

Coding 1 - UAS

Level: 2

Career Cluster: Advanced Manufacturing, Digital Technology

Program: Engineering, Information Technology

Pathway(s): Engineering Applications UAS, Engineering, Coding, Networking Systems and Security

Course Catalog Description

Introduces students to foundational programming concepts through engineering and drone applications. Students develop skills in computational thinking, coding, debugging, data management, and ethical technology use while creating programs that solve real drone-related problems.

Summary

Develops foundational programming knowledge through the context of engineering and drone applications. Students explore career pathways and postsecondary opportunities in engineering, drones, and related technologies while building an understanding of how computing, software, and autonomous systems impact modern society. Throughout the course, students apply ethical and safe computing practices, use computational thinking to plan solutions, and write programs using core concepts such as variables, data types, operators, conditionals, loops, functions, and file handling. Students also learn to manage data, interpret digital information, and understand number systems and computer operations that support coding and drone technologies. Emphasis is placed on testing, debugging, quality assurance, and project development as students design, create, and present beginning-level programs that solve real drone-related problems.

Notes

This revised course keeps those core introductory coding concepts but reframes them within a more clearly defined engineering and unmanned aircraft systems context. The updated version is more pathway-specific and industry connected, emphasizing how programming supports drone systems, autonomous technologies, sensor data, navigation, stabilization, and mission-based problem solving rather than treating coding only as a general computer science skill. New updates include an expanded focus on ethical, safe, and responsible programming; stronger alignment to computational thinking, data and file management, testing and quality assurance, and number systems/computer operations; and the addition of an applied drone programming project that requires students to design, test, document, and present a solution to a drone-related challenge. This revision also adds explicit leadership and work-based learning outcomes, strengthens career exploration and Colorado postsecondary connections, and better integrates employability skills, communication, and authentic workforce expectations, making the course more relevant to the UAS pathway while preserving the foundational programming rigor of the original course.

Student Learning Outcomes

1. Career Exploration and Postsecondary Opportunities

Objective: *Students will analyze career pathways in engineering and drones by exploring industry roles, workplace expectations, required education, certifications, professional organizations, and postsecondary opportunities available in Colorado in order to make informed academic and career decisions.*

- 1) Identify employment opportunities in various fields with a focus in the area of engineering and drones
 - a. Recognize the work typically performed, tools and technology used, and nature of work environments
 - b. Identify potential certifications within the careers
 - c. Find membership organizations associated with the careers
 - d. Understand the necessary education associated within the careers
- 2) Define professionalism within the context of engineering and drones
 - a. Criticism and evaluation
 - b. Presentation
 - c. Assessment
- 3) Recognize laws, regulations, and ethics significant to the fields of engineering and drones
- 4) Identify postsecondary opportunities offered in the state of Colorado
 - a. Technical college, community college, 4-year institutions
 - b. Certificates, associates, bachelor, and advanced degree opportunities

2. Computing, Coding, and the World of Drones

Objective: *Students will explain how computer programming supports drone systems, autonomous technologies, and modern society.*

- 5) Identify major milestones in the development of computers, programming, and autonomous technologies.
- 6) Describe how software is used in drones for navigation, sensing, stabilization, and mission execution.
- 7) Explain the relationship between hardware, software, and user input in a drone system.
- 8) Compare the roles of programming in personal, social, economic, and industrial contexts.
- 9) Analyze how computational innovations have changed engineering, transportation, agriculture, public safety, and media.
- 10) Predict how coding and autonomous systems may continue to transform drone applications in the future.

3. Ethical, Safe, and Responsible Programming

Objective: *Students will evaluate ethical and safety considerations related to programming, data use, and autonomous drone operations.*

- 11) Describe ethical issues related to privacy, security, copyright, piracy, fraud, sabotage, and misuse of software.

- 12) Explain why responsible programming practices are essential in drone and autonomous system design.
- 13) Analyze the risks of unreliable code in systems that collect data, make automated decisions, or control movement.
- 14) Evaluate programming decisions using ethical, legal, and safety-based criteria.
- 15) Defend responsible choices related to open-source software, intellectual property, and data protection.
- 16) Apply safe computing and digital citizenship practices while developing and testing code.

4. Computational Thinking and Program Design

Objective: *Students will design solutions to programming problems by applying computational thinking and pre-coding planning strategies.*

- 17) Define problems by identifying inputs, outputs, constraints, and desired results.
- 18) Break down multistep problems into smaller, manageable components.
- 19) Develop algorithms that solve programming and drone-related challenges.
- 20) Construct pseudocode and flowcharts before writing code.
- 21) Organize program logic using sequence, selection, and iteration.
- 22) Evaluate multiple solution paths and select the most efficient or appropriate strategy for a given task.
- 23) Design logic for a drone mission scenario such as obstacle detection, route decisions, or data collection.

5. Programming Fundamentals and Code Construction

Objective: *Students will write syntactically correct programs that use variables, data types, operators, conditionals, and loops to solve problems.*

- 24) Identify and use appropriate data types such as integers, floats, strings, booleans, and lists/arrays.
- 25) Create meaningful variable names and constants following accepted coding conventions.
- 26) Write programs that gather, store, process, and display information.
- 27) Apply arithmetic, relational, and logical operators to solve computational problems.
- 28) Develop code using conditional statements to support decision-making.
- 29) Develop code using loops to automate repetitive tasks.
- 30) Use functions or modular code to improve readability and reusability.
- 31) Convert planned algorithms into working program code in a teacher-selected programming language.
- 32) Create simple drone-themed applications such as altitude tracking, battery monitoring, waypoint selection, or mission timers.

6. Data, Files, and Information Management

Objective: *Students will manage data effectively within programs and use file structures appropriate for real-world coding tasks.*

- 33) Explain how programs receive, store, manipulate, and output data.
- 34) Organize program data using variables, lists, arrays, and basic file structures.

- 35) Read data from user input and files and write data to files or reports.
- 36) Apply file naming, organization, and management practices across software environments.
- 37) Distinguish between temporary program data and stored data.
- 38) Analyze how drone systems use data from sensors, cameras, GPS, and user commands.
- 39) Develop programs that process drone flight data or mission logs.
- 40) Interpret data outputs to support decisions about drone performance or mission success.

7. Testing, Debugging, and Quality Assurance

Objective: *Students will test, debug, and improve code using structured quality assurance practices.*

- 41) Test programs using valid, invalid, and edge-case inputs.
- 42) Locate syntax, logic, and runtime errors in code.
- 43) Debug code systematically using error messages, tracing, and test cases.
- 44) Evaluate whether a program meets its intended purpose and user requirements.
- 45) Revise code to improve correctness, efficiency, readability, and reliability.
- 46) Document testing procedures, bugs, and revisions.
- 47) Apply quality assurance practices such as version notes, test evidence, and performance checks.
- 48) Assess how faulty code could affect a drone's safety, navigation, or mission outcome.

8. Number Systems, Logic, and Computer Operations

Objective: *Students will explain how computers represent and process information using binary, hexadecimal, and logical structures.*

- 49) Convert numbers between base-10, binary, and hexadecimal.
- 50) Explain why binary representation is essential to digital systems.
- 51) Interpret how data, instructions, and signals are processed by computing devices.
- 52) Relate number systems to storage, memory, and communication in computing systems.
- 53) Analyze how digital logic supports sensor readings, controller outputs, and automated responses in drone systems.
- 54) Apply number system knowledge to practical computing examples.

9. Applied Drone Programming Project

Objective: *Students will create and present a beginning-level programming project that solves a drone-related problem.*

- 55) Plan a project using a defined problem statement, criteria, and constraints.
- 56) Design a program that addresses a drone or autonomous systems challenge.
- 57) Develop code that uses input, processing, output, decision-making, and repetition.
- 58) Test and refine the program based on evidence from trial runs.
- 59) Document the design process, code structure, and testing outcomes.
- 60) Present the final product and justify design decisions using technical vocabulary.
- 61) Evaluate the effectiveness of the project and propose logical next steps for improvement.

10. Leadership

Objective: *Students will demonstrate leadership, teamwork, and professional communication skills while contributing to coding and drone-related projects and classroom activities.*

- 62) Demonstrate leadership skills such as initiative, responsibility, dependability, and accountability in individual and team settings.
- 63) Collaborate respectfully with peers to plan, develop, test, and improve coding solutions.
- 64) Practice effective communication by sharing ideas, giving constructive feedback, and presenting technical information clearly.
- 65) Apply problem-solving and decision-making strategies when resolving team challenges or project obstacles.
- 66) Participate in leadership development opportunities through classroom roles, project management tasks, or CTSO-related activities.

11. Work-Based Learning

Objective: *Students will connect classroom learning to workplace expectations by exploring and engaging in authentic work-based learning experiences related to coding, engineering, and drone technologies.*

- 67) Identify the purpose and value of work-based learning experiences such as guest speakers, industry tours, job shadows, internships, apprenticeships, or school-based enterprises.
- 68) Describe workplace expectations related to attendance, communication, collaboration, safety, confidentiality, and professional conduct.
- 69) Apply technical and employability skills during authentic workplace tasks related to coding and drone applications.
- 70) Reflect on work-based learning experiences to evaluate personal strengths, growth areas, and career interests.
- 71) Document evidence of career readiness through artifacts such as resumes, reflections, portfolios, project summaries, or industry feedback.

12. Career and Technical Student Organizations

Objective: *Using the CTSO Resource Document, chapter activities will provide some of the best opportunities CTSO members will have to learn by doing. A successful program of work creates a positive learning atmosphere in the classroom. CTSO members learn how to accept responsibility, work as a team, manage a budget, and handle success and failure. Per the CTE Admin Handbook, CTSOs are embedded into curriculum.*

Professional Development

Prepare each member for entry into the workforce and provide a foundation for success in a career. Becoming a professional does not stop with acquiring a skill, but involves an increased awareness of the meaning of good citizenship and the importance of labor and management in the world of work.

Community Service

Promote and improve goodwill and understanding among all segments of the community through services donated by CTSO chapters, and to instill in its members a lifetime commitment to community service.

Business and Industry Connections

Increase awareness of quality job practices and attitudes, and increase the opportunities for employer engagement and eventual employment.

Financial Leadership

Plan and participate in activities to allow all members to carry out the chapter's projects. Educate members on their own personal finance for current and future success.

Public Relations

Make the general public aware of the good work that students in career and technical education are doing to better themselves and their community, state, nation, and world.

Social Activities

To increase cooperation in the school and community through activities that allow CTSO members to get to know each other in something other than a business or classroom setting.

Academic Standards Alignment

Colorado CTE courses are required to integrate academic and essential skills into course content. The following academic standards have been identified as aligned to this course and should be integrated throughout instruction.

[CO CTE Academic Standards Resources](#) (Rubrics, integration strategies, etc)

[CDE Colorado Academic Standards Online](#) (Online standards lookup tool)

Note: Mathematics and Science standards will be added in the future. Visit the [Colorado Academic Standards](#) page for the most current version.

Reading, Writing and Communicating Crosswalk

Course: Coding 1 - UAS

Colorado Academic Standards – Reading, Writing, and Communicating

Standard	Prepared Graduate Competency	Description	Course Evidence / Alignment
Standard 1: Oral Expression and Listening	Prepared Graduate 1 – Collaboration	Collaborate effectively as group members or leaders.	Students collaborate on coding challenges, discuss logic and errors, and provide and receive peer feedback.
Standard 1: Oral Expression and Listening	Prepared Graduate 2 – Oral Presentations	Deliver effective oral presentations for varied audiences.	Students explain code functionality, present projects or applications, and verbally walk through algorithms.
Standard 2: Reading for All Purposes	Prepared Graduate 4 – Informational Texts	Read a wide range of informational texts.	Students read programming documentation, tutorials, examples, problem statements, and specifications.
Standard 2: Reading for All Purposes	Prepared Graduate 5 – Language in Context	Understand how language functions in different contexts.	Students learn and apply domain-specific language such as syntax, commands, and logic, and interpret meaning across programming contexts.

Standard 3: Writing and Composition	Prepared Graduate 7 – Informational Writing	Craft informational/explanatory texts.	Students write code comments, explain algorithms and solutions, and document program behavior.
Standard 3: Writing and Composition	Prepared Graduate 9 – Writing Process	Demonstrate mastery of their own writing process.	The course reinforces drafting, testing, debugging, and refining, resulting in clean, correct, and readable code and explanations.
Standard 4: Research Inquiry and Design	Prepared Graduate 10 – Research and Ethics	Gather information from a variety of sources.	Students research solutions, syntax, and structures, evaluate sources for accuracy and relevance, and apply information ethically.

Free Resource Links

1. Career Exploration and Postsecondary Opportunities

- Bureau of Labor Statistics Engineering Careers (engineering careers drone careers job outlook education pathways)
<https://www.bls.gov/ooh/>
- FAA Unmanned Aircraft Systems (FAA rules certifications drone careers aviation pathways)
<https://www.faa.gov/uas>
- CareerOneStop Engineering Pathways (career exploration certifications education planning)
<https://www.careeronestop.org>
- NASA STEM Careers (STEM fields engineering workforce skills future jobs)
<https://www.nasa.gov/stem>

2. Computing, Coding, and the World of Drones

- Code.org Computer Science (coding fundamentals, programming concepts, real world applications)
<https://code.org>
- DroneBlocks Platform (drone programming, block coding, autonomous flight)
<https://droneblocks.io>
- DJI Education Hub (drone systems, sensors, navigation, applications)
<https://edu.dji.com>
- Scratch MIT Programming (intro coding, logic, simulations, beginner projects)
<https://scratch.mit.edu>

3. Ethical, Safe, and Responsible Programming

- Common Sense Digital Citizenship (ethics, online safety, responsible technology use)
<https://www.commonsense.org/education>
- CodeHS Cybersecurity and Ethics (ethical programming, data privacy, security awareness)
<https://codehs.com>
- Stanford Copyright and Fair Use (copyright law, intellectual property, fair use basics)
<https://fairuse.stanford.edu>
- Google Be Internet Awesome (internet safety, responsible online behavior)
<https://beinternetawesome.withgoogle.com>

4. Computational Thinking and Program Design

- Google Computational Thinking (problem solving, decomposition, abstraction, algorithms)
<https://edu.google.com>
- Lucidchart Flowchart Guide (flowcharts, diagramming, algorithm planning)
<https://www.lucidchart.com/pages/flowchart>
- Khan Academy Algorithms (algorithms, logic, problem solving basics)
<https://www.khanacademy.org/computing>

- CS Unplugged Activities (computational thinking, offline activities, problem solving)
<https://csunplugged.org>

5. Programming Fundamentals and Code Construction

- W3Schools Python Tutorial (syntax, variables, loops, conditionals, basics)
<https://www.w3schools.com/python>
- Replit Coding Environment (online coding, projects, collaboration, debugging)
<https://replit.com>
- CodeHS Programming Course (coding exercises, practice, structured learning)
<https://codehs.com>
- FreeCodeCamp Programming (projects, coding tutorials, debugging skills)
<https://www.freecodecamp.org>

6. Data, Files, and Information Management

- Khan Academy Data Basics (data storage, variables, arrays, organization)
<https://www.khanacademy.org>
- GeeksforGeeks File Handling (file input, output, data processing, programming)
<https://www.geeksforgeeks.org/file-handling-python>
- Google Sheets Training (data organization, spreadsheets, analysis skills)
<https://support.google.com/docs>
- Code.org Data Systems (data storage, management, computing systems)
<https://studio.code.org>

7. Testing, Debugging, and Quality Assurance

- MIT Scratch Debugging (debugging strategies, troubleshooting, logic errors)
<https://scratch.mit.edu>
- Guru99 Software Testing (testing concepts, debugging, quality assurance basics)
<https://www.guru99.com/software-testing.html>
- FreeCodeCamp Debugging Guide (debugging techniques, troubleshooting, coding errors)
<https://www.freecodecamp.org/news>
- Khan Academy Testing Concepts (testing, validation, logic correctness, verification)
<https://www.khanacademy.org>

8. Number Systems, Logic, and Computer Operations

- Khan Academy Binary and Hexadecimal (number systems, conversions, computing basics)
<https://www.khanacademy.org>
- Cisco Binary Game (binary practice, interactive learning, computing logic)
<https://www.netacad.com>
- All About Circuits Logic Gates (logic gates, computing circuits, AND OR NOT)
<https://www.allaboutcircuits.com>
- Crash Course Computer Science (computer systems, data processing, basics)
<https://www.youtube.com>

9. Applied Drone Programming Project

- DroneBlocks Projects (drone missions, programming, project based learning)
<https://droneblocks.io>
- NASA STEM Projects (engineering design, projects, problem solving activities)
<https://www.nasa.gov/stem>
- Code.org Project Builder (project planning, coding workflows, development)
<https://code.org>
- Arduino Project Hub (embedded systems, coding, electronics projects)
<https://projecthub.arduino.cc>

10. Leadership

- Barclays LifeSkills Leadership Lesson (leadership qualities, teamwork, communication, inclusive leadership)
<https://barclayslifeskills.com/help-others/lessons/leadership/>
- Lead4Change (student leadership, service learning, teamwork, goal setting)
<https://lead4change.org/>
- Share My Lesson Career and Technical Education Resources (leadership activities, career readiness, classroom lessons)
<https://sharemylesson.com/subject/career-and-technical-education>
- Worksheet Zone Leadership Skills Activities (leadership practice, collaboration, decision making, problem solving)
<https://worksheetzone.org/blog/leadership-skills-activities-for-students>

11. Work-Based Learning

- Colorado Work-Based Learning Definitions (job shadowing, internships, workplace tours, project-based learning)
<https://dhr.colorado.gov/work-based-learning-skills-based-hiring/state-agency-work-based-learning-definitions>
- SkillsUSA Work-Based Learning (career readiness, employability, workplace skills, reflection)
<https://www.skillsusa.org/students/work-based-learning/>
- NAF Work-Based Learning Experiences for High School Students (mentorships, internships, career training, confidence)
https://naf.org/news_articles/work-based-learning-experiences-for-high-school-students/
- NTACT:C Work-Based Learning Experiences (job shadowing, service learning, mentorship, workplace expectations)
<https://transitionta.org/topics/pre-ets/pre-ets-wble/>



Technology Student Association – National Competitive Events

Coding

Objective: *Students will demonstrate coding knowledge and develop a software solution to an on-site programming challenge aligned with TSA Coding competition requirements.*

- Study coding concepts such as syntax, data structures, control flow, and object-oriented programming.
- Prepare for an individual coding-concepts test used to determine semifinalist placement.
- Analyze an on-site coding problem, evaluation criteria, required tools, and programming constraints.
- Design and develop a software solution within the allotted competition time.
- Use an approved programming language, operating system, or API specified for the challenge.
- Test and debug the program to improve accuracy, functionality, and efficiency.
- Present the completed solution for objective testing and judge evaluation.

Resources

- Technology Student Association – Coding competition guidelines
<https://tsaweb.org/competitions-programs/tsa/themes-problems>
- Code.org – Computer science and programming lessons
<https://code.org/>
- freeCodeCamp – Programming tutorials and practice
<https://www.freecodecamp.org/>
- W3Schools – Programming references and examples
<https://www.w3schools.com/>



Technology Student Association – National Competitive Events

Software Development

Objective: *Students will design, develop, document, test, and present a software application that responds to the annual TSA Software Development challenge.*

- Analyze the annual software development challenge and identify the intended users, purpose, and functional requirements.
- Plan the software application using appropriate design tools such as pseudocode, flowcharts, diagrams, or wireframes.
- Develop a functioning software product using programming practices appropriate to the selected platform.
- Create documentation that explains the software purpose, development process, features, testing, and revisions.
- Apply quality assurance practices through testing, debugging, and refinement.
- Prepare a presentation or demonstration that explains the software solution and development decisions.
- Evaluate the final software using TSA criteria for functionality, design quality, documentation, usability, and presentation.

Resources

- Technology Student Association – Software Development competition guidelines
<https://tsaweb.org/competitions-programs/tsa/themes-problems>
- GitHub Docs – Version control and project collaboration
<https://docs.github.com/>
- Replit – Browser-based coding environment
<https://replit.com/>
- freeCodeCamp – Software development tutorials
<https://www.freecodecamp.org/>



Technology Student Association – National Competitive Events

Webmaster

Objective: *Students will design, build, launch, and explain a multi-page website that addresses the annual TSA Webmaster design challenge.*

- Research the annual Webmaster topic and identify the website’s purpose, audience, content needs, and design requirements.
- Plan the website structure using a sitemap, wireframes, page layouts, or other web design planning tools.
- Develop a multi-page website that communicates the team’s solution to the annual theme and problem.
- Apply web design practices related to navigation, accessibility, usability, visual consistency, and responsive layout.
- Launch the website on a publicly accessible web server according to TSA submission requirements.
- Test the website for functionality, links, layout, readability, and access issues before submission.
- Participate in an interview explaining the website, design choices, research, and development process.

Resources

- Technology Student Association – Webmaster competition guidelines
<https://tsaweb.org/competitions-programs/tsa/themes-problems>
- Mozilla Developer Network – HTML, CSS, and JavaScript documentation
<https://developer.mozilla.org/>
- W3Schools – Web development tutorials
<https://www.w3schools.com/>
- GitHub Pages – Website hosting for student projects
<https://pages.github.com/>