CompuScholar, Inc.

Alignment to South Carolina **Introduction to Computer Programming (5050)** Standards
(Previously **Computer Programming 1**)

South Carolina Course Details:

Course Title:	5050 - Introduction to Computer Programming
Grade Level:	9th - 12th grades
Standards Link:	Introduction to Computer Programming (Jan 2021 Version)

CompuScholar Course Details:

Course Title:	Java Programming
Course ISBN:	978-1-946113-99-3
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

Introduction to Computer Programming, formerly known as Computer Programming 1, is designed to emphasize the fundamentals of computer programming. Topics include computer software, program design and development, and practical experience in programming using modern, text-based programming languages.

Course Standards

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

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

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 2 (coding) Chapter 11 (debugging) Chapter 13 (design tools) Chapter 27 (SDLC)
2. Identify proper netiquette when using e-mail, social media, and other technologies for communication purposes.	Chapter 1, Lesson 4
3. Identify potential abuse and unethical uses of laptops, tablets, computers, and/or networks.	Chapter 1, Lessons 4, 5 Supplemental Chapter 3, Lessons 1, 2
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 1, Lessons 4, 5 Supplemental Chapter 3, Lessons 1, 2
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.	Supplemental Chapter 3, Lesson 5
6. Describe ethical and legal practices of safeguarding the confidentiality of business-related information.	Chapter 1, Lessons 4, 5
7. Describe possible threats to a laptop, tablet, computer, and/or network and methods of avoiding attacks.	Chapter 1, Lessons 4, 5 Supplemental Chapter 3, Lesson 1

D. PERSONAL QUALITIES AND EMPLOYABILITY SKILLS	CITATION(S)
1. Demonstrate creativity and innovation.	Chapter 27, Lesson 1 Chapter 27, Activity 1
2. Demonstrate critical thinking and problem-solving skills.	Chapter 27, Lesson 1 Chapter 27, Activity 1

3. Demonstrate initiative and self-direction.	Chapter 27, Lesson 1 Chapter 27, Activity 1
4. Demonstrate integrity.	Chapter 27, Lesson 1 Chapter 27, Activity 1
5. Demonstrate work ethic.	Chapter 27, Lesson 1 Chapter 27, Activity 1
6. Demonstrate conflict resolution skills.	Chapter 27, Lesson 1 Chapter 27, Activity 1
7. Demonstrate listening and speaking skills.	Chapter 27, Lesson 1 Chapter 27, Activity 1
8. Demonstrate respect for diversity.	Chapter 27, Lesson 1 Chapter 27, Activity 1
9. Demonstrate customer service orientation.	Chapter 27, Lesson 1 Chapter 27, Activity 1
10. Demonstrate teamwork.	Chapter 27, Lesson 1 Chapter 27, Activity 1

E. PROFESSIONAL KNOWLEDGE	**CITATION(S)**
1. Demonstrate global or “big picture” thinking.	Chapter 27, Lessons 1, 2 / Activity 1
2. Demonstrate career and life management skills and goal-making.	Chapter 27, Lesson 1 / Activity 1
3. Demonstrate continuous learning and adaptability skills to changing job requirements.	Chapter 27, Lessons 2, 3 / Activities 2, 3
4. Demonstrate time and resource management skills.	Chapter 27, Lesson 1 / Activity 1
5. Demonstrates information literacy skills.	Supplemental Chapter 1, Lessons 1, 2, 6 Supplemental Chapter 2, Lesson 4
6. Demonstrates information security skills.	Chapter 1, Lesson 5 Supplemental Chapter 3, Lesson 1
7. Demonstrates information technology skills.	Chapters 1, 11, 28 Supplemental Chapter 1, Lessons 1, 2
8. Demonstrates knowledge and use of job-specific tools and technologies.	Chapter 2 (coding) Chapter 11 (debugging) Chapter 13 (design tools) Chapter 27 (SDLC)
9. Demonstrate job-specific mathematics skills.	Chapter 4, Lesson 1 Chapter 7 Chapter 13, Lesson 3
10. Demonstrates professionalism in the workplace.	Chapter 27, Lesson 1 and throughout the collaborative project

11. Demonstrates reading and writing skills.	Chapter 1, lesson 1 Chapter 27, Lesson 2 / Activity 1
12. Demonstrates workplace safety.	Supplemental Chapter 2, Lesson 5

F. COMPUTING SYSTEMS	CITATION(S)
1. Describe the hardware requirements needed to run operational systems and systems software. (ex,. application software, database	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 3, 5, 14, 18, 19, 21
4. Research computing solutions to problems in different countries, considering the personal, ethical, social, economic, and cultural impact	Supplemental Chapter 3, Lesson 2, 3 / Activities 2, 3

G. PROGRAM DEVELOPMENT AND DESIGN	CITATION(S)
PROBLEM-SOLVING AND PROGRAM DESIGN	
1. Document steps in the design process.	Chapter 27, Lessons 1, 2
2. Develop an algorithm for a program using a design tool (e.g., pseudocode, flowcharts, human-language algorithm, etc.).	Chapter 13, Lessons 1, 2
3. Construct data flows based on program requirements.	Chapter 13 Chapter 27, Lesson 2 / Activities 1, 2
4. Utilize version control as a part of the design process to document revisions in an iterative development cycle.	Chapter 27, Lesson 2 / Activity 1
5. Identify the advantages and disadvantages of diverse perspectives and backgrounds when solving computational problems.	Chapter 27, Lesson 1
6. Make use of standard programming control structures during algorithm design.	Chapters 8, 12, 13, 20, 24
7. Analyze the sequence of instructions to determine if they will accomplish a task.	Chapters 11, 13
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 13, 17, 20, 24, 27
9. Adapt predefined algorithms to solve computational problems.	Chapter 7, Lesson 3 Chapter 13, Lessons 3, 4 Chapters 20, 24
10. Select appropriate data types to store information used in the program.	Chapters 3, 18, 19, 21
11. Demonstrate an understanding of how to collect requirements.	Chapter 27, Lesson 2 / Activity 1 Supplemental Chapter 2, Lesson 1

12. Decompose tasks into smaller, reusable parts to facilitate the design, implementation, and review of programs.	Chapter 14, Lessons 1, 2, 5 Chapters 15, 22
DATA STRUCTURES	
1. Create valid variables and constants using appropriate data types to store information used in the program.	Chapter 3, Lessons 1, 2
2. Determine the scope and lifetime of variables (e.g., local, global, static).	Chapter 15, Lesson 2
3. Create valid variables and constants using appropriate scope (e.g., local, global, static).	Chapter 3, Lessons 1, 2 Chapter 15, Lesson 2
4. Describe the properties of a data set that could be used to explore a real world phenomenon or support a claim.	Chapter 27, Activity 1 Chapter 33 Supplemental Chapter 1, Lesson 4
5. Compare and contrast data sets that could be used to explore a real-world phenomenon or support a claim.	Chapter 27, Activity 1 Chapter 33 Supplemental Chapter 1, Lesson 4
6. Create data sets that could be used to explore a real world phenomenon or support a claim.	Chapter 27, Activity 1 Chapter 33 Supplemental Chapter 1, Lesson 4
7. Organize collected data to communicate the solution to a real-world phenomenon and support a claim.	Chapter 27, Activity 1 Chapter 33 Supplemental Chapter 1, Lesson 4
CONTROL STRUCTURES	
1. Summarize the differences of Sequential Programming and Event Driven Programming.	Chapter 30, Lesson 3
2. Develop an interactive program that includes features to get input and provide feedback/information (e.g., alerts, messages, input boxes).	Chapter 5, Lesson 6 (and most later exercises include interactive I/O) Chapters 30, 31
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 8
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 8, 9, 12, 24
5. Develop a program that correctly utilizes Loops to produce multiple outcomes based on input given from a user.	Chapter 12
6. Trace the flow of execution of a program that uses a combination of control structures (e.g., conditionals, loops, event handlers, recursion).	Chapter 8, Lesson 3 Chapters 11, 20, 24 Chapter 12, Lesson 4
7. Design and iteratively develop programs that combine control structures (e.g., conditionals, loops, event handlers, recursion).	Chapter 27, 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).	Chapters 11, 14, 15, 22, 23

9. Design a solution through systematic analysis using programming constructs (e.g., procedures, modules, objects).	Chapters 14, 15, 22, 23, 27
10. Explain different decision structures that control program flow.	Chapters 8, 9, 12, 24
11. Select from different looping/iteration structures that control program flow.	Chapters 12, 13, 24
TESTING, DEBUGGING, AND REVISIONS	
1. Plan and develop programs for a variety of audiences using a process that incorporates development, feedback, and revision.	Chapter 27, Lessons 1, 2, 3 and associated Activities
a. Review the program.	Chapter 11 Chapter 27, Lesson 3 / Activity 3
b. Build the program.	Chapter 27, Activities 2, 3
c. Execute the program to test the logical validity of an application program given appropriate data.	Chapter 11 Chapter 27, Lesson 3 / Activity 3
d. Identify values of variables at different points in the flow of execution.	Chapter 11 Chapter 27, Lesson 3 / Activity 3
e. Debug the program for errors (e.g., syntax and build errors).	Chapter 11
2. Systematically test programs using a range of test cases to meet design specifications (e.g., specific outcomes, functionality, user interface, error handling).	Chapter 27, Lesson 3 / Activity 3
a. Develop a test strategy.	Chapter 27, Lesson 3 / Activity 3
b. Create a test plan.	Chapter 27, Lesson 3 / Activity 3
c. Design a test suite of conditions to evaluate best and worst cases of a program.	Chapter 11, Lesson 1 Chapter 27, Lesson 3 / Activity 3
d. Identify the difference between a test case and test script.	Chapter 27, Lesson 3
e. Create a test script.	Chapter 27, Activity 3
f. Demonstrate the ability to debug the program for errors (e.g., run-time/exception, logic/semantic).	Chapter 11
3. Develop code to solve the smaller parts of a decomposed task that can be reused to solve similar problems (e.g., procedures, functions,	Chapters 14, 15, 22, 23, 27
4. Seek and incorporate feedback to refine a solution (e.g., users, team members, code review, teachers).	Chapter 27, Activities 1, 2, 3