

Florida Department of Education
Curriculum Framework

Program Title: Web Application Development & Programming
Program Type: Career Preparatory
Career Cluster: Information Technology

Secondary – Career Preparatory

Program Number	9007500
CIP Number	0511020112
Grade Level	9-12
Program Length	7 credits
Teacher Certification	Refer to the Program Structure section.
CTSO	FBLA, BPA, FL-TSA
SOC Codes (all applicable)	15-1151 – Computer User Support Specialists 15-1131 – Computer Programmers
CTE Program Resources	http://www.fldoe.org/academics/career-adult-edu/career-tech-edu/program-resources.stml

Purpose

This program offers a sequence of courses that provides coherent and rigorous content aligned with challenging academic standards and relevant technical knowledge and skills needed to prepare for further education and careers in the Information Technology career cluster; provides technical skill proficiency, and includes competency-based applied learning that contributes to the academic knowledge, higher-order reasoning and problem-solving skills, work attitudes, general employability skills, technical skills, and occupation-specific skills, and knowledge of all aspects of the Information Technology career cluster.

The content includes but is not limited to the fundamentals of programming and software development; procedural and object-oriented programming; creating web-based applications, including testing, monitoring, debugging, documenting, and maintaining applications.

Additional Information relevant to this Career and Technical Education (CTE) program is provided at the end of this document.

Program Structure

This program is a planned sequence of instruction consisting of seven (7) credits.

To teach the courses listed below, instructors must hold at least one of the teacher certifications indicated for that course.

The following table illustrates the secondary program structure:

Course Number	Course Title	Teacher Certification	Length	SOC Code	Level	Graduation Requirement
8207310	Digital Information Technology	DIT Teacher Certifications	1 credit	15-1151	2	CT
9007210	Foundations of Programming	BUS ED 1 @2 COMP SCI 6 COMP PROG 7 G	1 credit	15-1131	3	CT
9007220	Procedural Programming		1 credit	15-1131	3	CT
9007230	Object-Oriented Programming Fundamentals		1 credit	15-1131	3	CT
9007510	Web Programming		1 credit	15-1131	3	CT
9007520	JavaScript Programming		1 credit	15-1131	3	CT
9007530	PHP Programming		1 credit	15-1131	3	CT

(Graduation Requirement Codes: CT=Career & Technical Education, EQ= Equally Rigorous Science, EC= Economics, MA=Mathematics, PL=Personal Financial Literacy)

Common Career Technical Core – Career Ready Practices

Career Ready Practices describe the career-ready skills that educators should seek to develop in their students. These practices are not exclusive to a Career Pathway, program of study, discipline or level of education. Career Ready Practices should be taught and reinforced in all career exploration and preparation programs with increasingly higher levels of complexity and expectation as a student advances through a program of study.

1. Act as a responsible and contributing citizen and employee.
2. Apply appropriate academic and technical skills.
3. Attend to personal health and financial well-being.
4. Communicate clearly, effectively and with reason.
5. Consider the environmental, social and economic impacts of decisions.
6. Demonstrate creativity and innovation.
7. Employ valid and reliable research strategies.
8. Utilize critical thinking to make sense of problems and persevere in solving them.
9. Model integrity, ethical leadership and effective management.
10. Plan education and career path aligned to personal goals.
11. Use technology to enhance productivity.
12. Work productively in teams while using cultural/global competence.

Standards

Digital Information Technology (8207310) is the first course in this and other programs within the Information Technology Career Cluster. Standards 01.0 – 15.0 are associated with this course.

After successfully completing this program, the student will be able to perform the following:

- 01.0 Demonstrate knowledge, skill and application of information technology to accomplish job objectives and enhance workplace performance.
- 02.0 Develop an awareness of microcomputers.
- 03.0 Demonstrate an understanding of networks.
- 04.0 Use word processing applications to enhance the effectiveness of various types of documents and communication.
- 05.0 Use presentation applications to enhance communication skills.
- 06.0 Use spreadsheet applications to enhance communication skills.
- 07.0 Use database applications to store and organize data.
- 08.0 Use electronic mail to enhance communication skills.
- 09.0 Investigate individual assessment and job/career exploration and individual career planning that reflect the transition from school to work, lifelong learning and personal and professional goals.
- 10.0 Incorporate appropriate leadership and supervision techniques, customer service strategies, and standards of personal ethics to accomplish job objectives and enhance workplace performance.
- 11.0 Demonstrate competence using computer networks, internet and online databases to facilitate collaborative or individual learning and communication.
- 12.0 Develop awareness of computer languages, web-based and software applications, and emerging technologies.
- 13.0 Demonstrate an understanding of basic html by creating a simple web page.
- 14.0 Demonstrate comprehension and communication skills.
- 15.0 Use social media to enhance online communication and develop an awareness of a digital footprint.
- 16.0 Use oral and written communication skills in creating, expressing and interpreting information and ideas.
- 17.0 Explore the characteristics, tasks, work attributes, options and tools associated with a career in software development.
- 18.0 Demonstrate an understanding of the characteristics, use, and selection of numerical, non-numerical and logical data types.
- 19.0 Distinguish between iterative and non-iterative program control structures.
- 20.0 Describe the processes, methods, and conventions for software development and maintenance.
- 21.0 Explain the types, uses, and limitations of testing for ensuring quality control.
- 22.0 Create a program design document using common design tool.
- 23.0 Solve problems using critical thinking skills, creativity and innovation.
- 24.0 Describe the importance of security and privacy information sharing, ownership, licensure and copyright.
- 25.0 Create programs that solve a problem using non-iterative and iterative algorithms.
- 26.0 Design a computer program to meet specific physical, operational and interaction criteria.
- 27.0 Create and document a computer program that uses a variety of internal and control structures for manipulating varied data types.
- 28.0 Create and document an interactive computer program that employs functions, subroutines, or methods to receive, validate and process user input.
- 29.0 Effectively communicate and collaborate.

- 30.0 Demonstrate responsible use of technology and information.
- 31.0 Differentiate among procedural, object-oriented, compiled, interpreted and translated programming languages.
- 32.0 Explain key concepts that distinguish object-oriented programming from procedural programming.
- 33.0 Create a project plan for an object-oriented programming project that defines requirements, structural design, time estimates and testing elements.
- 34.0 Design, document, and create object-oriented computer programs.
- 35.0 Design a unit test plan for an object-oriented computer program, test and debug the program and report the results.
- 36.0 Understand human-AI interaction.
- 37.0 Demonstrate proficiency using HTML and XHTML to create web content.
- 38.0 Demonstrate proficiency using cascading style sheets (CSS) to format webpages.
- 39.0 Demonstrate proficiency using basic client-side scripting to control the content and the behavior of HTML and XHTML documents.
- 40.0 Demonstrate an understanding of JavaScript programming fundamentals.
- 41.0 Demonstrate proficiency in assigning and handling variables in JavaScript programs and functions.
- 42.0 Use event handlers in JavaScript programs and functions.
- 43.0 Recognize and assign data types appropriate to their use.
- 44.0 Demonstrate proficiency in using appropriate operators to achieve a planned output.
- 45.0 Write executable statements.
- 46.0 Demonstrate an understanding of variable scope.
- 47.0 Use good programming practices.
- 48.0 Demonstrate use of the Document Object Module (DOM).
- 49.0 Use conditional control statements in JavaScript.
- 50.0 Use iterative control statements in JavaScript.
- 51.0 Use nested loop iterative control statements in JavaScript.
- 52.0 Use JavaScript to produce input and output for programs.
- 53.0 Demonstrate proficiency in using Form Objects in JavaScript programs and functions.
- 54.0 Demonstrate proficiency in using methods in JavaScript programs and functions.
- 55.0 Demonstrate proficiency in using parameters in JavaScript programs and functions.
- 56.0 Utilize debugging techniques in programs.
- 57.0 Recognize security risks in programs.
- 58.0 Use plug-ins and libraries.
- 59.0 Demonstrate proficiency in programming for mobile delivery technology (e.g., iPhone/Android).
- 60.0 Demonstrate an understanding of Personal Home Page (PHP) programming language.
- 61.0 Demonstrate proficiency in PHP configuration.
- 62.0 Demonstrate an understanding of PHP language basics.
- 63.0 Demonstrate proficiency in the use of server processes.
- 64.0 Demonstrate an understanding of object-oriented programming in PHP.
- 65.0 Demonstrate proficiency in writing PHP code to handle file input/output (I/O) operations.
- 66.0 Demonstrate proficiency in creating, populating, and using arrays in PHP.
- 67.0 Demonstrate proficiency handling strings in PHP.
- 68.0 Demonstrate proficiency in using PHP to access databases via Open Database Connectivity (ODBC).

- 69.0 Demonstrate proficiency in applying best practices for ensuring creation of a secure program.
- 70.0 Demonstrate an understanding of key technologies, protocols and architectures associated with web development and programming.

Florida Department of Education
Student Performance Standards

Course Title: Digital Information Technology
Course Number: 8207310
Course Credit: 1

Course Description:

This core course is designed to provide a basic overview of current business and information systems and trends, and to introduce students to fundamental skills required for today's business and academic environments. Emphasis is placed on developing fundamental computer skills. The intention of this course is to prepare students to be successful both personally and professionally in an information-based society. Digital Information Technology includes the exploration and use of: databases, the internet, spreadsheets, presentation applications, management of personal information and email, word processing and document manipulation, HTML, web page design and the integration of these programs using software that meets industry standards.

Digital Information Technology (8207310) is part of several programs across the various CTE career clusters. To ensure consistency, the standards and benchmarks for this course (01.0 – 15.0) have been placed in a separate document. To access this document, visit: [Digital Information Technology \(8207310\)](#).

**Florida Department of Education
Student Performance Standards**

Course Title: Foundations of Programming
Course Number: 9007210
Course Credit: 1

Course Description:

This course introduces concepts, techniques, and processes associated with computer programming and software development.

CTE Standards and Benchmarks	
16.0	Use oral and written communication skills in creating, expressing and interpreting information and ideas. The student will be able to:
16.01	Select and employ appropriate communication concepts and strategies to enhance oral and written communication in the workplace.
16.02	Locate, organize and reference written information from various sources.
16.03	Construct writings and/or communications using developmentally appropriate terminology.
16.04	Analyze the positive and negative impacts of technology on popular culture and personal life.
16.05	Discuss how technology has changed the way people build and manage organizations and how technology impacts personal life.
16.06	Evaluate ways in which adaptive technologies may assist users with special needs.
16.07	Explain how societal and economic factors are affected by access to critical information.
16.08	Discuss the challenges (e.g., political, social, and economic) in providing equal access and distribution of technology in a global society.
17.0	Explore the characteristics, tasks, work attributes, options and tools associated with a career in software development. The student will be able to:
17.01	Explore a variety of careers to which computing is central.
17.02	Discuss the impact of computing on business and commerce (e.g., automated inventory processing, financial transactions, e-commerce, virtualization and cloud computing).
17.03	Evaluate the impacts of irresponsible use of information (e.g., plagiarism and falsification of data) on collaborative projects.
17.04	Identify tasks performed by programmers.
17.05	Describe how businesses use computer programming to solve business problems.
17.06	Investigate job opportunities in the programming field.
17.07	Explain different specializations and the related training in the computer programming field.
17.08	Explain the need for continuing education and training of computer programmers.

CTE Standards and Benchmarks	
17.09	Understand and identify ways to use technology to support lifelong learning.
17.10	Explain software as a service (SaaS) and how it impacts business.
17.11	Describe ethical responsibilities of computer programmers.
17.12	Identify credentials and certifications that may improve employability for a computer programmer.
17.13	Identify devices, tools, and other environments for which programmers may develop software.
18.0	Demonstrate an understanding of the characteristics, use, and selection of numerical, non-numerical and logical data types. The student will be able to:
18.01	Identify the characteristics (e.g., size, limits) and uses of different numerical and non-numerical data types.
18.02	Explain the types and uses of variables in programs.
18.03	Determine the best data type to use for given programming problems.
18.04	Compare and contrast simple data structures and their uses.
18.05	Identify the types of operations that can be performed on different data types (e.g., math operations on numerical data types, concatenation and other string operations).
18.06	Evaluate arithmetic and logical expressions using appropriate operator precedence.
18.07	Explain how computers store different data types in memory.
18.08	Demonstrate the difference between "data" and "information".
18.09	Use different number systems to represent data.
18.10	Explain how national and international standards (i.e., ASCII, UNICODE) are used to represent non-numerical data.
18.11	Use Boolean logic to perform logical operations using Boolean algebra and truth tables.
19.0	Distinguish between iterative and non-iterative program control structures. The student will be able to:
19.01	Identify the uses of non-iterative and iterative programming structures using pseudocode and flowcharts.
19.02	Create iterative programming structures and their uses.
19.03	Explain how sequence, selection, and iteration are building blocks of algorithms.
20.0	Describe the processes, methods, and conventions for software development and maintenance. The student will be able to:
20.01	Describe a software development process that is used to solve problems at different software development stages.
20.02	Define alternative methods of program development (e.g., rapid prototyping, waterfall, spiral model, peer coding).
20.03	List and explain the steps in the program development cycle.
20.04	Describe different types of documentation used in the program development cycle (e.g., requirements document, program design

CTE Standards and Benchmarks	
	documents, test plans).
	20.05 Describe different methods used to facilitate version control.
21.0	Explain the types, uses, and limitations of testing for ensuring quality control. The student will be able to:
21.01	Explain the uses and limits of testing in ensuring program quality.
21.02	Explain testing performed at different stages of the program development cycle (e.g., unit testing, system testing, user acceptance testing).
21.03	Describe and identify types of programming errors.
22.0	Create a program design document using common design tool. The student will be able to:
22.01	Describe different design methodologies and their uses (e.g., object-oriented design, structured design, rapid application development).
22.02	Describe and use tools for developing a program design (e.g., flowcharts, design documents, pseudocode).
22.03	Explain the role of existing libraries and packages in facilitating programmer productivity.
22.04	Participate and contribute to a design review of a program design developed using a common program design tool (e.g., UML, flowcharts, design documents, pseudocode).
22.05	Develop a software artifact (independently and collaboratively) in phases (or stages) according to a common software development methodology (e.g., Waterfall or Spiral model).
22.06	Define input and output for a program module using standard design methodology.
23.0	Solve problems using critical thinking skills, creativity and innovation. The student will be able to:
23.01	Employ critical thinking skills independently and in teams to solve problems and make decisions.
23.02	Employ critical thinking and collaborative skills to resolve conflicts.
23.03	Identify and document workplace performance goals and monitor progress toward those goals.
23.04	Conduct technical research to gather information necessary for decision-making.
23.05	Discuss digital tools or resources to use for a real-world task based on their efficiency and effectiveness, individually and collaboratively.
24.0	Describe the importance of security and privacy information sharing, ownership, licensure and copyright. The student will be able to:
24.01	Describe security and privacy issues that relate to computer networks including the permanency of data on the Internet, online identity, and privacy.
24.02	Discuss the impact of government regulation on privacy and security.
24.03	Describe how different types of software licenses (e.g., open source and proprietary licenses) can be used to share and protect intellectual property.
24.04	Explain how access to information may not include the right to distribute the information.

CTE Standards and Benchmarks	
24.05	Describe differences between open source, freeware and proprietary software licenses, and how they apply to different types of software.
24.06	Discuss security and privacy issues that relate to computer networks.
24.07	Identify computer-related laws and analyze their impact on digital privacy, security, intellectual property, network access, contracts and harassment.
25.0	Create programs that solve a problem using non-iterative and iterative algorithms. The student will be able to:
25.01	Apply the developmental cycle methodologies to create a program.
25.02	Develop a program using string and/or numeric data types.
25.03	Develop a program using sequential algorithms.
25.04	Develop a program using selection structures.
25.05	Develop a program using looping structures.

**Florida Department of Education
Student Performance Standards**

Course Title: **Procedural Programming**
Course Number: **9007220**
Course Credit: **1**

Course Description:

This course continues the study of computer programming concepts with a focus on the creation of software applications employing procedural programming techniques.

CTE Standards and Benchmarks	
26.0	Design a computer program to meet specific physical, operational, and interaction criteria. The student will be able to:
26.01	Choose appropriate data types depending on the needs of the program.
26.02	Define appropriate user prompts for clarity and usability (e.g., user guidance for data ranges, data types).
26.03	Design and develop program for efficiency (e.g., less memory usage, less inputs/outputs, faster processing).
26.04	Compare techniques for analyzing massive data collections.
26.05	Identify the software environment required for a program to run (e.g., operating system required, mobile, web-based, desktop, delivery method).
26.06	Create mobile computing applications and/or dynamic webpages through the use of a variety of design and development tools, programming languages and mobile devices/emulators.
26.07	Explain the role of an application programming interface (API) in the development of applications and the distinction between a programming language’s syntax and the API.
26.08	Identify the tools required to develop a program (e.g., editors, compilers, linkers, integrated development environments, APIs, libraries).
27.0	Create and document a computer program that uses a variety of internal and control structures for manipulating varied data types. The student will be able to:
27.01	Use appropriate naming conventions to define program variables and methods.
27.02	Use a program editor to write the source code for a program.
27.03	Write programs that use selection structures.
27.04	Write programs that use repetition structures.
27.05	Write programs that use nested structures.
27.06	Use internal documentation (e.g., single-line and multi-line comments, program headers, module descriptions, meaningful variable and function/module names) to document a program according to accepted standards.

CTE Standards and Benchmarks	
27.07	Compile, run, test and debug programs.
27.08	Write programs that use standard arithmetic operators with different numerical data types.
27.09	Write programs that use standard logic operators.
27.10	Write programs that use a variety of common data types.
27.11	Write programs that perform data conversion between numeric and string data types.
27.12	Write programs that define, use, search and sort arrays.
27.13	Write programs that use user-defined data types.
27.14	Demonstrate understanding and use of appropriate variable scope.
27.15	Use global and local scope appropriately in program implementation.
27.16	Distinguish between binary and sequential searches.
28.0	Create and document an interactive computer program that employs functions, subroutines, or methods to receive, validate and process user input. The student will be able to:
28.01	Determine the results of code segments.
28.02	Write programs that perform user input and output.
28.03	Write programs that validate user input (e.g., range checking, data formats, valid/invalid characters).
28.04	Write program modules such as functions, subroutines or methods.
28.05	Write program modules that accept arguments.
28.06	Write program modules that return values.
28.07	Write program modules that validate arguments and return error codes.
28.08	Design and implement a simple simulation algorithm to analyze, represent and understand natural phenomena.
28.09	Use APIs and libraries to facilitate programming solutions.
28.10	Participate in a peer code review to verify program functionality, programming styles, program usability and adherence to common programming standards.
28.11	Explain how abstraction manages complexity.
29.0	Effectively communicate and collaborate. The student will be able to:
29.01	Evaluate modes of communication and collaboration.
29.02	Select appropriate tools within a project environment to communicate with project team members.
29.03	Utilize project collaboration tools (such as version control systems and integrated development environments) while working on a

CTE Standards and Benchmarks	
	collaborative software project.
29.04	Generate, evaluate, and prioritize questions that can be researched through digital resources and online tool.
29.05	Perform advanced searches to locate information and/or design a data-collection approach to gather original data.
29.06	Communicate and publish key ideas and details to a variety of audiences using digital tools and media-rich resources.
30.0	Demonstrate responsible use of technology and information. The student will be able to:
30.01	Implement an encryption, digital signature or authentication method.
30.02	Describe computer security vulnerabilities and methods of attack and evaluate their social and economic impact on computer systems and people (e.g., phishing, keylogging, virus, malware, intercepting data over public networks).
30.03	Identify and explain the existence of biases in computer programming.
30.04	Explain how computing can play a role in social and political issues.
31.0	Differentiate among procedural, object-oriented, compiled, interpreted and translated programming languages. The student will be able to:
31.01	Differentiate between multiple levels of operating system, translation and interpretation that support program execution.
31.02	Explain the program execution process (by an interpreter and in CPU hardware).
31.03	Describe object-oriented concepts.
31.04	Explain the characteristics of procedural and object-oriented programming languages.
31.05	Compare and contrast programming languages that are compiled, interpreted and translated.

**Florida Department of Education
Student Performance Standards**

Course Title: Object-Oriented Programming Fundamentals
Course Number: 9007230
Course Credit: 1

Course Description:

This course continues the study of computer programming concepts with a focus on the creation of software applications employing object-oriented programming techniques.

CTE Standards and Benchmarks	
32.0	Explain key concepts that distinguish object-oriented programming from procedural programming. The student will be able to:
32.01	Demonstrate the understanding and use of classes, objects, attributes and behaviors.
32.02	Demonstrate the understanding and use of inheritance.
32.03	Demonstrate the understanding and use of data encapsulation.
32.04	Demonstrate the understanding and use of polymorphism.
32.05	Use predefined functions and parameters, classes, and methods to divide a complex problem into simpler parts by using the principle of abstraction to manage complexity (e.g., by using searching and sorting as abstractions).
33.0	Create a project plan for an object-oriented programming project that defines requirements, structural design, time estimates, and testing elements. The student will be able to:
33.01	Write a project plan for completion of a project that includes gathering program requirements, developing the program, and testing it.
33.02	Write a program requirements document that identifies business purpose, functional requirements, system requirements and other common components of a requirements document.
33.03	Design an object-oriented program using standard design methodology.
33.04	Work with other team members to develop a project plan for a program.
33.05	Work with other team members to write a design document for a program with multiple functions and shared data.
33.06	Participate in design meetings that review program design documents for conformance to program requirements.
33.07	Estimate the time to develop a program or module.
33.08	Evaluate algorithms by their efficiency, correctness, and clarity (e.g., by analyzing and comparing execution times, testing with multiple inputs or data sets, and by debugging).
34.0	Design, document, and create object-oriented computer programs. The student will be able to:
34.01	Compare and contrast recursive functions to other iterative methods.

CTE Standards and Benchmarks	
34.02	Understand the implementation of character strings in the programming language.
34.03	Write programs that perform string processing (e.g., manipulating, comparing strings, concatenation).
34.04	Write programs that implements user-defined data types.
34.05	Decompose a problem by defining new functions and classes.
34.06	Write object-oriented programs that implement inheritance.
34.07	Write object-oriented programs that implement polymorphism.
34.08	Develop class constructors.
34.09	Write programs that define and use program constants.
34.10	Write programs that perform error handling.
34.11	Participate in program code review meetings to evaluate program code for validity, quality, performance, data integrity, and conformance to program design documents.
34.12	Describe the concept of parallel processing as a strategy to solve large problems.
34.13	Demonstrate concurrency by separating processes into threads of execution and dividing data into parallel streams.
34.14	Update a program module to implement enhancements or corrections and demonstrate appropriate documentation (internal and external) related to version control.
34.15	Write programs that are event-driven.
34.16	Write programs that perform file input and output (i.e., sequential and random-access file input/output).
34.17	Explain the value of heuristic algorithms to approximate solutions for unmanageable problems (e.g., a heuristic solution to Towers of Hanoi).
35.0	Design a unit test plan for an object-oriented computer program, test and debug the program, and report the results. The student will be able to:
35.01	Develop a test plan for an object-oriented program.
35.02	Write test plans for programs that perform file input and output.
35.03	Perform test and debug activities on object-oriented programs, including those written by someone else.
35.04	Perform test and debug activities on programs that perform file input and output and verify the correctness of output files.
35.05	Document the findings of testing in a test report.
36.0	Understand human-AI interaction. The student will be able to:
36.01	Describe the unique features of computers embedded in mobile devices and vehicles.
36.02	Describe the common physical and cognitive challenges faced by users when learning to use software and hardware.

CTE Standards and Benchmarks

36.03	Describe the process of designing software to support specialized forms of human-computer interaction.
36.04	Explain the notion of intelligent behavior through computer modeling and robotics.
36.05	Describe common measurements of machine intelligence (e.g., Turing test).
36.06	Describe a few of the major branches of artificial intelligence (e.g., expert systems, natural language processing, machine perception, machine learning).
36.07	Describe major applications of artificial intelligence and robotics, including, but not limited to, the medical, space and automotive fields.

**Florida Department of Education
Student Performance Standards**

Course Title: Web Programming
Course Number: 9007510
Course Credit: 1

Course Description:

This course continues the study of computer programming concepts specific to the Internet and Internet-based software applications.

CTE Standards and Benchmarks	
37.0	Demonstrate proficiency using HTML and XHTML to create web content. The student will be able to:
37.01	Use storyboarding techniques for designing a website (e.g., linear, hierarchical).
37.02	Identify elements of a webpage.
37.03	Create webpages using HTML and XHTML tags that create basic elements (e.g., links, lists, formatted text, tables).
37.04	Create webpages that utilize tables to achieve complex layout.
37.05	Add graphic content to webpages.
37.06	Create webpages that utilize client-side image maps.
37.07	Develop, integrate, and apply the use of forms in website design.
37.08	Optimize Web content for desirable search engine placement.
37.09	Demonstrate an understanding of browser compatibility issues by designing pages that comply with the current Web Content Accessibility Guidelines issued by the World Wide Web Consortium (W3C).
37.10	Demonstrate an understanding of Web accessibility issues by developing pages that meet Bobby accessibility checker criteria.
37.11	Explain basic XML syntax and how XHTML conforms to the XML standard.
37.12	Use a WYSIWYG editor to develop and manage a website.
37.13	Use markup validation tools to test HTML and XHTML documents for well-formed elements and make all corrections necessary to ensure compliance with W3C standards.
37.14	Analyze and modify HTML and XHTML source code developed by others.
38.0	Demonstrate proficiency using cascading style sheets (CSS) to format webpages. The student will be able to:
38.01	Explain the advantages and disadvantages of using Cascading Style Sheets (CSS) to format webpages.
38.02	Describe the difference between linked, embedded, imported and inline styles and explain how styles are inherited.

CTE Standards and Benchmarks	
38.03	Explain the difference between classes, id, and span elements.
38.04	Utilize CSS properties within webpages to control page layout, fonts, colors, backgrounds and other presentation effects.
38.05	Demonstrate understanding of the Box Model.
38.06	Demonstrate proficiency in creating 1 to 3 column layouts.
38.07	Create navigation system through CSS.
39.0	Demonstrate proficiency using basic client-side scripting to control the content and the behavior of HTML and XHTML documents. The student will be able to:
39.01	Describe the difference between server-side and client-side processing.
39.02	Describe the term “scripting language” and explain how scripting languages differ from compiled languages.
39.03	Create webpages that employ client-side scripting to control content and display.

Florida Department of Education
Student Performance Standards

Course Title: JavaScript Programming
Course Number: 9007520
Course Credit: 1

Course Description:

This course continues the study of computer programming concepts specific to client-side JavaScript.

CTE Standards and Benchmarks	
40.0	Demonstrate an understanding of JavaScript programming fundamentals. The student will be able to:
40.01	Describe server side versus client side applications including interpreters.
40.02	Describe the purpose and use of an interpreter in relation to JavaScript.
40.03	Describe differences in different types of JavaScript implementations (i.e., Jscript, ECMA).
40.04	Declare and initialize variables.
40.05	Assign new values to variables.
40.06	Create and use constant variables.
40.07	Describe the difference of programming languages versus scripting languages.
40.08	Describe object based nature and platform independence.
40.09	Describe and demonstrate inline scripting.
41.0	Demonstrate proficiency in assigning and handling variables in JavaScript programs and functions. The student will be able to:
41.01	Describe how variables are used in programs.
41.02	Identify which data type should be used for a given value.
41.03	Identify the syntax for using variables.
41.04	Declare and initialize variables.
41.05	Assign new values to variables.
41.06	Create and use constant variables.
41.07	Describe and demonstrate the use of properties.
41.08	Describe identifiers and identify valid and invalid identifiers.

CTE Standards and Benchmarks	
41.09	Describe and identify reserved words, delimiters, literals and comments.
42.0	Use event handlers in JavaScript programs and functions. The student will be able to:
42.01	Describe the event model and five events (form, image, map, link and window).
42.02	Demonstrate and use the window events load, focus, blur and unload.
42.03	Demonstrate and use the form events change, reset and submit.
42.04	Demonstrate and use the text events cut, paste, select and copy.
42.05	Demonstrate and use the mouse events mousemove, mousedown, click, mousewheel, mouseover, mouseout and mouseup.
42.06	Demonstrate and use the keyboard events keyup, keydown and keypress.
42.07	Demonstrate using the appropriate event handlers with their associated events.
43.0	Recognize and assign data types appropriate to their use. The student will be able to:
43.01	Describe the data type categories.
43.02	Give examples of var, primitives, null and undefined data types.
43.03	Demonstrate the use of var in relation to other datatypes.
44.0	Demonstrate proficiency in using appropriate operators to achieve a planned output. The student will be able to:
44.01	Construct statements using arithmetic operators.
44.02	Construct statements using relational operators.
44.03	Construct and use statements using logical operators.
44.04	Construct and use statements using string concatenation and strict comparison.
44.05	Construct and use statements using assignment operators.
44.06	Construct and execute statements using operator precedence.
45.0	Write executable statements. The student will be able to:
45.01	Construct variable assignment statements.
45.02	Construct statements using built-in functions.
45.03	Describe when implicit data type conversions take place.
45.04	List the drawbacks of implicit data type conversions.
45.05	Construct statements using functions to explicitly convert data types.

CTE Standards and Benchmarks	
46.0	Demonstrate an understanding of variable scope. The student will be able to:
46.01	Understand the scope and visibility of variables.
46.02	Write programs using local variables.
46.03	Describe the scope of a variable.
47.0	Use good programming practices. The student will be able to:
47.01	List examples of good programming practices.
47.02	Insert comments into code.
47.03	Demonstrate the use of <no script> tag.
47.04	Follow formatting guidelines when writing code.
47.05	Understand the different types of errors produced by programs.
48.0	Demonstrate use of the Document Object Module (DOM). The student will be able to:
48.01	Create and use user defined objects.
48.02	Create user defined objects with properties and methods.
48.03	Describe and Use the Array Object including its parameters, properties, and methods (chop, join, pop, push, splice, split).
48.04	Describe and Use the Date Object including its multiple constructors, properties, and methods (getDay, getMonth, getYear, getMinutes, getHous, getTime).
48.05	Describe and use the Window Object including \properties, and methods.
48.06	Describe and use the Image Object including its properties and methods.
48.07	Describe and use the History Object including its properties and methods.
48.08	Describe and use the RegEx Object for basic and complex regular expressions.
48.09	Describe and use the String Object including its properties and methods.
48.10	Describe and use the Math Object including its properties and methods.
49.0	Use conditional control statements in JavaScript. The student will be able to:
49.01	Construct and use an if statement.
49.02	Construct and use a switch statement.
49.03	Construct and use a while, do while and for loop.
49.04	Construct and use a conditional operator.

CTE Standards and Benchmarks	
50.0	Use iterative control statements in JavaScript. The student will be able to:
50.01	Describe the types of loop statements and their uses.
50.02	Construct and use the while and do while loop.
50.03	Construct and use the for loop.
50.04	Describe when a while loop is used.
50.05	Describe when a for loop is used.
51.0	Use nested loop iterative control statements in JavaScript. The student will be able to:
51.01	Construct and execute a program using nested loops.
51.02	Construct and execute a loop using break and continue.
51.03	Evaluate a nested loop construct and sentinel value.
52.0	Use JavaScript to produce input and output for programs. The student will be able to:
52.01	Describe and use the prompt() and confirm() to input data into programs.
52.02	Describe and demonstrate the use of the alert() to produce output to the console.
52.03	Describe and demonstrate how to input data using JavaScript Events.
52.04	Describe and demonstrate how to output using the document.write().
52.05	Explain the difference of prompt() and confirm() functions.
52.06	Create and use escape sequences.
53.0	Demonstrate proficiency in using Form Objects in JavaScript programs and functions. The student will be able to:
53.01	Use Form objects to validate input.
53.02	Access the value of the form object through its associated method.
53.03	Describe and use button, checkbox, textarea, select, radio, hidden, and text objects.
53.04	Access and modify values and attributes at runtime using getElementById, getElementsByName, getElementsByTagName, and innerHTML.
54.0	Demonstrate proficiency in using methods in JavaScript programs and functions. The student will be able to:
54.01	Differentiate between anonymous methods and methods.
54.02	Identify the benefits of using methods.
54.03	Describe and use inner method.

CTE Standards and Benchmarks	
54.04	Create a method.
54.05	Describe how a method is invoked.
55.0	Demonstrate proficiency in using parameters in JavaScript programs and functions. The student will be able to:
55.01	Describe how parameters are passed into functions.
55.02	Define a parameter.
55.03	Create a method using a parameter.
55.04	Invoke a method that has parameters.
55.05	Distinguish between formal and actual parameters.
56.0	Utilize debugging techniques in programs. The student will be able to:
56.01	Use the display property to enable/disable code blocks.
56.02	Use document.write() to log program execution.
56.03	Test program in different browsers and mobile devices for compatibility errors.
56.04	Use comments as a flow control while debugging.
57.0	Recognize security risks in programs. The student will be able to:
57.01	Describe the security risk of cookies and browsers.
57.02	Identify security responsibilities of browsers and operating system.
57.03	Describe security systems such as frame to frame URL changing.
57.04	Describe the use of signed scripts.
57.05	Create and use cookies in a secure manner.
58.0	Use plug-ins and libraries. The student will be able to:
58.01	Use external libraries in the program.
58.02	Describe and contrast the following industry libraries JQuery, Dojo, LightBox, and Moo Tools, PhoneGap.
58.03	Describe different types of libraries full, effects, tools, graphing, math, cryptography, and AJAX.
58.04	Identify how load and reference external and user made scripts.
58.05	Describe AJAX elements and procedures.
58.06	Describe XML.

CTE Standards and Benchmarks	
58.07	Demonstrate the use of XMLHttpRequest to retrieve data.
59.0	Demonstrate proficiency in programming for mobile delivery technology (e.g., iPhone/Android). The student will be able to:
59.01	Respond to multi-touch and gesture events.
59.02	Describe and demonstrate the use of webkit CSS.
59.03	Use the meta tag to enable native look and feel.
59.04	Create a splash screen.
59.05	Describe and demonstrate app caching.
59.06	Describe and demonstrate use of JQuery for mobile development.
59.07	Describe how to publish the app using XCode.

Florida Department of Education
Student Performance Standards

Course Title: PHP Programming
Course Number: 9007530
Course Credit: 1

Course Description:

This course continues the study of computer programming concepts specific to PHP programming.

CTE Standards and Benchmarks	
60.0	Demonstrate an understanding of Personal Home Page (PHP) programming language. The student will be able to:
60.01	Describe the evolution of PHP as a programming language.
60.02	Discuss the strengths and limitations of PHP.
61.0	Demonstrate proficiency in PHP configuration. The student will be able to:
61.01	Set up a PHP host (wamp, mamp, online).
61.02	Configure PHP for File Transfer Protocol (FTP) access.
61.03	Configure the config.php file.
62.0	Demonstrate an understanding of PHP language basics. The student will be able to:
62.01	Describe how variables are declared, referenced, and passed.
62.02	Describe the control structures inherent with PHP programming.
62.03	Describe the three types of arrays used in PHP.
62.04	Describe how functions in PHP are created, called, and controlled.
63.0	Demonstrate proficiency in the use of server processes. The student will be able to:
63.01	Describe a session and explain its importance and use in web programming.
63.02	Describe the server processes associated with forms handling.
63.03	Compare and contrast the use of GET and POST.
63.04	Describe cookies and explain their use, population, control, and risks.
63.05	Describe HTTP Headers and their role in web development.
63.06	Describe HTTP Authentication.

CTE Standards and Benchmarks	
64.0	Demonstrate an understanding of object-oriented programming in PHP. The student will be able to:
64.01	Create classes using PHP.
64.02	Describe inheritance and its role in PHP programming.
64.03	Write PHP code to handle exceptions.
64.04	Write PHP code to accommodate different interfaces.
65.0	Demonstrate proficiency in writing PHP code to handle file input/output (I/O) operations. The student will be able to:
65.01	Write PHP code to perform open, read, and write operations on files.
65.02	Write PHP code to initiate file system functions.
65.03	Write PHP code to handle streams.
66.0	Demonstrate proficiency in creating, populating, and using arrays in PHP. The student will be able to:
66.01	Create, populate and write code to extract information from a numeric array in PHP.
66.02	Create, populate and write code to extract information from an associative array in PHP.
66.03	Create, populate and write code to extract information from a multidimensional array in PHP.
67.0	Demonstrate proficiency handling strings in PHP. The student will be able to:
67.01	Write PHP code to retrieve or extract one or more characters from a string.
67.02	Write PHP code to convert a string from data type to another.
67.03	Write PHP code to manipulate the display characteristics of string data.
67.04	Write PHP code that uses string date to control program flow.
67.05	Write PHP code to join array elements with a string.
68.0	Demonstrate proficiency in using PHP to access databases via Open Database Connectivity (ODBC). The student will be able to:
68.01	Write PHP code to create an ODBC connection and retrieve information from a database using the appropriate Structured Query Language (SQL) statement.
68.02	Describe a prepared statement and discuss its primary advantages (e.g., code efficiency, SQL Injection prevention).
68.03	Create a prepared statement to perform specific SQL actions.
68.04	Describe a PHP Data Object (PDO) transaction and explain its primary advantages.
68.05	Create a prepared statement and associated result set using PDOStatement.
69.0	Demonstrate proficiency in applying best practices for ensuring creation of a secure program. The student will be able to:

CTE Standards and Benchmarks	
69.01	Describe an SQL Injection, its consequences, and ways in which it may be prevented via programming.
69.02	Describe the Remote Code Injection vulnerability in PHP and ways in which it may be prevented.
69.03	Describe the risk of session hijacking in PHP and ways to program around it.
69.04	Describe the risks associated with uploading files in PHP and ways in which risks might be mitigated.
69.05	Describe Secure Sockets Layer (SSL) and usage issues related to PHP.
70.0	Demonstrate an understanding of key technologies, protocols, and architectures associated with web development and programming. The student will be able to:
70.01	Describe SimpleXML functions.
70.02	Describe Extensible Markup Language (XML) Extension.
70.03	Describe XML Path Language (XPath).
70.04	Describe Web Services.
70.05	Describe Simple Object Access Protocol (SOAP).
70.06	Describe Representational State Transfer (REST).
70.07	Describe JavaScript Object Notation (JSON).
70.08	Describe Asynchronous JavaScript and XML (AJAX).

Additional Information

Laboratory Activities

Laboratory investigations that include scientific inquiry, research, measurement, problem solving, emerging technologies, tools and equipment, as well as, experimental, quality, and safety procedures are an integral part of this career and technical program/course. Laboratory investigations benefit all students by developing an understanding of the complexity and ambiguity of empirical work, as well as the skills required to manage, operate, calibrate and troubleshoot equipment/tools used to make observations. Students understand measurement error; and have the skills to aggregate, interpret, and present the resulting data. Equipment and supplies should be provided to enhance hands-on experiences for students.

Florida Standards for English Language Development (ELD)

English language learners communicate for social and instructional purposes within the school setting. ELD.K12.ELL.SI.1

English Language Development (ELD) Standards Special Notes:

Teachers are required to provide listening, speaking, reading and writing instruction that allows English language learners (ELL) to communicate for social and instructional purposes within the school setting. For the given level of English language proficiency and with visual, graphic, or interactive support, students will interact with grade level words, expressions, sentences and discourse to process or produce language necessary for academic success. The ELD standard should specify a relevant content area concept or topic of study chosen by curriculum developers and teachers which maximizes an ELL's need for communication and social skills. For additional information on the development and implementation of the ELD standards, please contact the Bureau of Student Achievement through Language Acquisition at sala@fldoe.org.

Career and Technical Student Organization (CTSO)

Future Business Leaders of America (FBLA), Business Professionals of America (BPA) and Florida Technology Student Association (FL-TSA) are the co-curricular career and technical student organizations providing leadership training and reinforcing specific career and technical skills for secondary students. Career and Technical Student Organizations provide activities for students as an integral part of the instruction offered.

Cooperative Training – OJT

On-the-job training is appropriate but not required for this program. Whenever offered, the rules, guidelines, and requirements specified in the OJT framework apply.

Accommodations

Federal and state legislation requires the provision of accommodations for students with disabilities as identified on the secondary student's Individual Educational Plan (IEP) or 504 plan or postsecondary student's accommodations' plan to meet individual needs and ensure equal access. Accommodations change the way the student is instructed. Students with disabilities may need accommodations in such areas as instructional methods and materials, assignments and assessments, time demands and schedules, learning environment, assistive technology and special communication systems. Documentation of the accommodations requested and provided should be maintained in a confidential file.

In addition to accommodations, some secondary students with disabilities (students with an IEP served in Exceptional Student Education (ESE)) will need modifications to meet their needs. Modifications change the outcomes or what the student is expected to learn, e.g., modifying the curriculum of a secondary career and technical education course. Note: postsecondary curriculum and regulated secondary programs cannot be modified.

Some secondary students with disabilities (ESE) may need additional time (i.e., longer than the regular school year), to master the student performance standards associated with a regular course or a modified course. If needed, a student may enroll in the same career and technical course more than once. Documentation should be included in the IEP that clearly indicates that it is anticipated that the student may need an additional year to complete a Career and Technical Education (CTE) course. The student should work on different competencies and new applications of competencies each year toward completion of the CTE course. After achieving the competencies identified for the year, the student earns credit for the course. It is important to ensure that credits earned by students are reported accurately. The district's information system must be designed to accept multiple credits for the same course number for eligible students with disabilities.