Alabama CTE

Programming Design and Development (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.** PDD.FD.1
2. **Demonstrate effective workplace and employability skills, including communication, awareness of diversity, positive work ethic, problem-solving, time management, and teamwork.** PDD.FD.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.** PDD.FD.3
4. **Advocate and practice safe, legal, responsible, and ethical use of information and technology tools specific to the industry pathway.** PDD.FD.4
5. **Participate in a Career and Technical Student Organization (CTSO) to increase knowledge and skills and to enhance leadership and teamwork.** PDD.FD.5
6. **Use technology to collaborate with peers and/or experts to create digital artifacts that can be published online for a target audience.** PDD.FD.6
7. **Formulate new ideas, solve problems, or create products through the design and engineering process by utilizing testing, prototypes, and user feedback.** PDD.FD.7

Customer Service

1. **Research and collect data to create a solution that aligns with the client's needs and goals.** PDD.CS.1

   Example: Design and utilize a questionnaire to assess customers' needs.
2. **Design an information technology-based project plan utilizing researched strategies to solve a given problem, including aspects of planning and cybersecurity, design implementation, and project management.** PDD.CS.2
3. **Perform quality assurance protocols to enable the delivery of working software products according to specifications.** PDD.CS.3

   Examples: quality audits, quality testing, inspection, checkpoint reviews

Skip to main content

	4 **Deliver and evaluate basic technical documents, presentations, and group interactions, using a variety of authoring tools and desktop and cloud-based software.** PDD.CS.4
Software Design	**5** **Demonstrate the effective use of tools for software development.** PDD.SD.5 Examples: IDEs, professional and amateur repositories
	6 **Classify program structure, blocks, and storage types according to operational efficiency.** PDD.SD.6
	7 **Construct console and file input and output, functions, arrays, and strings.** PDD.SD.7
	8 **Develop a software program that demonstrates input/output, processing, and storage in order to outline the flow of data for each phase.** PDD.SD.8
	9 **Create an advanced algorithm using plain language and incorporating pseudocode to solve a real-world programming problem.** PDD.SD.9
	10 **Design a program that uses data, functions, looping and iteration, sequencing, abstraction, list, and selection.** PDD.SD.10 Examples: if-else statements, comparison
	11 **Integrate mathematical concepts into a program by writing the code, performing unit testing, and debugging the program.** PDD.SD.11 Examples: logical reasoning, order of operations, functional reasoning, proportional reasoning
	12 **Utilize Boolean operators, mathematical operators, and relational operators in creating program code.** PDD.SD.12
	13 **Create an algorithm that includes an input and an output to solve a real-world problem.** PDD.SD.13
	14 **Debug processes within a program by identifying and locating the problem, removing the faulty source code, and repairing the code.** PDD.SD.14
	15 **Utilize efficient searching algorithms to solve a given problem.** PDD.SD.15 Examples: linear, binary, jump, interpolation, exponential, ternary
	16 **Utilize mathematical formulas to assess the efficiency of sorting and searching algorithms and choose the more efficient one to use in a given situation.** PDD.SD.16 Example: Use BigO notation to determine whether an algorithm is both correct and efficient.

Skip to main content

17 **Create complex applications using input, calculations, output, control structures, and data structures.** PDD.SD.17

18 **Construct recursive algorithms to solve a problem.** PDD.SD.18

19 **Develop class constructors using method overloading concepts.** PDD.SD.19

Examples: changing the number of arguments to determine which instance of the class will be created

20 **Construct multidimensional arrays and use the input and output data to solve a problem.** PDD.SD.20