Computer Science (2026-2027)

Cosby High School

Part 1: Course Information

Instructor Information

Course: Computer Science
Instructor: Dr. Donna Jones
School Telephone: 423-487-5602
E-mail: jonesd1@cocke.k12.tn.us
ParentSquare: Use the search function

Course Overview and Goals

The Tennessee Computer Science Foundations course is intended to introduce students to the fundamentals of computer science. The course is designed to cover the Tennessee High School Computer Science standards. The course explores the fundamentals of computer science including the impact of technology on society, cybersecurity, data analysis, and programming. Throughout the course, students will use JavaScript as they explore basic programming concepts such as variables, loops, conditional statements, and lists.

Prerequisite

- none

General Education/High School Pathway Area

- This course can be used for the required fine arts credit for graduation.

Textbook & Course Materials

Required Text

- codehs.com

Required Online Course Platforms

- Canvas
- Clever
- ParentSquare

Course Requirements

- none

Course Structure

This course is a hands-on learning classroom. Students will participate in lectures, discussions, activities, and lab-based learning.

Computer Science (2026-2027)

Learning Environment: The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, example programs to explore, and written programming exercises, adding up to over 90 hours of hands-on programming practice in total. Units end with a comprehensive unit test that assesses students' mastery of the material from that unit as well as challenge problems where students can display their understanding of the material.

Assessment Methods (point system)

- Check for Understanding
- Free Response
- Projects
- Coding Activities

Remote Work:

- To be determined

Midterm/Final Exam

- To be determined

Part 2: Student Learning Outcomes

CS.AT: Algorithmic Thinking

1) Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
2) Systematically design and develop programs for broad audiences by incorporating feedback from users.
3) Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
4) Use effective communication and accurate computer science terminology to explain problem solving when completing a task.

CS.DA: Data Analysis

1) Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.
2) Utilize data to answer a question using a variety of computing & data visualization methods.
3) Use data analysis tools and techniques to identify patterns in data representing complex systems.

CS.NI: Networking and the Internet

1) Explain the tradeoffs when selecting and implementing cybersecurity recommendations.
2) Identify laws regarding the use of technology and their consequences and implications.
3) Evaluate strategies to manage digital identity and reputation with awareness of the permanent impact of actions in a digital world.
4) Demonstrate how to apply techniques to mitigate effects of user tracking methods.
5) Show an understanding of the ramifications of end-user license agreements and terms of service associated with granting rights to personal data and media to other entities.
6) Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.
7) Demonstrate a fundamental understanding of API (Application Programming Interface).

CS.PC: Programming Concepts

1) Choose and apply an appropriate iterative design process to systematically test and refine a program to increase performance.
2) Develop a plan to manage and assign data values of different types (strings, numeric, character, integer, and date) to a variable
3) Create and refine programs with Boolean conditionals to demonstrate the use of branches and logical operators.
4) Design and develop iterative programs that combine control structures, including nested loops and compound conditionals.
5) Create parameters to organize a program to make it easier to follow, test, and debug.
6) Incorporate existing code, media, and libraries into original programs, and give proper attribution.

7) Debug (identify and fix) errors in an algorithm or program that includes sequences and simple and complex loops following a two-step debugging process.

CS.IC: Impacts of Computing

1) Evaluate and debate the social and economic implications of computing in the context of safety, law, and ethics.
   a) Discuss the ethical ramifications of hacking and its impact on society.
   b) Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users (Bots, Chatbots, Spiders or Crawlers, Web Scraping, keyloggers etc.).
   c) Explain the positive and negative consequences that intellectual property laws can have on innovation.
2) Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.
3) Research the impact of computing technology on possible education and career pathways.
4) Predict how computational innovations that have revolutionized aspects of our culture might evolve.

Part 3: Topic Outline/Schedule

Unit 1: Our Digital World (3 weeks/15 hours)

This module includes topics on Internet etiquette and how to stay safe on the world wide web. It will also look at the potential effects of our digital footprints and how to navigate the legal aspects of the internet, including copyright laws, terms of service, and privacy policies. Finally, the module looks forward by exploring careers in computer science and the future of technology.

Objectives/Topics Covered	● Digital Footprint and Reputation ● Cyberbullying ● Internet Safety ● Creative Credit and Copyright ● Terms of Service & Privacy Policies ● Computer Science Careers ● Future of Computing
Example Assignments/Labs	● Digital Footprint and Reputation ● Cyberbullying ● Internet Safety ● Creative Credit and Copyright ● Terms of Service & Privacy Policies ● Computer Science Careers ● Future of Computing ● Project: Create a Public Service Announcement

Unit 2: Cybersecurity (2 weeks/8 hours)

This module includes topics on cybersecurity, including an introduction to the CIA Triad. Students start by understanding the threats and how they can be hacked, and then armed with this knowledge, they look at ways to protect themselves.

Objectives/Topics Covered	● What is Cybersecurity? ● The CIA Triad ● Security ● Hacking Ethics ● Web Tracking
Example Assignments/Labs	● What is Cybersecurity? ● The CIA Triad ● Security ● Hacking Ethics ● Web Tracking

Unit 3: Data (2 weeks/8 hours)

This module includes topics on data and data science. Students start by understanding what data is available and how it can be organized and presented. From there, they get into looking at different ways to visualize the data and understand the limitations of their data.

Objectives/Topics Covered	● What is Data Science? ● Information Literacy ● Gathering Data ● Visualizing and Interpreting Data ● Data Collection and Limitations
Example Assignments/Labs	● What is Data Science? ● Information Literacy ● Gathering Data ● Visualizing and Interpreting Data ● Data Collection and Limitations

Unit 4: Introduction to Programming with Karel the Dog (4 weeks/20 hours)

Module four is programming with Karel. Karel is a dog that only knows how to move, turn left, and place tennis balls in his world. You can give Karel these commands to instruct him to do certain things. We use Karel to show you what it means to program and allow you to focus on problem solving.

Objectives/Topics Covered	● Commands ● Defining vs. Calling Methods ● Designing methods ● Program entry points ● Control flow	● Looping ● Conditionals ● Classes ● Commenting code ● Preconditions and Postconditions ● Top-Down Design
Assignments/Labs	● 26 Karel Programming Exercises and Challenges in total ● Program-specific tasks for Karel the Dog ● Teach Karel new commands like turnRight() or makePancakes() ● Solve large Karel problems by breaking them down into smaller, more manageable problems using Top-Down Design ● Using control structures and conditionals to solve general problems	

Unit 5: JavaScript & Graphics (1.5 weeks/7 hours)

Introduces you to the basics of JavaScript, including variables, user input, mathematics, basic graphics, and image representations.

Objectives/Topics Covered	● Variables ● User Input ● Arithmetic Expressions ● Graphics ● Image Representation
Assignments/Labs	● 7 JavaScript & graphics programming exercises in total ● Using variables and getting user input using JavaScript ● Create basic graphics in JavaScript

Unit 6: JavaScript Control Structures (3 weeks/15 hours)

Learn how to use control structures such as if/else statements and loops to make more advanced programs in JavaScript.

Objectives/Topics Covered	● Booleans ● For Loops ● Conditionals ● Nested Control Structures ● While Loops
Assignments/Labs	● 22 control structures programming exercises in total ● Using comparison and logical operators to control the flow of the program ● Using for loops ● Drawing basic graphics using control structures

Unit 7: Functions, Events, & Lists (5 weeks, 25 hours)

Learn to write reusable code with functions and parameters.

Objectives/Topics Covered	● Functions with and without parameters ● Functions with and without return values ● Nested Control Structures ● Local variables and scope	● Timers ● Randomizing Games ● Mouse Events ● Keyboard Events ● APIs
Assignments/Labs	● 32 functions programming exercises in total	

Unit 8: Final Project (2-4 weeks, 10-20 hours)

In this module you'll take all the skills you've learned throughout the course to create a final program.

Computer Science (2026-2027)

Objectives/Topics Covered	● Project Planning ● Final Project Creation ● Testing and peer feedback
Assignments/Labs	● Collaborative open-ended final project which encourages creativity ● Program Requirements: ○ must use JavaScript Graphics ○ must allow the user to interact with your project with either their mouse of keyboard ○ must use at least one timer ○ must break down the program into multiple functions ○ must utilize control structures where applicable

Part 4: Grading Policy

Graded Course Assignments

- A point system will be implemented in this course.
- Please check your grades often on Aspen at cockecountyschools.org
- If you do not know your Aspen login information, please see me ASAP.

***Disclaimer: Assignments may change at any time.

Technical Snafus

This class will depend on access to computers and the Internet. Throughout the semester, you will have technology issues. For example, your laptop will crash, the Wi-Fi will go down, or you will forget your username/password. These issues are part of twenty-first-century life, not emergencies.

To succeed in college and your career, work habits that take such snafus into account need to be developed. These problems are solely your responsibility to determine the proper steps to ensure your work will not be lost; if one device or service is not operational, locate one that will. I will not provide you with an extension based on problems you may be having.

Grading

The final grade will be determined by the total points earned. The following grading scale will be used:

90-100 = A
80-89 = B
70-79 = C
60-69 = D
Below 60 = F

Extra Credit

- No extra credit will be provided. If students come to school every day, make a substantial effort during class, turn in completed work on time, and pass all tests. Students will not need extra credit.

Part 5: Course Policies/Rules

Rules

- Come to class on time
- Be ready to learn (including bringing all required materials)
- Listen and follow directions
- Be respectful to others
- Believe in yourself
- Be aware of and follow school rules

Academic Dishonesty Policy

People learn most effectively and build their strength of character by doing their reading, writing, test-taking, projects, research, and assignments. Students learn most from their education by evaluating, reflecting, and revising their work. Therefore, educating students about academic honesty and clarify the school's policy on academic dishonesty.

Definitions of cheating and plagiarism:

Cheating

Cheating is the unauthorized possession, giving, sharing, taking, or presentation of information and material benefits to a student.
Examples of cheating include, but are not limited to

- the passing of information during an assessment,
- having access to and utilizing unauthorized material and technology during an assessment,
- passing information about an exam from one class to another, and
- submitting work that is not one's own.

Plagiarism

According to the Harbrace Handbook, the 15th edition:

- Plagiarism is defined as “presenting someone else’s ideas, research, or opinions as your own without proper documentation, even if it has been rephrased.”
- This includes, but is not limited to:
  - Copying verbatim all or part of another’s written work;
  - Using phrases, figures, or illustrations without citing the source;
  - Paraphrasing ideas, conclusions, or research without citing the source;
  - Using all or part of a literary plot, poem, or film without attributing the work to the creator.
- Consequences of Plagiarism
  - Plagiarism is a form of stealing and academic fraud. Students who are found guilty of plagiarism have the option of redoing the assignment within a specified period, accepting a letter drop, or taking a zero on the assignment. Parents are to be involved in making the decision.

Artificial Intelligence

Artificial Intelligence (AI) programs, as defined by state law, may be used by staff and students in the district.

Academic Integrity

Students shall be instructed on responsible use standards, including but not limited to the following:

1. Effective use of generative AI;
2. Guidelines for when to appropriately use AI in assignments.
3. How to determine whether AI responses are accurate.
4. Users assume responsibility for incorporating AI content responsibly; and
5. The difference between cheating and seeking support.

Religion in the Classroom

The Board affirms that it is essential that the teaching about religion - and not of a religion- be conducted in a factual, objective, and respectful manner in accordance with the following guidelines:

1. Religious themes may be part of the curriculum for school-sponsored activities and programs, provided they are essential to the learning experience in the various fields of study and are presented objectively.
2. The inclusion of religion shall be for educational purposes only.
3. The emphasis on religious themes should be only as extensive as necessary for a balanced and comprehensive study of the curriculum. Such studies shall never be used to proselytize, establish, foster, or demean any particular religion, religious tenets, or beliefs; and
4. Student-initiated expressions to questions or assignments that reflect their beliefs or non-beliefs about a religious theme shall be accommodated.