## CompuScholar, Inc.

# Correlations to the Texas Essential Knowledge and Skills (TEKS): Computer Science II

#### **Texas Course Details:**

Chapter	Chapter 127. Texas Essential Knowledge and Skills for CTE
Subchapter	Subchapter O. STEM
Course	§127.790 Computer Science II
Standards	Subchapter O (STEM)
TEKS Coverage	100%

#### **CompuScholar Course Details:**

Course Title:	Java Programming
Course ISBN:	978-1-946113-99-3
Course Year:	2023

**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 re-enforce the main lesson concepts, and the Lesson Text contains full details.

### **Course Standards**

**Knowledge and Skills Statement**: (1) Employability. The student identifies various employment opportunities in the computer science field. The student is expected to:

Student Expectation	Citation(s)
(1.A) identify job opportunities and accompanying job duties and tasks;	Supplemental Chapter 2, Lesson 2
(1.B) examine the role of certifications, resumes, and portfolios in the computer science profession;	Supplemental Chapter 2, Lesson 2
(1.C) employ effective technical reading and writing skills;	Chapter 2, Lesson 4 Chapter 27, Lesson 2
	Supplemental Chapter 2, Lesson 4
(1.D) employ effective verbal and non-verbal communication skills;	Supplemental Chapter 2, Lesson 2
(1.E) solve problems and think critically;	Chapter 11, Lesson 1
	Chapter 13, Lesson 3

(1.F) demonstrate leadership skills and function effectively as a team	Chapter 27, Lessons 1, 2
member;	Supplemental Chapter 2, Lesson 2
(1.G) communicate an understanding of legal and ethical	Chapter 1, Lessons 4, 5
responsibilities in relation to the field of computer science;	Supplemental Chapter 3, Lesson 2
(1.H) demonstrate planning and time-management skills; and	Chapter 27, Lessons 1, 2
(1.I) compare university computer science programs.	Supplemental Chapter 2, Lesson 2

Knowledge and Skills Statement: (2) Creativity and innovation. The student develops products and generates new understandings by extending existing knowledge. The student is expected to:

Student Expectation	Citation(s)
(2.A) use program design problem-solving strategies to create program	Chapter 13, Lessons 1, 2
solutions;	Chapter 27, Lesson 2
(2.B) read, analyze, and modify programs and their accompanying	Chapter 2, Lessons 3, 4
documentation such as an application programming interface (API),	Chapter 10, Lesson 3
internal code comments, external documentation, or readme files;	Chapter 11, Lesson 1
	Chapter 13, Lesson 3
(2.C) follow a systematic problem-solving process that identifies the	Chapter 11, Lessons 1, 2
purpose and goals, the data types and objects needed, and the	Chapter 27, Lessons 1, 2
subtasks to be performed;	
(2.D) compare design methodologies and implementation techniques	Chapter 14, Lesson 1
such as top-down, bottom-up, and black box;	Chapter 15, Lesson 3
	Chapter 22, Lesson 1
	Chapter 27, Lesson 2
(2.E) trace a program, including inheritance and black box	Chapter 11, Lesson 1
programming;	Chapter 14, Lesson 1
	Chapter 15, Lesson 3
(2.F) choose, identify, and use the appropriate abstract data type,	Chapter 14, Lesson 1
advanced data structure, and supporting algorithms to properly	Chapter 18, Lesson 1
represent the data in a program problem solution; and	Chapter 19, Lesson 1
	Chapter 21, Lessons 1, 4
(2.G) use object-oriented programming development methodology,	Chapter 14, Lesson 1
including data abstraction, encapsulation with information hiding,	Chapter 15, Lesson 3
inheritance, and procedural abstraction in program development.	Chapter 22, Lesson 1
	Chapter 23, Lesson 2

**Knowledge and Skills Statement**: (3) Communication and collaboration. The student communicates and collaborates with peers to contribute to his or her own learning and the learning of others. The student is expected to:

Student Expectation	Citation(s)
(3.A) use the principles of software development to work in software	Chapter 27, Lessons 1, 2, 3
design teams;	

(3.B) break a problem statement into specific solution requirements;	Chapter 13, Lessons 1, 2 Chapter 27, Lesson 2
(3.C) create a program development plan;	Chapter 27, Lessons 1, 2
(3.D) code part of a solution from a program development plan while a partner codes the remaining part;	Chapter 27, Lesson 1
(3.E) collaborate with a team to test a solution, including boundary and standard cases; and	Chapter 27, Lesson 3
(3.F) develop presentations to report the solution findings.	Chapter 27, Lesson 3 Supplemental Chapter 3, Lesson 4

**Knowledge and Skills Statement**: (4) Data literacy and management. The student locates, analyzes, processes, and organizes data. The student is expected to:

Student Expectation	Citation(s)
(4.A) use programming file structure and file access for required	Chapter 25, Lessons 1, 2, 3
resources;	
(4.B) acquire and process information from text files, including files of	Chapter 25, Lesson 2
known and unknown sizes;	
(4.C) manipulate data using string processing;	Chapter , Lessons 2, 3
	Chapter 25, Lesson 2
(4.D) manipulate data values by casting between data types;	Chapter 4, Lesson 3
	Chapter 22, Lesson 3
(4.E) use the structured data type of one-dimensional arrays to	Chapter 18, Lessons 1, 2, 3
traverse, search, modify, insert, and delete data;	Chapter 20, Lesson 4
(4.F) identify and use the structured data type of two-dimensional	Chapter 21, Lessons 1, 2, 4
arrays to traverse, search, modify, insert, and delete data;	
(4.G) identify and use a list object data structure to traverse, search,	Chapter 19, Lessons 1, 2, 3, 4
insert, and delete data; and	
(4.H) differentiate between categories of programming languages,	Chapter 2, Lesson 1
including machine, assembly, high-level compiled, high-level	

**Knowledge and Skills Statement**: (5) Critical thinking, problem solving, and decision making. The student uses appropriate strategies to analyze problems and design algorithms. The student is expected to:

Student Expectation	Citation(s)
(5.A) develop sequential algorithms using branching control	Chapter 8, Lessons 2, 3
making problems;	
(5.B) develop choice algorithms using selection control statements	Chapter 8, Lesson 4
based on ordinal values;	
(5.C) demonstrate the appropriate use of short-circuit evaluation in	Chapter 9, Lesson 2
certain situations;	
(5.D) use Boolean algebra, including De Morgan's Law, to evaluate and	Chapter 9, Lesson 3
simplify logical expressions;	

(5.E) develop iterative algorithms using nested loops;	Chapter 12, Lesson 4
(5.F) identify, trace, and appropriately use recursion in programming solutions, including algebraic computations;	Chapter 24, Lesson 1
(5.G) trace, construct, evaluate, and compare search algorithms,	Chapter 20, Lesson 4
including linear searching and binary searching;	Chapter 33, Lessons 1, 2
(5.H) identify, describe, trace, evaluate, and compare standard sorting	Chapter 20, Lessons 1, 2, 3
algorithms, including selection sort, bubble sort, insertion sort, and	Chapter 24, Lesson 3
merge sort;	Chapter 33, Lessons 1, 2
(5.I) measure time and space efficiency of various sorting algorithms,	Chapter 33, Lesson 2
including analyzing algorithms using "big-O" notation for best, average,	
and worst-case data patterns;	
(5.J) develop algorithms to solve various problems such as factoring,	Chapter 13, Lesson 3
summing a series, finding the roots of a quadratic equation, and	Chapter 18, Lesson 4
generating Fibonacci numbers;	Chapter 19, Lesson 4
(5.K) test program solutions by investigating boundary conditions;	Chapter 11, Lesson 1
testing classes, methods, and libraries in isolation; and performing	Chapter 17, Lesson 1
stepwise refinement;	Chapter 27, Lessons 2, 3
(5.L) identify and debug compile, syntax, runtime, and logic errors;	Chapter 11, Lesson 1
(5.M) compare efficiency of search and sort algorithms by using	Chapter 20, Lessons 1, 2
informal runtime comparisons, exact calculation of statement	Chapter 33, Lessons 1, 2
execution counts, and theoretical efficiency values using "big-O"	
notation, including worst-case, best-case, and average-case time/space	
analysis;	
(5.N) count, convert, and perform mathematical operations in the	Chapter 7, Lesson 2
decimal, binary, octal, and hexadecimal number systems;	Chapter 12, Lesson 3
(5.0) identify maximum integer boundary, minimum integer boundary,	Chapter 7, Lessons 1, 4
imprecision of real number representations, and round-off errors;	
(5.P) create program solutions to problems using a mathematics	Chapter 7, Lesson 3
library;	
(5.Q) use random number generator algorithms to create simulations;	Chapter 7, Lesson 3
(5.R) use composition and inheritance relationships to identify and	Chapter 14, Lessons 1, 3, 4
create class definitions and relationships;	Chapter 22, Lessons 1, 2
(5.S) explain and use object relationships between defined classes,	Chapter 15, Lesson 5
abstract classes, and interfaces;	Chapter 22, Lesson 1
	Chapter 23, Lessons 2, 3
(5.T) create object-oriented class definitions and declarations using	Chapter 14, Lessons 1, 3, 4, 5
variables, constants, methods, parameters, and interface	Chapter 15, Lessons 3, 5
implementations;	
(5.U) create adaptive behaviors using polymorphism;	Chapter 23, Lessons 1, 2, 3

(5.V) use reference variables for object and string data types;	Chapter 5, Lesson 2
	Chapter 6, Lesson 1
	Chapter 14, Lesson 3
(5.W) use value and reference parameters appropriately in method	Chapter 5, Lessons 3, 4, 5
definitions and method calls;	Chapter 14, Lessons 4, 5
(5.X) implement access scope modifiers;	Chapter 14, Lessons 1, 3, 5
(5.Y) use object comparison for content quality;	Chapter 6, Lesson 1
	Chapter 9, Lesson 1
(5.Z) duplicate objects using the appropriate deep or shallow copy;	Chapter 26, Lesson 3
(5.AA) apply functional decomposition to a program solution;	Chapter 26, Lesson 1
	Chapter 27, Lesson 2
(5.BB) create objects from class definitions through instantiation; and	Chapter 5, Lessons 2, 5
	Chapter 14, Lesson 4
(5.CC) examine and mutate the properties of an object using accessors	Chapter 14, Lesson 3
and modifiers	Chapter 15, Lesson 3