Florida CTE

Middle School CTE: Information and Communications Technology (ICT) Essentials 3 (2024)

Use Web 2.0 or Internet-based collaborative technology (e.g., Wikis, Wimba, Moodle, Edmodo, Facebook, Schoology, Gaggle) to facilitate a web development or research project. The student will be able to: 15.0	1 **Create and use a collaborative environment for communicating and sharing among project team members.** 15.1 2 **Create and use a social media page (e.g., Wikis, Wimba, Moodle, Edmodo, Facebook, Schoology, Gaggle) to share and publish project components (e.g., content, images, graphics, videos) for gauging visitor reaction and obtaining feedback.** 15.2
Demonstrate an understanding of computer networks. The student will be able to: 16.0	1 **Define "network" and give examples of networks used at home, school, and work.** 16.1 2 **Compare types of networks, including LAN, WAN, MAN, VPN, intranet, extranet, the Internet.** 16.2 3 **Compare common network topologies, including bus, star, ring, mesh.** 16.3 4 **Compare various network models and their advantages, including client/server, mainframe/terminal, peer-to-peer.** 16.4 5 **Compare various methods and media for network connections, including broadband, wireless, Bluetooth, cellular, satellite.** 16.5 6 **Describe the functions of various network hardware devices, including NIC, hub, switch, router, bridge, gateway, access point.** 16.6 7 **Describe the purpose of protocols, and identify the protocols commonly used in networks, including TCP/IP, DHCP, DNS, HTTP, FTP, IMAP, POP, SMTP.** 16.7 8 **Describe the purpose and function of IP addressing and distinguish between public and private IP addresses.** 16.8

Skip to main content

	9 **Describe the OSI reference model and its layers, including tracing the flow of data between two network nodes through the OSI layers.** 16.9
Demonstrate proficiency in webpage development. The student will be able to: 17.0	1 **Identify website domains, and relate a site's domain to its purpose.** 17.1
	2 **Relate basic components of a webpage (e.g. color, space, written content, typography, images, links, multimedia) to aesthetic, functional and/or usable design principals.** 17.2
	3 **Define aesthetic design, and explain how aesthetics can affect a visitors' perception of a website's information.** 17.3
	4 **Demonstrate knowledge of color wheel concepts and effective use of color on a website.** 17.4
	5 **Compare functional and usable design principles, and explain how usability can affect a website's success.** 17.5
	6. **Critique the aesthetic design, usability and accessibility of sample websites.** 17.6
	7 **Define multimedia, and identify its role in webpage interactivity.** 17.7
	8 **Explain the primary steps of the website planning process.** 17.8
	9 **Apply the website planning process to plan the design for basic website.** 17.9
	10 **Build the site navigation scheme for a website.** 17.10
	11 **Compare webpage creation using an HTML text editor to using a graphical user interface (GUI) editor.** 17.11
	12 **Compare website creation using an online site builder, an offline site builder and a content management system (CMS).** 17.12
	13 **Modify an existing webpage template to create an effective look and feel for a website.** 17.13
	14 **Create a website using a template.** 17.14
	15 **Define "HTML (Hypertext Markup Language)" and related terms, including tag vs. element, container vs. empty tag, block-level vs. inline element, attribute value, semantic tag.** 17.15
	16 **Identify HTML elements required to create webpage structure.** 17.16
	17 **Create webpages using basic HTML tags (e.g., headings, lists, character styles, text alignment, tables, comments).** 17.17
	Use HTML to create hyperlinks to external sites. 17.18

Skip to main content

	19 Use HTML to insert common image file formats into webpages, and use an image as a hyperlink. 17.19
	20 Explain Cascading Style Sheet (CSS) technology. 17.20
	21 Apply CSS styles to an HTML page. 17.21
	22 Create and/or edit animation files, and integrate them into a webpage. 17.22
	23 Create and/or edit video files, and integrate them into a webpage. 17.23
	24 Use Dynamic HTML (DHTML) to enhance webpage interactivity. 17.24
	25 Create and use a wiki or similar tool for collaborating among project team members. 17.25
	26 Create and use a social media page (e.g., Facebook, Wimba, Edmodo) and/or a blog to share content and collaborate on projects. 17.26
	27 Review webpage content, verify copyright restrictions, and create meta-data before publishing a site to the internet. 17.27
	28 Test webpages for display, functionality, and accessibility before publishing a site to the Internet. 17.28
	29 Validate webpage code using W3C validation tools before publishing a site to the Internet. 17.29
	30 Describe network issues relating to websites, including bandwidth, compression, streaming, web hosting. 17.30
	31 Explain the purpose of File Transfer Protocol (FTP) in accessing information on the Internet. 17.31
	32 Publish a website using FTP. 17.32
	33 Describe website security methods, including secure server vs. unsecured served, SSL, SSH, encryption. 17.33
Demonstrate proficiency in game development. The student will be able to: 18.0	**1** Describe the role of games in modern society (e.g., education, task training, social networking, therapy, recreation). 18.1
	2 Identify various types of games (e.g., chance, skill, knowledge, role-playing, and storytelling). 18.2
	3 Identify the steps of the design process for creating a game. 18.3
	4 Apply the design process to solving a problem. 18.4
	5 Analyze (deconstruct) existing games. 18.5

Skip to main content

6 **Identify the tools and skills needed for creating games.** 18.6

7 **Identify design criteria and constraints.** 18.7

8 **Create storyboards to model a game's program flow and functionality.** 18.8

9 **Identify the programmer's role in creating games.** 18.9

10 **Identify common programming languages and applications used to create computer games.** 18.10

11 **Compare sequential, iteration (loop) and selection programming structures.** 18.11

12 **Define the term algorithm (i.e., a set of repeatable steps) and how it applies to problem solving.** 18.12

13 **Create an algorithm to solve a problem or complete a task.** 18.13

14 **Use pseudo-code to model a game program's flow.** 18.14

15 **Define logic errors and identify them in a game program or model.** 18.15

16 **Explain the types and uses of variables in game programming.** 18.16

17 **Describe basic Boolean concepts, including logical operators, order of precedence, expressions.** 18.17

18 **Describe the use of events, event handlers and functions in game programming.** 18.18

19 **Describe the use of parameters and arguments in game programming.** 18.19

20 **Describe the use of objects, classes and instances in game programming.** 18.20

21 **Describe the use of properties and methods with objects in game programming.** 18.21

22 **Write appropriate code to create a simple game using structured programming.** 18.22

23 **Test and evaluate the game program you created.** 18.23

24 **Modify the game program as needed to solve a problem.** 18.24

25 **Create an animated object (i.e., sprite) to be used in a game program.** 18.25

26 **Use programming code to control the behavior of an animated object (i.e., sprite) in a game program.** 18.26

Skip to main content

Demonstrate proficiency in basic programming. The student will be able to: 19.0

1. **Define "programming" and discuss its role in computing.** 19.1
2. **Explain the binary representation of data and programs in computers.** 19.2
3. **Distinguish among the three types of programming languages (machine, assembly, high-level), and give examples.** 19.3
4. **Compare and contrast languages that are usually compiled (e.g., C++, Java) and interpreted (e.g., JavaScript, Python).** 19.4
5. **Describe the structure of a simple program, and explain why sequencing is important.** 19.5
6. **Write a program design document using pseudo-code that shows program flow.** 19.6
7. **Explain strategies used in problem-solving, and relate them to computer programming.** 19.7
8. **Define the term "algorithm," and explain how it relates to problem-solving.** 19.8
9. **Explain the three types of programming errors (i.e., logic, syntax, runtime), and describe the forms of testing that can be used to locate and debug errors.** 19.9
10. **Solve a problem using logic by planning a strategy, designing and testing a hypothesis, and/or creating a set of step-by-step instructions to perform a task.** 19.10
11. **Define "structured programming" and discuss the advantages of this approach.** 19.11
12. **Define the three main programming control structures used in structured programming: sequential, selection (decision), and iteration (loops).** 19.12
13. **Describe iterative programming structures (e.g., while, do/while) and how they are used in programming.** 19.13
14. **Describe selection programming structures (e.g., if/then, else) and explain the logic used for if statements.** 19.14
15. **Write a simple program in pseudo-code that uses structured programming to solve a problem.** 19.15
16. **Explain the types and uses of variables in programming.** 19.16
17. **Explain basic object-oriented concepts.** 19.17
18. **Describe fundamental Boolean concepts, including Boolean algebra, operators, logic.** 19.18

Skip to main content

19 **Create animated objects using a high-level programming environment (e.g., Alice, Greenfoot) to control their behavior.** 19.19

20 **Create a simple program that uses animated objects.** 19.20

21 **Convert a simple program from pseudo-code into a common high-level programming environment (e.g. Alice, Greenfoot).** 19.21

22 **Troubleshoot and debug errors in code.** 19.22