

Mobile App Development

Level: 3

Career Cluster: Digital Technology

Program: Information Technology

Pathway(s): Coding, Web Design

Course Catalog Description

This course introduces students to the design, development, testing, and evaluation of applications for mobile devices. Students apply programming, user interface design, problem-solving, and project development skills to create mobile app solutions while exploring industry tools, workflows, and responsible digital practices.

Summary

Introduces students to the design, planning, development, testing, and evaluation of applications for mobile devices. Students explore mobile platforms, user interface and user experience design, event-driven programming, app development workflows, device features, APIs, networking, security, privacy, and responsible digital practices.

Throughout the course, students analyze user needs, define project requirements, and apply industry-relevant development tools and processes to build applications that are functional, efficient, and responsive to real-world problems. Through hands-on projects, collaborative problem-solving, iterative design, and real-world application of technical skills, students create mobile app solutions while building career readiness, leadership, communication, and project management skills relevant to the information technology industry and postsecondary study. The course also encourages students to think critically about how mobile technologies can be designed to meet user needs and improve everyday experiences.

Notes

This revised course expands the scope by providing a more clearly organized and comprehensive sequence of student learning outcomes. In addition to foundational app development concepts, the updated course places stronger emphasis on user interface and user experience design, project planning and requirements analysis, event-driven programming, device features and APIs, networking and security, digital ethics and privacy, testing and deployment, work-based learning, leadership, and a final assessment project. Overall, the new version reflects a broader, more industry-aligned approach that strengthens both technical skills and career readiness. It also provides a more intentional progression from foundational concepts to applied development and presentation of a complete mobile app project.

Student Learning Outcomes

1. Career Exploration and Postsecondary Opportunities

Objective: *Students will analyze career pathways in information technology, 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 information technology
 - 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 information technology
 - a. Criticism and evaluation
 - b. Presentation
 - c. Assessment
- 3) Recognize laws, regulations, and ethics significant to the fields of information technology
- 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. Foundations of Mobile App Development

Objective: *Students will build foundational knowledge of mobile application development, including mobile platforms, app types, and development processes.*

- 5) Describe the purpose and function of mobile applications.
- 6) Differentiate between desktop and mobile applications.
- 7) Compare browser-based, native, and hybrid mobile applications.
- 8) Identify common mobile platforms and operating systems.
- 9) Describe the basic workflow of mobile application development.
- 10) Explain how mobile technologies have evolved over time.
- 11) Recognize the role of developer tools in mobile app creation.

3. User Interface and User Experience Design

Objective: *Students will design mobile interfaces that are functional, visually effective, and appropriate for intended users and devices.*

- 12) Identify features of effective mobile user interfaces.
- 13) Create interface layouts appropriate for a mobile device and intended purpose.
- 14) Design interfaces for browser-based, native, and hybrid applications.
- 15) Apply basic user experience principles to improve usability.
- 16) Analyze user needs when developing app screens and navigation.

- 17) Select appropriate interface components for specific application needs.
- 18) Revise interface designs based on testing or feedback.

4. App Planning, Requirements, and Development Workflow

Objective: *Students will analyze app requirements and plan development using structured workflows and project criteria.*

- 19) Identify project stakeholders and their perspectives.
- 20) Collect information to determine app requirements.
- 21) Analyze input, output, and processing requirements for a mobile application.
- 22) Define hardware and software specifications for a project.
- 23) Describe the stages of a mobile application development workflow.
- 24) Use time-management strategies to meet project deadlines.
- 25) Organize project tasks using basic planning tools or documentation.
- 26) Describe a proposed solution to a mobile app problem or need.

5. Coding, Logic, and Event-Driven Development

Objective: *Students will apply programming concepts and development tools to create functional mobile app components.*

- 27) Recognize the role of event-based programming in mobile applications.
- 28) Compare programming languages used in mobile app development.
- 29) Select an appropriate development environment for a project.
- 30) Use available libraries and components in app development.
- 31) Create basic mobile application components to meet identified needs.
- 32) Develop simple browser-based or device-based app features.
- 33) Apply logic and sequencing to produce functional app behavior.
- 34) Test and debug app features to improve performance.

6. Devices, APIs, Networks, and Security

Objective: *Students will evaluate how device features, connectivity, and security requirements affect mobile app design and performance.*

- 35) Explain how hardware limitations affect mobile app design decisions.
- 36) Describe how memory management affects mobile applications.
- 37) Explain how bandwidth and device mobility affect app performance.
- 38) Compare networks and their impact on mobile application development.
- 39) Identify mobile device controls and device-specific functions used by apps.
- 40) Describe the role of APIs in mobile application development.
- 41) Identify design strategies related to mobile network and device security.
- 42) Investigate security measures such as passwords, virus detection, and virus prevention.
- 43) Evaluate risks and benefits associated with mobile applications.

7. Digital Ethics, Privacy, and Responsible Development

Objective: *Students will demonstrate responsible and ethical practices when designing, developing, and sharing mobile applications.*

- 44) Explore intellectual property, privacy, copyright, and software licensing.
- 45) Model ethical acquisition and use of digital information.
- 46) Demonstrate proper digital etiquette and responsible software use.
- 47) Apply acceptable use policies when developing and sharing digital work.
- 48) Model respect for intellectual property in the use of graphics, media, and sound.
- 49) Describe how data privacy affects mobile app development.
- 50) Evaluate how app features may affect user safety and privacy.
- 51) Explain the importance of ethical decision-making in app design.

8. Testing, Simulation, and Deployment

Objective: *Students will test, refine, and prepare mobile applications for real-world use across devices and platforms.*

- 52) Use a simulation tool to emulate mobile device functionality.
- 53) Test applications on actual mobile devices.
- 54) Evaluate application performance across devices or environments.
- 55) Identify issues that affect usability, function, or compatibility.
- 56) Revise app components based on testing results.
- 57) Describe basic deployment and publishing considerations for mobile applications.
- 58) Prepare app assets and documentation for presentation or submission.

9. Work-Based Learning

Objective: *Students will apply mobile app development skills through real-world or workplace-connected tasks and project experiences.*

- 59) Describe how mobile applications are used to solve real-world problems.
- 60) Participate in project tasks connected to client, user, or community needs.
- 61) Apply app development skills to complete assigned workplace-style tasks.
- 62) Follow project requirements, timelines, and expectations.
- 63) Communicate progress and project needs with peers, instructors, or stakeholders.
- 64) Document tasks completed, tools used, and revisions made during a project.
- 65) Explain how classroom learning connects to real-world app development.

10. Leadership in App Development

Objective: *Students will demonstrate leadership, collaboration, and responsibility while contributing to app development projects.*

- 66) Describe the role of leadership in team-based development projects.
- 67) Demonstrate teamwork during collaborative design and development tasks.
- 68) Communicate respectfully and effectively with team members.
- 69) Take responsibility for assigned tasks and deadlines.
- 70) Support group problem-solving and decision-making.
- 71) Demonstrate organization and time management during project work.
- 72) Reflect on personal contributions to a team or project.

11. Final Assessment Project

Objective: *Students will apply course knowledge and technical skills to design, develop, test, and present a mobile application project.*

- 73) Plan a mobile application project based on a defined purpose or need.
- 74) Design a user interface appropriate for the intended audience and device.
- 75) Develop functional app components using appropriate tools and techniques.
- 76) Test the application using simulation tools or mobile devices.
- 77) Revise the application based on testing results or feedback.
- 78) Present the completed project and explain the design and development process.
- 79) Evaluate the strengths and limitations of the final product.
- 80) Reflect on skills gained and areas for future improvement.

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: Mobile App Development

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 work in teams to plan, design, and develop mobile applications, manage workflows, and incorporate peer feedback during the development process.
Standard 1: Oral Expression and Listening	Prepared Graduate 2 – Oral Presentations	Deliver effective oral presentations	Students explain app concepts, design decisions, development processes, and completed applications to peers and instructors.
Standard 2: Reading for All Purposes	Prepared Graduate 4 – Informational Texts	Read a wide range of informational texts	Students read and interpret technical documentation, APIs, platform guidelines, project specifications, and development resources.

Standard 2: Reading for All Purposes	Prepared Graduate 5 – Language in Context	Understand how language functions in different contexts	Students use precise technical vocabulary and adapt explanations of mobile app functionality for technical and non- technical audiences.
Standard 3: Writing and Composition	Prepared Graduate 7 – Informational Writing	Craft informational/explanatory texts	Students create written documentation such as app requirements, design descriptions, code comments, and deployment instructions.
Standard 3: Writing and Composition	Prepared Graduate 9 – Writing Process	Demonstrate mastery of the writing process	Students revise and refine technical documentation and code-related writing to produce clear, coherent, and professional products.
Standard 4: Research Inquiry and Design	Prepared Graduate 10 – Research and Ethics	Gather and ethically use information	Students research mobile technologies, platforms, libraries, and user requirements; evaluate sources; and apply information responsibly.

Free Resource Links

1. Career Exploration and Postsecondary Opportunities

- Android Developers Teach Android Development (careers, classroom curriculum, Kotlin, industry pathways)
<https://developer.android.com/teach>
- MIT App Inventor Teach Resources (teaching resources, app careers, classroom support, mobile computing)
<https://appinventor.mit.edu/explore/library>
- AppInventor.org Course In A Box (teacher lessons, beginner app development, career exploration, course materials)
<http://mit.appinventor.org/>
- Android Developers Training Courses (learning pathways, certification preparation, Android careers, skill development)
<https://developer.android.com/courses/>

2. Foundations of Mobile App Development

- Android Developers Guide (mobile platforms, app types, development processes)
<https://developer.android.com/guide>
- Apple Xcode Documentation (iOS development, mobile tools, app architecture)
<https://developer.apple.com/documentation/xcode>
- MIT App Inventor Tutorials (beginner app development, visual coding, mobile basics)
<https://appinventor.mit.edu/explore/ai2/tutorials>
- Google Mobile App Development Overview (mobile ecosystems, development concepts, platforms)
<https://developers.google.com/learn/topics/mobile>

3. User Interface and User Experience Design

- Google Material Design (UI design, UX principles, mobile interface guidelines)
<https://material.io/design>
- Apple Human Interface Guidelines (mobile UX, interface design, accessibility)
<https://developer.apple.com/design/human-interface-guidelines>
- UX Design Foundation Basics (user experience principles, usability, interface design)
<https://www.interaction-design.org/literature/topics/ux-design>
- Nielsen Norman Group UX Articles (usability, interaction design, design research)
<https://www.nngroup.com/articles>

4. App Planning Requirements and Development Workflow

- Atlassian Agile Project Management Guide (project planning, workflows, agile development)
<https://www.atlassian.com/agile>

- Google Project Management Fundamentals (project planning, task organization, workflows)
<https://grow.google/certificates/project-management>
- Lucidchart Flowchart Resources (workflow diagrams, system design, planning tools)
<https://www.lucidchart.com/pages/flowchart>
- Trello Guide for Students (task management, project organization, collaboration)
<https://trello.com/guide>

5. Coding Logic and Event Driven Development

- W3Schools JavaScript Tutorial (programming, logic, event-driven coding)
<https://www.w3schools.com/js>
- freeCodeCamp JavaScript Curriculum (coding fundamentals, logic, problem solving)
<https://www.freecodecamp.org/learn>
- Khan Academy Programming Basics (coding logic, interactive programming, fundamentals)
<https://www.khanacademy.org/computing/computer-programming>
- MDN Web Docs JavaScript Guide (events, programming concepts, scripting)
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

6. Devices APIs Networks and Security

- Google Developers API Guide (APIs, integration, mobile services)
<https://developers.google.com/apis-explorer>
- MDN Web APIs Documentation (device features, APIs, browser integration)
<https://developer.mozilla.org/en-US/docs/Web/API>
- Cisco Networking Basics (network fundamentals, connectivity, mobile networks)
<https://www.netacad.com/courses/networking-basics>
- OWASP Mobile Security Resources (mobile security, risk management, best practices)
<https://owasp.org/www-project-mobile-top-10>

7. Digital Ethics Privacy and Responsible Development

- Common Sense Education Digital Citizenship (ethics, privacy, responsible technology use)
<https://www.commonsense.org/education/digital-citizenship>
- Electronic Frontier Foundation Privacy Guide (data privacy, security, digital rights)
<https://ssd EFF.org>
- Creative Commons Licensing Guide (copyright, intellectual property, licensing)
<https://creativecommons.org/about/cclicenses>
- Google Safety Center (data protection, security, responsible technology)
<https://safety.google>

8. Testing Simulation and Deployment

- Android Testing Documentation (app testing, debugging, performance)
<https://developer.android.com/training/testing>
- Apple App Store Connect Guide (deployment, publishing, app management)
<https://developer.apple.com/app-store-connect>
- BrowserStack App Testing Resources (cross-platform testing, simulation, device testing)
<https://www.browserstack.com/guide/mobile-app-testing>
- Firebase App Distribution (testing workflows, deployment, performance monitoring)
<https://firebase.google.com/products/app-distribution>

9. Work Based Learning

- Advance CTE Resource Center (work-based learning, career readiness, industry connections)
<https://careertech.org/resource-center>
- OCTAE Work Based Learning Toolkit (workplace learning, employer partnerships, program design)
<https://octae.ed.gov/wbltoolkit>
- CareerOneStop Explore Careers (career pathways, job skills, workforce readiness)
<https://www.careeronestop.org>
- O*NET Career Exploration Tools (occupations, skills, job requirements, workforce data)
<https://www.onetonline.org>

10. Leadership in App Development

- Lead4Change Leadership Program (leadership development, teamwork, project-based learning)
<https://lead4change.org>
- Barclays LifeSkills Leadership Lessons (leadership skills, communication, collaboration)
<https://barclayslifeskills.com/help-others/lessons/leadership>
- SkillsUSA Framework Resources (leadership skills, workplace readiness, teamwork)
<https://www.skillsusa.org/competitions/skillsusa-framework>
- MindTools Leadership Skills Library (leadership strategies, communication, decision making)
https://www.mindtools.com/pages/main/newMN_LDR.htm

11. Final Assessment Project

- MIT App Inventor Project Gallery (student projects, app examples, inspiration)
<https://appinventor.mit.edu/explore/gallery>
- Google Developer Case Studies (real-world apps, project examples, development insights)
<https://developers.google.com/case-studies>

- Code.org App Lab Projects (student app development, project examples, coding applications)
<https://studio.code.org/projects/applab>
- GitHub Student Developer Pack (project hosting, collaboration, version control)
<https://education.github.com/pack>



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 TSA 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, wireframes, or interface mockups.
- Develop a functioning software product using programming practices appropriate to the selected platform.
- Create documentation that explains the software purpose, development process, features, code structure, 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/>
- MIT App Inventor – Mobile app development platform
<https://appinventor.mit.edu/>



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/>



Technology Student Association – National Competitive Events

Video Game Design

Objective: *Students will design, develop, test, document, and present an original video game that meets TSA Video Game Design requirements for gameplay, technical quality, creativity, documentation, and presentation.*

- Research the annual TSA Video Game Design theme and develop an original game concept.
- Plan gameplay, rules, goals, mechanics, user interaction, scoring, levels, and technical requirements.
- Develop a playable game using appropriate programming, game engine, or interactive media tools.
- Apply programming concepts such as variables, conditionals, loops, functions, objects, collision logic, and user input.
- Test and refine the game to improve functionality, usability, balance, clarity, and player experience.
- Prepare documentation that explains the game concept, development process, technical tools, testing, and revisions.
- Present the game to judges and explain design decisions, technical implementation, and response to the annual theme.

Resources

- Technology Student Association – Video Game Design competition guidelines
<https://tsaweb.org/competitions-programs/tsa/themes-problems>
- Unity Learn – Game development tutorials
<https://learn.unity.com/>
- Godot Engine Documentation
<https://docs.godotengine.org/>
- MIT App Inventor – Mobile app development platform
<https://appinventor.mit.edu/>