

STRANDS AND STANDARDS

EXPLORING COMPUTER SCIENCE



Course Description

Exploring Computer Science is designed to introduce students to the breadth of the field of computer science through an exploration of engaging and accessible topics. The course focuses on the conceptual ideas of computing and helps students understand why certain tools or languages might be utilized to solve particular problems. The goal of Exploring Computer Science is to develop in students the computational thinking practices of algorithm development, problem solving, and programming within the context of problems that are relevant to the lives of today's students. Students will also be introduced to topics such as artificial intelligence, web development, programming, and physical computing.

Intended Grade Level	9-12
Units of Credit	0.5
Core Code	35.02.00.00.007
Concurrent Enrollment Core Code	None
Prerequisite	None
Skill Certification Test Number	802
Test Weight	0.5
License Area of Concentration	CTE and/or Secondary Education 6-12
Required Endorsement(s)	
Endorsement 1 or	Cybersecurity
Endorsement 2 or	Information Technology Systems
Endorsement 3 or	Intro to Computer Science
Endorsement 4 or	Multimedia
Endorsement 5 or	Programming & Software Development
Endorsement 6	Web Development

STRAND 1

Computer Science Practices

Students will employ the following practices throughout the course. They provide a framework and serve as helpful reminders of the high-level skills and dispositions computer scientists should be continually developing.

Standard 1

Critical Thinking

- Use the structured problem-solving process to help address new problems
- View challenges as solvable
- Decompose or break down larger problems into smaller components

Standard 2

Persistence

- Expect and value mistakes as a natural and productive part of problem solving
- Continue working with new ideas and consider multiple possible approaches
- Iterate and continue to improve partial solutions

Standard 3

Creativity

- Incorporate personal interests and ideas into activities and projects
- Experiment with new ideas and consider multiple possible approaches
- Extend or build upon the ideas and projects of others

Standard 4

Collaboration

- Work with others to develop solutions that incorporate all contributors
- Mediate disagreements and help teammates agree on a common solution
- Actively contribute to the success of group projects

Standard 5

Communication

- Structure work so that it can be easily understood by others
- Consider the perspective and background of your audience when presenting work
- Provide and accept constructive feedback in order to improve work

STRAND 2

Problem Solving with Computers

Students will learn how computers input, output, store, and process information. Students will gain the importance of solving problems, and/or automating tasks with the aid of computers, as well as a basic understanding of the algorithms computers use.

Standard 1

Students will understand computer hardware and the tasks they perform:

- Students will identify required functions for a device to be classified as a computer (receives input, processing; output; storage)
 - Analyze the characteristics of hardware components including processor, operating system, RAM, ROM, hard drive, and input and output devices.
 - Understand the relationship between bits and bytes
 - Compare and convert between the following sizes: kilobyte, megabyte, gigabyte, terabyte.
 - Discuss the speed of the computer (gigahertz).
- Students will identify examples of tasks that can and cannot be accomplished with a computer.

Standard 2

Students will describe changes technology has made on communication, privacy, and social interactions.

- Impacts of technology on society from the following perspectives: social, economic, political, legal, ethical and moral issues
- Permanence of online information
- Consider issues around privacy and collection of data
- Methods of communication appropriate for different situations including appropriate use of social media
- Online safety

Standard 3

Students will discuss how and why binary is used to represent data in a computer.

- Describe how binary digits (bits) are stored in different media
- Understand the binary system or pattern for counting up to 8 digits
- Convert numbers between 0 and 128 from decimal to binary and vice versa.
- Describe how real-world phenomena such as numbers, characters (ASCII), or images (RGB) are digitized and represented in a computer.

Standard 4

Students will understand different algorithms used in problem solving.

- Solve a problem through an iterative process.
 1. **Define** - Understand the Problem
 2. **Prepare** - Plan the Solution (design via pseudocode/flowcharts)
 3. **Try** - Carry out the Plan (Code)
 4. **Reflect** - Review and Discuss your Solution (Testing / Feedback)
 5. **Repeat** - Reiterate through the steps until the problem is solved
- Explain when a binary search would be more efficient than a linear search

- Visualize and compare common sorting algorithms (e.g. insertion, selection, bubble, quicksort, merge sort)

Standard 5

Students will gain knowledge and skills while considering the social, moral, and ethical impacts of Artificial Intelligence (AI) systems and usage.

- Students will explain the idea of intelligence specifically as it relates to computers.
- Students will explain what it means for a machine to learn (Turing Test)
- Students will identify the AI being used, such as image recognition, speech recognition, translation.
- Students will train and test an existing AI system (machine learning).
- Students will explore and explain the social and ethical impacts of AI (human and algorithmic bias, worker obsolescence through automation, user interface improvements, human/machine augmentation, etc.)
- Students will gain an understanding of how AI is changing different sectors such as medicine, agriculture, manufacturing, etc.

STRAND 3

Web Development

Students will learn social responsibility and ethics with regard to web development and how to use the basic building blocks of the World Wide Web: HTML5 and Cascading Style Sheets (CSS). Students will follow the steps to create a website by planning, designing, and coding a personal website.

Standard 1

Social Responsibility of Website Development

- Students will understand ethical behavior as it relates to an AUP, Intellectual Property, Netiquette, Respecting Privacy, Anti-Spamming Laws, etc.
- Students will demonstrate knowledge of standard copyright rules.
 - Understand copyright for original creations.
 - Understand the creative commons license
 - Understand when to obtain permission for non-original work.
- Students will identify the use and purpose of acceptable use policy (AUP).
 - Comply with the school's AUP

Standard 2

Design Process

Students will understand the need to know the purpose of website design in relationship to the intended audience and client needs.

- Students will plan a website design using storyboards, sketches, or wireframes (rough drafts).

Standard 3

HTML

Students will understand that the HTML programming language is used to create all websites on the internet and acts as the structure for a website.

- Students will code the foundation for a basic webpage including the element tags DOCTYPE, html, head, title, and body.
- Students will create pages with tags and attributes at the inline level. (DOCTYPE, title, head, body, h1, h2, h6, p, br, etc.)
- Students will create web pages with text formatting, links, images, and lists.

Standard 4

CSS

Students will understand that CSS (Cascading Style Sheets) are used to customize the style or looks of a website.

- Students will apply CSS to a website.
 - Apply CSS to an element using an inline style. (An inline style may be used to apply a unique style for a single element.)
 - Apply CSS to a website using an external stylesheet. (Best Coding Practice - One file changes the entire website.)
- Students will format web pages using CSS
 - Modify background properties such as color and image.
 - Modify font properties such as font-family, size, and color.
 - Modify border properties such as width, style, and color.
 - Implement tags and classes to modify an HTML element.

Standard 5

Careers in Web Development

Students will explore various careers in Web Development including front end developer, back end developer, full stack developer, and UX/UI designer.

STRAND 4

Programming and Algorithms

Students will understand that an algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are then translated into programs, or code, to provide instructions for computing devices. Programs control all computing systems and empower people to communicate with the world in new ways and solve compelling problems.

Standard 1

Program Design

Students will identify how planning strategies (such as flowcharts, storyboards, prototypes or pseudocode) are used when creating a program.

Standard**Algorithms**

Define an algorithm as a set of clearly defined, logical steps to solve a problem.

- Students will describe the steps needed to efficiently solve a non-computing problem using a pseudocode algorithm
- Students will examine traditional programming algorithms such as searches, sorts, and minimal spanning trees.
- Students will examine and formulate algorithms that solve specific problems.

Standard 3**Input/Output**

Students will recognize a variety of different user input sources such as text input, sensors, mouse response, movement, or event. Students will recognize a variety of different outputs such as sounds, light, vibrations, movement, text and/or graphics.

Standard 4**Variables**

- Students will understand that variables are named locations in memory.
- Students will be able to identify variables and when they should be used in code.

Standard 5**Loops**

Students will understand that programs use loops (iteration) to be more efficient and avoid code duplication.

Standard 6**Conditionals**

Students will understand that programs use conditionals to perform different computations or actions based on whether a condition is true or false (booleans).

Standard 7**Operators**

Students will understand that programs use mathematical symbols (+, -, *, /, >, <, ==, AND, OR) in a program to perform specific operations (mathematical, relational, or logical) and produce a single result.

Standard 8**Functions**

Students will understand that a function is a named block of code that performs a specific task. Functions encourage efficiency, reusability, and readability.

Standard 9**Debugging**

Students will understand that debugging is finding and removing errors from a program so it can operate as intended. Strategies students might learn for debugging could include:

- Guess and Check
- Deactivating sections to identify problematic code
- Looking for typos, missing tags, or incorrect syntax
- Making the problem smaller - identifying important points (changing variable values, getting input, etc.)
- Asking a friend or team member for help
- Printing, watching, or changing variable values while the program runs
- Using a debugging tool
- Thinking about when the code last worked and what has been added since then

Standard 10

Physical Computing

Students will demonstrate an understanding of the relationship between hardware and software.

- Students will define and explain an algorithm for a physical computing device. (sequence of instructions processed by the device.)
- Students will create a prototype of a physical computing device that uses algorithms to solve a computational problem.
- Students will create a physical project or program a physical device
 - Students will illustrate ways the project or physical device implements logic, input, and output through hardware components (sensors, buttons, switches, etc.)
 - Students will systematically identify and fix problems with the project or physical device.

Standard 11

Careers in Programming

Students will explore various careers in programming such as Software Engineer, Video Game Developer, Mobile App Developer, and Web Developer.

Performance Skills

- Students will design algorithms and create programming solutions to a variety of computational problems using a block or text programming language.
- Students will develop a program or programs that:
 - makes a decision based on data or user input (conditionals).
 - accepts user and/or sensor input and stores the result in a variable.
 - uses variables that represent different data types.
 - uses structures that repeat blocks/lines of code (loops).
 - uses operators.
 - uses functions.
- Students will analyze, test, improve and debug computer programs.

- Students will design and create a physical project or program a physical device and debug the project or device

Workplace Skills

- Communication
- Problem Solving
- Teamwork
- Critical Thinking

Skill Certificate Test Points by Strand

Test Name	Test #	Number of Test Points by Strand										Total Points	Total Questions
		1	2	3	4	5	6	7	8	9	10		
Exploring Computer Science (ESC)	802	0	19	14	14							47	34

Glossary

Strand 2 - Problem Solving with Computers

Algorithm: A list of steps to finish a task.

ASCII: a common numeric representation of a text character (American Standard Code for Information Interchange)

Artificial Intelligence: a branch of computer science dealing with the simulation of intelligent behavior in computers

Augmentation: the action or process of making or becoming greater in size or amount.

Automation: the use of electronics and computer-controlled devices to assume control of processes

Bias: prejudice in favor of or against one thing, person, or group compared with another.

Binary: A way of representing information using only two options.

Binary Search: compares the target value to the middle element of the array. If they are not equal, the half in which the target cannot lie is eliminated and the search continues on the remaining half, again taking the middle element to compare to the target value and repeating this until the target value is found.

Bit: the single unit of information in a computer, typically represented as a 0 or 1.

Byte: The most common fundamental unit of digital data eg. Kilobyte, Megabyte, etc. A single byte is 8 bits-worth of data.

Decimal: a term that describes the base-10 number system, probably the most commonly used number system. The decimal number system consists of ten single-digit numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.

Digital Footprint: The collected information about an individual across multiple websites on the Internet.

Gigabyte: a unit of information equal to 1024 megabytes or 2^{30} (1,073,741,824) bytes

Gigahertz: a unit of alternating current (AC) or electromagnetic (EM) wave frequency equal to one billion hertz (1,000,000,000 Hz) in some computers, to express microprocessor clock speed.

Hertz/Gigahertz: a unit measuring the waves or frequencies of electric changes each second, such as the clock rate of a computer CPU.

Input: the action of entering data into your computer. This can be text typed in a word processing document, keywords entered in a search engine's search box, or data entered into a spreadsheet. Input can be something as simple as moving the mouse or clicking the mouse button or it can be as complex as scanning a document or downloading photos from a digital camera.

Kilobyte: is the smallest unit of measurement greater than a byte; a kilobyte is 10^3 or 1,000 bytes.

Linear Search: process in which a sequential search is made over all items one by one. Every item is checked and if a match is found then that particular item is returned, otherwise the search continues till the end of the data collection.

Megabyte: is equal to 1,000 kilobytes; a megabyte is 10^6 or 1,000,000 bytes.

Operating system: is software that communicates with the hardware and allows other programs to run. It is comprised of system software, or the fundamental files your computer needs to boot up and function. Every desktop computer, tablet, and smartphone includes an operating system that provides basic functionality for the device; common desktop operating systems include Windows, OS X, and Linux.

Output: what the computer produces based on user input. This includes data produced at a software level, such as the result of a calculation, or at a physical level, such as a printed document. A basic example of software output is a calculator program that produces the result of a mathematical operation. A more complex example is the results produced by a search engine, which compares keywords to millions of pages in its Web page index.

Pixel: is actually short for "Picture Element." These small little dots are what make up the images on computer displays, whether they are flat-screen (LCD) or tube (CRT) monitors. The screen is divided up into a matrix of thousands or even millions of pixels. Typically, you cannot see the individual pixels, because they are so small. This is a good thing, because most people prefer to look at smooth, clear images rather than blocky, "pixelated" ones.

Privacy: is a branch of data security concerned with the proper handling of data – consent, notice, and regulatory obligations. More specifically, practical data privacy concerns often revolve around:

- Whether or how data is shared with third parties.
- How data is legally collected or stored.
- Regulatory restrictions such as GDPR, HIPAA, GLBA, or CCPA.

Processor: is a small chip that resides in computers and other electronic devices. Its basic job is to receive input and provide the appropriate output.

Processing: a series of operations on data, especially by a computer, to retrieve, transform, or classify information.

RAM: "Random Access Memory" is a common hardware component found in electronic devices, including desktop computers, laptops, tablets, and smartphones. The amount of RAM in a device determines how much memory the operating system and open applications can use. When a device has sufficient RAM, several programs can run simultaneously without any slowdown. When a device uses close to 100% of the available RAM, memory must be swapped between applications, which may cause a noticeable slowdown.

RGB: a numeric representation of a color based on Red, Green, and Blue values.

ROM: "Read-Only Memory" is memory containing hardwired instructions that the computer uses when it boots up, before the system software loads. In PCs, the instructions are read from a small program in the ROM, called the BIOS (Basic Input/Output System).

Sorting Algorithm: a sorting algorithm is an algorithm that puts elements of a list in a certain order.

Terabyte: is equal to 1,000 gigabytes; a terabyte is 10^{12} or 1,000,000,000,000 bytes.

Turing Test: a test for intelligence in a computer, requiring that a human being should be unable to distinguish the machine from another human being by using the replies to questions put to both.

Strand 3 - Web Development

Back End Web Developer: Person who uses scripting languages such as PHP and other databases to design how a website works.

Copyright: the exclusive legal right to print, publish, perform, film, or record literary, artistic, or musical material, and to authorize others to do the same.

Creative Commons: a set of various licenses that allow people to share their copyrighted work to be copied, edited, built upon, etc., while retaining the copyright to the original work (often used attributively).

CSS (Cascading Style Sheets): a language used to describe how HTML elements should be styled.

CSS Class: An identifier that allows multiple elements in an HTML document to be styled in the same way

CSS Selector: the part of a CSS rule-set that defines which HTML elements should be applied to the style.

Digital Footprint: a trail of data you create while using the Internet. It includes the websites you visit, emails you send, and information you submit to online services.

Front End Web Developer: Person who uses HTML, CSS and image editing software to design the front end or public side of the website.

Full Stack Development: Developer who is comfortable working with both back-end and front-end technologies.

Headings: are HTML tags that are used to indicate sections of a page. The names of each heading indicates its level in a hierarchy.

HTML: Hypertext Markup Language, a language used to create web pages

HTML Element: A piece of a website, marked by a start tag and often closed with an end tag

HTML Tag: The special set of characters that indicates the start and end of an HTML element and that element's type

HTTP (Hypertext Transfer Protocol): is the set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hyperlink: is a word, phrase, or image that you can click on to jump to a new document or a new section within the current document. Text hyperlinks are often blue and underlined, but don't have to be. When you move the cursor over a hyperlink, whether it is text or an image, the arrow should change to a small hand pointing at the link. When you click it, a new page or place in the current page will open.

Netiquette: the rules of etiquette that apply when communicating over computer networks, especially the Internet.

Web Browser: is an application used to access and view websites. Common web browsers include Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, and Apple Safari.

Website Content: the text and images on a website

Website Structure: how the content of a website is organized

UX (User Experience) Designer: Designers are primarily concerned with how the product feels.

UI (User Interface) Designer: Designers are particular about how the product is laid out.

Strand 4 - Programming & Algorithms

Boolean Logic: Boolean logic deals with the basic operations of truth values: AND, OR, NOT and combinations thereof.

Conditional Statement: a set of rules performed if a certain condition is met. It is sometimes referred to as an If-Then statement, because IF a condition is met, THEN an action is performed.

Debugging: A systematic approach to problem solving that is often used to find and resolve a problem, error, or fault within software or a computer system.

Decompose: Break a problem down into smaller pieces.

Flowchart: as a graphic representation of an algorithm, a step-by-step approach to solving a task.

Functions: are "self contained" modules of code that accomplish a specific task. Functions usually "take in" data, process it, and "return" a result. Once a function is written, it can be used over and over and over again. Functions can be "called" from the inside of other functions.

Iterate: is the repetition of a function or process in a computer program. Iterations of functions are common in computer programming, since they allow multiple blocks of data to be processed in sequence. This is typically done using a "while loop" or "for loop". These loops will repeat a process until a certain number or case is reached.

Loop: A programming structure that repeats a sequence of instructions as long as a specific condition is true.

Operators: is a symbol that usually represents an action or process. These symbols were adapted from mathematics and logic. An operator is capable of manipulating a certain value or operand. Operators are the backbone of any program and they are used for everything from very simple functions like counting to complex algorithms like security encryption.

Physical Computing: building interactive physical systems by the use of software and hardware that can sense and respond to the analog world.

Pseudocode: A detailed yet readable description of what a computer program or algorithm must do, expressed in a formally-styled natural language rather than in a programming language.

Prototype: An early approximation of a final product or information system, often built for demonstration purposes.

Sensors: a device that detects and responds to some type of input from the physical environment. Sensors send data to a microprocessor (computer).

Storyboard: a graphic organizer that provides the viewer with a high-level view of a project. In software development, a storyboard can help developers quickly get a sense of what work still needs to be completed.

User: A person for whom a hardware or software product is designed (as distinguished from the developers).

Variable: A placeholder for a piece of information that can change (Code.org K–5) *Note: This definition differs from that used in math.*