Alabama CTE

Programming Foundations (2022)

Foundational Standards

1. Incorporate safety procedures in handling, operating, and maintaining tools and machinery; handling materials; utilizing personal protective equipment; maintaining a safe work area; and handling hazardous materials and forces. F.1
2. Demonstrate effective workplace and employability skills, including communication, awareness of diversity, positive work ethic, problem-solving, time management, and teamwork. F.2
3. Explore the range of careers available in the field and investigate their educational requirements, and demonstrate job-seeking skills including resume-writing and interviewing. F.3
4. Advocate and practice safe, legal, responsible, and ethical use of information and technology tools specific to the industry pathway. F.4
5. Participate in a Career and Technical Student Organization (CTSO) to increase knowledge and skills and to enhance leadership and teamwork. F.5
6. Use technology to collaborate with peers and/or experts to create digital artifacts that can be published online for a target audience. F.6
7. Formulate new ideas, solve problems, or create products through the design and engineering process by utilizing testing, prototypes, and user feedback. F.7

Digital Literacy

1. Describe ethical and legal practices for safeguarding the confidentiality of business-related information. 1
2. Describe possible threats to a laptop, tablet, computer, and/or network and methods for avoiding attacks related to programming. 2
3. Explain the consequences of social engineering, illegal, and unethical uses of technology. Examples: piracy, illegal downloading, licensing infringement; inappropriate use of software, hardware, or mobile devices in the work environment 3
4. Describe computing innovations which have the potential to advance ırogramming or other aspects of computer science. Examples: artificial ıtelligence, quantum computing, low- or no-code programming 4

Skip to main content

Computer Systems	**5 Describe the flow of data and instructions through computer systems.** 5
	6 Explain how data is represented, manipulated, and stored in a computer. 6
	7 Describe the components of the programming development environment (the hardware and software used by programmers). Examples: text editor, compiler, debugging, profiler, IDE, modeling 7
Software Design and Programming	**8 Compare and contrast current programming languages utilized by business and industry and determine features, functions, and benefits of each.** 8
	9 Identify and explain various kinds of cryptographic algorithms. Examples: hashing, symmetric, asymmetric 9
	10 Explain why any input-processing algorithm must correctly handle all problem variants. 10
	11 Write an algorithm to solve mathematical problems using formulas, equations, and functions. 11
	12 Represent the logical flow of a program graphically. Examples: flowcharts, data traces, input/output charts 12
	13 Utilize and explain techniques for code commenting and documentation. Example: inserting meta text in source code 13
	14 Design a program that uses mathematical operations, data, functions, looping and iteration, sequencing, abstraction, lists, and selection. Examples: if-else statements, comparison 14
	a Design a program using visual modeling software to illustrate abstraction of languages from the solutions. Examples: Appian, Claris FileMaker, DWkit, Google AppSheet, Looker 7, Mendix, Microsoft PowerApps, OutSystems, Robocoder Rintagi, Salesforce Lightning, Sisense, Skyve Foundry, Temenos (formerly Kony), SIB Visions VisionX, Wix Editor X, Yellowfin 9, Zoho Creator 14.A
	15 Design a program that passes arguments and parameters (variables). 15
	16 Evaluate algorithms based on given designs to discuss their efficiency, correctness, and clarity. Examples: analyzing and comparing execution times, testing with multiple inputs or data sets, debugging 16
	17 Construct programs that utilize logical algorithms from specifications and requirement statements. 17
	18 Create a model software program which involves coding, testing, and documenting according to industry coding standards and guidelines. 18
Co Sy	**Explain how strings of 0s and 1s are used in programming.** 19

Skip to main content

20 **Summarize how numerical values are represented using different bases, including decimal and binary.** 20

21 **Demonstrate how numbers with decimals can have fixed-point or floating-point representations in binary.** 21

22 **Compare and contrast quantum and classical computing notation systems. Example: qubits** 22