

CompuScholar, Inc.

Alignment to South Carolina **Computer Programming 1 with Python (5064)** Standards

South Carolina Course Details:

Course Title:	5064 - Computer Programming 1 with Python
Grade Level:	9th - 12th grades
Standards Link:	Computer Programming 1 with Python (Feb 2021 Version)

CompuScholar Course Details:

Course Title:	Computer Science Foundations
Course ISBN:	978-1-946113-02-3
Course Year:	2024

Note 1: Citation(s) listed may represent a subset of the instances where objectives are met throughout the course.

Note 2: Citation(s) for a "Lesson" refer to the "Lesson Text" elements and associated "Activities" within the course, unless otherwise noted. The "Instructional Video" components are supplements designed to introduce or reinforce the main lesson concepts, and the Lesson Text contains full details.

Note 3: "Supplemental" or "Suppl." citation(s) refer to Supplemental chapters included at the end of the course.

Course Description

The Computer Programming 1 with Python course is designed to emphasize the fundamentals of computer programming using Python. Topics include computer software, program design and development, and practical experience in programming, using modern, object-oriented languages.

Course Standards

A. SAFETY	CITATION(S)
1. Review school safety policies and procedures.	Chapter 29, Lesson 3
2. Review classroom safety rules and procedures.	Chapter 29, Lesson 3
3. Review safety procedures for using equipment in the classroom.	Chapter 29, Lesson 3
4. Identify major causes of work-related accidents in office environments.	Chapter 29, Lesson 3
5. Demonstrate safety skills in an office/work environment.	Chapter 29, Lesson 3

B. STUDENT ORGANIZATIONS	CITATION(S)
1. Identify the purpose and goals of a Career and Technology Student Organization (CTSO).	Chapter 29, Lesson 4
2. Explain how CTSOs are integral parts of specific clusters, majors, and/or courses.	Chapter 29, Lesson 4
3. Explain the benefits and responsibilities of being a member of a CTSO.	Chapter 29, Lesson 4
4. List leadership opportunities that are available to students through participation in CTSO conferences, competitions, community service, philanthropy, and other activities.	Chapter 29, Lesson 4
5. Explain how participation in CTSOs can promote lifelong benefits in other professional and civic organizations.	Chapter 29, Lesson 4

C. TECHNOLOGY KNOWLEDGE	CITATION(S)
1. Demonstrate proficiency and skills associated with the use of technologies that are common to a specific occupation.	Throughout the course ,e.g.: Chapter 3 (coding) Chapter 14 (collaboration) Chapters 26-28 (computing skills)
2. Identify proper netiquette when using e-mail, social media, and other technologies for communication purposes.	Chapter 19, Lesson 1
3. Identify potential abuse and unethical uses of laptops, tablets, computers, and/or networks.	Chapters 19, 20
4. Explain the consequences of social, illegal, and unethical uses of technology (e.g., piracy; illegal downloading; licensing infringement; inappropriate uses of software, hardware, and mobile devices in the work environment).	Chapter 18, Lesson 1, 2 Chapters 19, 20
5. Discuss legal issues and the terms of use related to copyright laws, fair use laws, and ethics pertaining to downloading of images, photographs, documents, video, sounds, music, trademarks, and other elements for personal use.	Chapter 19, Lesson 3
6. Describe ethical and legal practices of safeguarding the confidentiality of business-related information.	Chapter 19, Lessons 1, 2 Chapter 20, Lessons 2, 3
7. Describe possible threats to a laptop, tablet, computer, and/or network and methods of avoiding attacks.	Chapter 20

D. PERSONAL QUALITIES AND EMPLOYABILITY SKILLS	CITATION(S)
1. Demonstrate creativity and innovation.	Chapter 14, Lesson 1 Chapter 14, Activity 1
2. Demonstrate critical thinking and problem-solving skills.	Chapter 14, Lesson 1 Chapter 14, Activity 1
3. Demonstrate initiative and self-direction.	Chapter 14, Lesson 1 Chapter 14, Activity 1

4. Demonstrate integrity.	Chapter 14, Lesson 1 Chapter 14, Activity 1
5. Demonstrate work ethic.	Chapter 14, Lesson 1 Chapter 14, Activity 1
6. Demonstrate conflict resolution skills.	Chapter 14, Lesson 1 Chapter 14, Activity 1
7. Demonstrate listening and speaking skills.	Chapter 14, Lesson 1 Chapter 14, Activity 1
8. Demonstrate respect for diversity.	Chapter 14, Lesson 1 Chapter 14, Activity 1
9. Demonstrate customer service orientation.	Chapter 14, Lesson 1 Chapter 14, Activity 1
10. Demonstrate teamwork.	Chapter 14, Lesson 1 Chapter 14, Activity 1

E. PROFESSIONAL KNOWLEDGE	CITATION(S)
1. Demonstrate global or “big picture” thinking.	Chapter 14, Lessons 2, 3 / Activity 1
2. Demonstrate career and life management skills and goal-making.	Chapter 14, Lesson 2 / Activity 1 Chapter 29, Lesson 1 / Activity
3. Demonstrate continuous learning and adaptability skills to changing job requirements.	Chapter 14, Lessons 3, 4 / Activities 2, 3
4. Demonstrate time and resource management skills.	Chapter 14, Lesson 2 / Activity 1
5. Demonstrates information literacy skills.	Chapter 17 and throughout the course
6. Demonstrates information security skills.	Chapter 20
7. Demonstrates information technology skills.	Chapters 26 - 28
8. Demonstrates knowledge and use of job-specific tools and technologies.	Chapter 3 (coding) Chapter 7 (debugging) Chapter 14 (SDLC) Chapter 15 (design tools)
9. Demonstrate job-specific mathematics skills.	Chapter 15, Lesson 3 Chapter 16, Lesson 1 Chapter 17, Lesson 3
10. Demonstrates professionalism in the workplace.	Chapter 14, Lesson 1 Chapter 14, Activity 1
11. Demonstrates reading and writing skills.	Chapter 1, Lesson 1 Chapter 14, Lesson 3 / Activity 1
12. Demonstrates workplace safety.	Chapter 29, Lesson 3

F. COMPUTING SYSTEMS	CITATION(S)
1. Describe the hardware requirements needed to run operational systems and systems software. (ex. application software, database management, networking software, etc.).	Chapter 1, Lessons 1, 2, 3
2. Compare and contrast the elements of a computing system by examining hardware elements for their intended use (e.g., input-output (I/O) devices, random access memory (RAM), read only memory (ROM), storage devices, motherboards, and processors...	Chapter 1, Lesson 2
3. Describe the various data storage tools and data organization methods.	Chapter 1, Lesson 2 Chapters 4, 9, 17
4. Research computing solutions to problems in different countries, considering the personal, ethical, social, economic, and cultural impact	Chapter 18

G. PROGRAM DEVELOPMENT AND DESIGN	CITATION(S)
PROBLEM-SOLVING AND PROGRAM DESIGN	
1. Document steps in the design process.	Chapter 14, Lesson 2 / Activity 1
2. Develop an algorithm for a program using a design tool (e.g., pseudocode, flowcharts, human-language algorithm, etc.).	Chapter 15, Lessons 1, 2
3. Construct data flows based on program requirements.	Chapter 14, Lesson 3 / Activity 2 Chapter 15, Lesson 3 Chapter 16, Lessons 2, 3
4. Utilize version control as a part of the design process to document revisions in an iterative development cycle.	Chapter 14, Lesson 3 / Activity 1
5. Identify the advantages and disadvantages of diverse perspectives and backgrounds when solving computational problems.	Chapter 14, Lesson 1 Chapter 17, Lesson 2
6. Make use of standard programming control structures during algorithm design.	Chapters 6, 8, 15
7. Analyze the sequence of instructions to determine if they will accomplish a task.	Chapter 7, Lessons 2, 3 Chapter 15 Chapter 16, Lesson 2
8. Create algorithms to solve computational problems that have an application in the real world (e.g., local community, church, civic organization, school, home life).	Chapters 14, 15 Chapter 16, Lessons 2, 3 / Activity
9. Adapt predefined algorithms to solve computational problems.	Chapter 10, Lessons 2, 3 Chapter 15, Lesson 3 Chapter 17, Lesson 3
10. Select appropriate data types to store information used in the program.	Chapter 4 Chapter 9, Lesson 1
11. Demonstrate an understanding of how to collect requirements.	Chapter 14, Lesson 3 Chapter 15, Lesson 1

12. Decompose tasks into smaller, reusable parts to facilitate the design, implementation, and review of programs.	Chapter 14, Lessons 1, 4
DATA STRUCTURES	
1. Create valid variables and constants using appropriate data types to store information used in the program.	Chapter 4, Lessons 2, 3, 4
2. Determine the scope and lifetime of variables (e.g., local, global, static).	Chapter 12, Lesson 3
3. Create valid variables and constants using appropriate scope (e.g., local, global, static).	Chapter 4, Lessons 2, 3, 4 Chapter 12, Lesson 3
4. Describe the properties of a data set that could be used to explore a real world phenomenon or support a claim.	Chapter 16, Lesson 3 Chapter 17, Lessons 1, 2
5. Compare and contrast data sets that could be used to explore a real-world phenomenon or support a claim.	Chapter 17, Lessons 1, 2
6. Create data sets that could be used to explore a real world phenomenon or support a claim.	Chapter 16, Lesson 3 Chapter 17, Lessons 1, 2
7. Organize collected data to communicate the solution to a real-world phenomenon and support a claim.	Chapter 16, Lesson 3 Chapter 17, Lessons 1, 2, 3 / Activity
CONTROL STRUCTURES	
1. Summarize the differences of Sequential Programming and Event Driven Programming.	Chapter 5, Lesson 2
2. Develop an interactive program that includes features to get input and provide feedback/information (e.g., alerts, messages, input boxes).	Chapter 5, Lesson 2 (and most later exercises include interactive I/O)
3. Develop a program that correctly utilizes conditionals (if, else if, else, switch) to produce multiple outcomes based on input given from a user.	Chapter 6
4. Develop a program that correctly utilizes the different Control structures (e.g., Sequence logic, Selection logic, iteration Logic) to basically analyze and choose in which direction a program flows based on certain parameters and conditions.	Chapters 6, 8
5. Develop a program that correctly utilizes Loops to produce multiple outcomes based on input given from a user.	Chapter 8
6. Trace the flow of execution of a program that uses a combination of control structures (e.g., conditionals, loops, event handlers, recursion).	Chapter 6, Lesson 4 Chapter 7, Lessons 2, 3 Chapter 8, Lessons 2, 3
7. Design and iteratively develop programs that combine control structures (e.g., conditionals, loops, event handlers, recursion).	Chapter 8, Lesson 3 / Activity Chapter 14, Activities 1, 2, 3
8. Trace the flow of execution of a program that uses a variety of programming constructs (e.g., procedures, modules, objects).	Chapter 7, Lessons 2, 3 Chapter 10, Lessons 1, 2, 3 Chapter 12, Lessons 1, 4
9. Design a solution through systematic analysis using programming constructs (e.g., procedures, modules, objects).	Chapter 10, Lessons 1, 2, 3 Chapter 12, Lessons 1, 4

10. Explain different decision structures that control program flow.	Chapters 6, 8
11. Select from different looping/iteration structures that control program flow.	Chapter 8
TESTING, DEBUGGING, AND REVISIONS	
1. Plan and develop programs for a variety of audiences using a process that incorporates development, feedback, and revision.	Chapter 14, Lessons 2, 3, 4 and associated Activities
a. Review the program.	Chapter 7, Lessons 2, 3 Chapter 14, Lesson 4 / Activity 3
b. Build the program.	Chapter 14, Activities 2, 3
c. Execute the program to test the logical validity of an application program given appropriate data.	Chapter 7, Lessons 2, 3 Chapter 14, Lesson 4 / Activity 3
d. Identify values of variables at different points in the flow of execution.	Chapter 7, Lessons 2, 3 Chapter 14, Lesson 4 / Activity 3
e. Debug the program for errors (e.g., syntax and build errors).	Chapter 7, Lessons 2, 3
2. Systematically test programs using a range of test cases to meet design specifications (e.g., specific outcomes, functionality, user interface, error handling).	Chapter 14, Lesson 4 / Activity 3
a. Develop a test strategy.	Chapter 14, Lesson 4 / Activity 3
b. Create a test plan.	Chapter 14, Lesson 4 / Activity 3
c. Design a test suite of conditions to evaluate best and worst cases of a program.	Chapter 14, Lesson 4 / Activity 3
d. Identify the difference between a test case and test script.	Chapter 14, Lesson 4
e. Create a test script.	Chapter 14, Activity 3
f. Demonstrate the ability to debug the program for errors (e.g., run-time/exception, logic/semantic).	Chapter 7, Lessons 2, 3
3. Develop code to solve the smaller parts of a decomposed task that can be reused to solve similar problems (e.g., procedures, functions, objects).	Chapter 12
4. Seek and incorporate feedback to refine a solution (e.g., users, team members, code review, teachers).	Chapter 14, Activities 2, 3, 4