorithms using flowcharts and pseudocode, as measured by algorithm design projects evaluated with a standardized rubric (minimum 70% proficiency).
- Apply the stages of the Software Development Life Cycle (SDLC) to an educational problem, as measured by a course design project and written documentation evaluated with a project rubric (minimum 70% proficiency).
- Determine and select an appropriate algorithm for a given problem, as measured by quizzes and written algorithm selection reports (minimum 70% proficiency).
- Predict the outputs of algorithms for a given input, as measured by quizzes and algorithm tracing assignments (minimum 70% proficiency).
- Identify an appropriate set of data necessary for testing a computer solution, as measured by test-case design assignments and exams using a scoring rubric (minimum 70% proficiency).

VII. Course Objectives

Students will:

- Practice problem decomposition through hands-on exercises and case studies.
- Create and test algorithms for a variety of problems using flowcharts and pseudocode.
- Document each phase of a small educational project using the SDLC framework.
- Analyze and compare different algorithmic approaches (e.g., sorting) for efficiency and appropriateness.
- Trace algorithm execution to predict outputs and identify logic errors.
- Develop comprehensive test data sets for positive and negative testing scenarios.

VIII. Required Texts and/or Readings and Course Materials

- There are no required text for this course. All of the course materials (videos, readings etc. will be available on the course website with free access. A computer with internet access and capable of running AI applications is required for this course.

IX. Supplementary (Optional) Texts and Materials

NA

X. How to Succeed in this Course

To succeed in this course, students should establish strong study habits from the beginning. This includes completing all weekly readings prior to attempting practical exercises and actively engaging with technical demonstrations. Time management is crucial - start assignments well before deadlines to allow for troubleshooting and maintain detailed documentation of your technical implementation process. Regular engagement with course materials, peers, and AI tools is essential for building practical skills. Finally, always maintain backup copies of all project work to prevent any potential data loss during technical exercises.

XI. Academic Continuity

As this course is already fully online, any disruptions due to emergencies or severe weather will have minimal impact on course delivery. All course materials, including recorded lectures, assignments, and resources, will remain accessible through Canvas. If USF systems are impacted, backup copies of essential materials will be made available through Microsoft Teams. The instructor will communicate any changes or contingency plans through Canvas announcements and email. While most activities are asynchronous, any scheduled synchronous sessions (like individual consultations) will be rescheduled if disrupted, with alternative times communicated through Canvas.

XII. Communication

Primary communication for this course will be through Canvas messaging and announcements. For urgent matters, you may contact your instructor with the email provided on top. Your instructor typically responds to messages within 24 hours during weekdays and 48 hours on weekends. Virtual office hours are available by appointment through Microsoft Teams. Professional communication etiquette is expected in all interactions - please include your name and course number in email subjects.

XIII. Grading Scale

Grading Scale (%)	
90-100	A
80 - 89	B
70 - 79	C
60 - 69	D
0 - 59	F

XIV. Grade Categories and Weights

The final grade is directly tied to the mastery of each sub-competency. Each assignment will be graded using a specialized rubric designed to assess mastery of its corresponding sub-competency.

Sub-Competency	Assessments	Weight of final Grade
1.1: Analyze a problem and apply appropriate solution strategies	Discussion 1.1: Problem Decomposition in Daily Life, Problem Analysis Assignment 1.1: Classroom Challenge Solution Design	15%
1.2: Apply the steps of algorithmic problem solving	Discussion 1.2: Stepwise Refinement in Practice,	20%

when designing solutions to problems	Algorithmic Problem-Solving Project 1.2: From High-Level to Detailed Algorithm	
1.3: Apply the stages of the software development life cycle	Discussion 1.3: SDLC Phase Interdependence, SDLC Design Project 1.3: Educational Tool Design Document	20%
1.4: Determine and select an appropriate algorithm for a given problem	Quiz 1.4, Algorithm Selection Report 1.4: Sorting Algorithm Comparison	15%
1.5: Predict outputs of algorithms for a given input	Quiz 1.5, Algorithm Tracing Assignment 1.5: Pseudocode Output Prediction	15%
1.6: Identify an appropriate set of data necessary for testing a computer solution	Quiz 1.6, Test Data Creation Assignment 1.6: Comprehensive Test Suite Design 6	10%
Course Participation	Consistent engagement in weekly modules and discussions	5%
TOTAL		100%

XV. Instructor Feedback Policy & Grade Dissemination

Course modules open every Monday at 12:00 AM and close Sunday at 11:59 PM EST. All assignments must be submitted within this one-week window; late submissions are not accepted given the full week provided for completion. Individual feedback and grades will be posted in Canvas by Wednesday 5:00 PM of the following week. Major assignments like projects and the final portfolio may require additional grading time, which will be communicated in advance. All grades and feedback can be accessed through the Canvas gradebook.

XVI. Course Schedule.

Week	Topics & Subitems	Assessment/Assignment
1	**The Art of Problem Solving (Sub-competency 1.1):** (1) Phases of problem-solving (2) Understanding the problem (3) Problem analysis (4) Alternative consideration and approach selection	**Discussion 1.1: Problem Decomposition in Daily Life** Decompose a routine daily task (like making breakfast or commuting to work) into its core components and propose an alternative, more efficient strategy.
2	**Mastering Problem Analysis (Sub-competency 1.1):** (1) Creative problem solving (2) Problem decomposition (3) Solution strategy application (4) The String Method example	**Problem Analysis Assignment 1.1: Classroom Challenge Solution Design** Given a scenario of students struggling with a multi-step math problem, analyze the root causes and design a step-by-step solution strategy a teacher could implement.

3	**Algorithmic Problem Solving (Sub-competency 1.2):** (1) The algorithm development process (2) Obtaining a problem description (3) Problem analysis (4) Developing a high-level algorithm	**Discussion 1.2: Stepwise Refinement in Practice** Post a high-level algorithm for "preparing a lesson plan" and provide a first refinement for one of the steps. Critique a peer's refinement for clarity and unambiguity.
4	**Representing Algorithms (Sub-competency 1.2):** (1) Refining algorithms with detail (2) Stepwise refinement (3) Algorithm review (4) Introduction to Flowcharts and Pseudocode	**Algorithmic Problem-Solving Project 1.2: From High-Level to Detailed Algorithm** Choose a process (e.g., a student login system, checking out a library book). Develop a high-level algorithm, refine it twice using stepwise refinement, and represent the final version as both a flowchart and pseudocode.
5	**Software Development Life Cycle - Part 1 (Sub-competency 1.3):** (1) SDLC phases (2) Requirement Management & Analysis (3) The importance of baselining (4) Project Planning	**Discussion 1.3: SDLC Phase Interdependence** Using the example of developing a gradebook app, explain how a mistake in the Requirements phase could negatively impact the Design and Testing phases.
6	**Software Development Life Cycle - Part 2 (Sub-competency 1.3):** (1) Architectural Design (2) Development & Quality Assurance (3) User Acceptance Testing & Deployment (4) Maintenance	**SDLC Design Project 1.3: Educational Tool Design Document** Write a simple design document for a new educational tool (e.g., a quiz app, a classroom behavior tracker). The document must outline its purpose, core features, a simple test plan, and a maintenance consideration.
7	**Determining and Selecting Algorithms (Sub-competency 1.4):** (1) Algorithm characteristics (ordered, unambiguous, executable, terminating) (2) Comparing algorithmic approaches	**Quiz 1.4:** Covers algorithm characteristics and the process of selecting an appropriate algorithm.
8	**Sorting Algorithms (Sub-competency 1.4):** (1) Introduction to sorting (2) Merge Sort (3) Selection Sort (4) Algorithm efficiency considerations	**Algorithm Selection Report 1.4: Sorting Algorithm Comparison** You need to sort a list of student names and a list of exam scores. Compare Merge Sort and Selection Sort. Justify which algorithm you would use for each task and explain your reasoning based on their characteristics.
9	**Predicting Algorithm Outputs (Sub-competency 1.5):** (1) Algorithm tracing (2) Dry-running pseudocode (3) Identifying logic errors (4) The role of prediction in debugging	**Quiz 1.5:** Tests understanding of how to trace algorithms and predict outcomes based on given inputs.
10	**Advanced Algorithm Prediction (Sub-competency 1.5):** (1) Regression vs. Decision Trees for prediction (2)	**Algorithm Tracing Assignment 1.5: Pseudocode Output Prediction** Given several short algorithms in pseudocode (e.g., a loop that calculates a sum, a conditional that

	Flowcharts for workflow (3) Practical prediction exercises	modifies a string), trace execution with sample inputs to predict the final outputs and identify any logic errors.
11	**Data for Testing Solutions (Sub-competency 1.6):** (1) What is test data? (2) Positive and negative testing (3) Creating effective test data sets (4) Boundary value analysis	**Quiz 1.6:** Covers the purpose and principles of creating test data.
12	**Designing Test Data (Sub-competency 1.6):** (1) Test data generation methods (2) Boolean operators in conditions (3) Creating data for all test scenarios	**Test Data Creation Assignment 1.6: Comprehensive Test Suite Design** For a program that calculates a student's final grade based on inputs (e.g., homework avg, exam score, participation), create a comprehensive set of test data that covers normal cases, edge cases (like 0%, 100%), and invalid inputs (like negative scores).
13	**Integration & Application:** (1) Case study analysis (2) Applying all competencies to a complex problem (3) Peer review	**Final Review Discussion:** Analyze a case study of a failed educational software rollout. As a group, identify which sub-competencies were likely neglected and propose how applying them could have led to a better outcome.
14	**Synthesis & Portfolio Finalization:** (1) Course synthesis (2) Portfolio assembly and review (3) Future application in K-12 settings	**All Final Assignments Due:** Final, polished versions of all major projects (1.1, 1.2, 1.3, 1.4, 1.5, 1.6) must be submitted as a cumulative portfolio by the end of Week 14.

* Note: The Schedule is subject to revision

XVII. General Education Statement (undergraduate only – Required if a Gen Ed course)

NA

XVIII. Integration of This Course into Your Academic Experience (Gen Ed courses only)

NA

XIX. Global Citizens Project (only required if a GCP course; must be verbatim)

NA

XX. USF Core Syllabus Policies

USF has a set of central policies related to student recording class sessions, academic integrity and grievances, student accessibility services, academic disruption, religious observances, academic continuity, food insecurity, pregnancy and related conditions, and sexual harassment that **apply to all courses at USF**. Be sure to review these online: usf.edu/provost/faculty-success/resources-policies-forms/core-syllabus-policy-statements.aspx

XXI. Course Policies: Grades

Late Work Policy: Offer specifics about your policy on late work.

Each module provides a full 7-day window for completion, and no late submissions will be accepted after the weekly deadline. To ensure success in this course, it is strongly recommended to begin each module on Monday rather than waiting until the weekend. Starting early allows time to troubleshoot technical issues, engage meaningfully in discussions, and seek clarification if

needed. Students who consistently complete work early in the module week typically perform better and experience less stress than those who leave work until the last minute.

Medical Excuses:
If illness prevents coursework completion, students must immediately notify the instructor and submit medical documentation. While this is an online course with no physical attendance requirements, prompt communication regarding any medical issues impacting your ability to complete coursework is essential for arranging accommodations.

Grades of "Incomplete":
For graduate courses: An Incomplete grade ("I") is exceptional and granted at the instructor's discretion only when students are unable to complete course requirements due to illness or other circumstances beyond their control. The course instructor and student must complete and sign the "I" Grade Contract Form that describes the work to be completed, the date it is due, and the grade the student would earn factoring in a zero for all incomplete assignments. The due date can be negotiated and extended by student/instructor as long as it does not exceed two semesters for undergraduate courses and one semester for graduate courses from the original date grades were due for that course. An "I" grade not cleared within the two semesters for undergraduate courses and one semester for graduate courses (including summer semester) will revert to the grade noted on the contract.

Attendance Policy:
For this online course, regular participation is measured through timely completion of weekly modules and engagement with course materials. Students are expected to log in to the course per week to review content, participate in discussions, and complete assignments.

Campus Free Expression:
It is fundamental to the University of South Florida's mission to support an environment where divergent ideas, theories, and philosophies can be openly exchanged and critically evaluated. Consistent with these principles, this course may involve discussion of ideas that you find uncomfortable, disagreeable, or even offensive. In the instructional setting, ideas are intended to be presented in an objective manner and not as an endorsement of what you should personally believe. "Objective" means that the idea(s) presented can be tested by critical peer review and rigorous debate, and that the idea(s) is supported by credible research. In this course you may be asked to engage with complex ideas and to demonstrate an understanding of the ideas. Understanding and engaging with an idea does not require you to believe it or to agree with it.

Final Examinations Policy:
No exam, non-applicable

XXII. Course Policies: Technology and Media (include sections as applicable to your course)

Canvas:
This course is fully delivered through Canvas Learning Management System. Students must log in regularly to access course materials, assignments, and announcements. Weekly modules open

Monday 12:00 AM and close Sunday 11:59 PM EST. All course communication, submission of assignments, and grade distribution will be conducted through Canvas. For technical support, contact USF IT at (813) 974-1222 or help@usf.edu.

XXIII. Course Policies: Student Expectations

Health and Wellness: Example statement supporting student health and wellbeing.

Your health is a priority at the University of South Florida. We encourage members of our community to look out for each other and to reach out for help if someone is in need. If you or someone you know is in distress, please make a referral at www.usf.edu/sos so that the Student Outreach & Support can contact and provide helpful resources to the student in distress. A 24-hour licensed mental healthcare professional, offered through the counseling center, is available by phone at 813-974-2831, option 3. Please remember that asking for help is a sign of strength. In case of emergency, please dial 9-1-1.

Title IX Policy: Title IX provides federal protections for discrimination based on sex, which includes discrimination based on pregnancy, sexual harassment, and interpersonal violence. In an effort to provide support and equal access, **USF has designated all faculty (TA, Adjunct, etc.) as Responsible Employees, who are required to report any disclosures of sexual harassment, sexual violence, relationship violence or stalking.** The Title IX Office makes every effort, when safe to do so, to reach out and provide resources and accommodations, and to discuss possible options for resolution. Anyone wishing to make a Title IX report or seeking accommodations may do so online, in person, via phone, or email to the Title IX Office. For information about Title IX or for a full list of resources please visit: https://www.usf.edu/title-ix/gethelp/resources.aspx. *If you are unsure what to do, please contact Victim Advocacy – a confidential resource that can review all your options – at 813-974-5756 or va@admin.usf.edu.*

Generative Artificial Intelligence (AI) Policy:

Students may use AI tools (e.g., ChatGPT) for brainstorming and learning support, but all submitted work must be the student's own original work. AI-generated content may not be submitted as final coursework unless explicitly authorized by the instructor. Students must acknowledge AI assistance when used. Unauthorized use of AI to complete assignments will be considered a violation of the USF Academic Integrity Policy.

Course Hero / Chegg Policy: The USF Academic Integrity Policy prohibits the use of contract cheating services and online platforms that facilitate academic dishonesty. Students may not upload, download, or use course materials or solutions from websites such as CourseHero, Chegg, or similar platforms for the purpose of completing assignments or assessments. Violations of this policy will be reported and may result in disciplinary action in accordance with university procedures.

Professionalism Policy: All course interactions must maintain professional etiquette. Students will communicate respectfully in discussion boards, emails, and group work. Written communications must be clear, appropriate, and maintain a professional tone. Disruptive behavior in online interactions may affect your final grade. For group activities, timely responses and meaningful contributions are required.

Netiquette Guidelines

Professional communication is essential in our online learning environment. When participating in course activities, maintain the same respect and courtesy as in face-to-face interactions. Consider diverse perspectives and remember that written communication lacks verbal cues. Keep messages clear, focused, and constructive. Always proofread for clarity, avoiding all caps and informal language. Complete thoughts on one topic before introducing new ones. Avoid sarcasm and humor that could be misinterpreted. Following these guidelines in Canvas will create a positive learning environment.

End of Semester Student Evaluations:
All classes at USF make use of an online system for students to provide feedback to the University regarding the course. These surveys will be made available at the end of the semester, and the University will notify you by email when the response window opens. Your participation is highly encouraged and valued.

XXIV. Learning Support and Campus Offices

Academic Accommodations
Students with disabilities are responsible for registering with Student Accessibility Services (SAS) to receive academic accommodations. Information about accommodations and resources is available through SAS for all USF campuses.

Canvas Technical Support
This course is delivered through Canvas. Students experiencing technical difficulties should consult the Canvas Help resources within the course or contact USF IT Support at **813-974-1222** or **help@usf.edu**.

Center for Victim Advocacy
The Center for Victim Advocacy provides confidential support and resources for students affected by violence, abuse, or harassment. Additional information and contact details are available on the USF Center for Victim Advocacy website.

Counseling Center
The USF Counseling Center offers confidential counseling services, consultations, and mental health resources for students. Students are encouraged to seek support as needed. Contact information is available on the USF Counseling Center website.

Tutoring Services
The Tutoring Hub offers free academic tutoring for USF students. Appointments are recommended but not required. Additional information is available through the USF Tutoring Hub website.

Writing Studio
The USF Writing Studio provides free writing support for undergraduate and graduate students at any stage of the writing process. Appointments are recommended but not required. Information is available on the USF Writing Studio website.

XXV.Important Dates to Remember

All dates, assignments, and course content are tentative and subject to change at the instructor's discretion. Note: While modules follow a weekly schedule, changes will be announced through Canvas at least one week in advance

For official USF academic deadlines and holidays, refer to the Academic Calendar at http://www.usf.edu/registrar/calendars/