

CompuScholar, Inc.Correlations to the Texas Essential Knowledge and Skills (TEKS):
"Computer Science II"**Texas Course Details:**

Chapter	Chapter 126. Texas Essential Knowledge and Skills for Tech. Apps
Subchapter	Subchapter C. High School
Course	§126.34. Computer Science II
TEKS Coverage	100%

CompuScholar Course Details:

Course Title:	Java Programming
Course ISBN:	978-1-946113-99-3
Course Year:	2022

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) Creativity and innovation. The student develops products and generates new understandings by extending existing knowledge. The student is expected to:	
Student Expectation	Citation(s)
(A) use program design problem-solving strategies to create program solutions	Chapter 13, Lessons 1, 2 Chapter 27
(B) demonstrate the ability to read and modify large programs, including design description and process development	Chapters 17, 27
(C) follow the systematic problem-solving process of identifying the specifications of purpose and goals, the data types and objects needed, and the subtasks to be performed	Chapter 27 Lessons 1, 2 Chapter 27, Actiities 1, 2 Supplemental Chapter 2, Lesson 1
(D) compare and contrast design methodologies and implementation techniques such as top-down, bottom-up, and black box	Chapter 13, Lessons 1, 2 Chapter 27, Lessons 1, 2 Supplemental Chapter 2, Lesson 1
(E) analyze, modify, and evaluate existing code by performing a case study on a large program, including inheritance and black box programming	Chapters 17, 27

(F) identify the data types and objects needed to solve a problem	Chapter 3, Lesson 1 Chapter 5, Lessons 1, 2 Chapter 14, Lessons 1, 2
(G) choose, identify, and use the appropriate abstract data type, advanced data structure, and supporting algorithms to properly represent the data in a program problem solution	Chapters 17, 19, 20, 21, 22, 23, 27
(H) use object-oriented programming development methodology, data abstraction, encapsulation with information hiding, and procedural abstraction in program development and testing	Chapters 14, 15, 16, 17, 22, 23, 27
(I) create, edit, and manipulate bitmap images that are used to enhance user interfaces and program functionality	Chapter 32, Lesson 3

Knowledge and Skills Statement: (2) 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)
(A) use the principles of software engineering to work in software design teams, break a problem statement into specific solution requirements, create a program development plan, code part of a solution from a program development plan while a partner codes the remaining part, team test the solution for correctness, and develop presentations to report the solution findings	Chapter 27
(B) create interactive console display interfaces with appropriate user prompts	Chapter 5, Lesson 5
(C) create interactive human interfaces to acquire data from a user and display program results using an advanced Graphical User Interface (GUI)	Chapters 30, 31
(D) write programs and communicate with proper programming style to enhance the readability and functionality of the code by using meaningful descriptive identifiers, internal comments, white space, indentation, and a standardized program style	Chapter 2, Lesson 3 Chapter 3, Lesson 2 Supplemental Chapter 1, Lesson 2
(E) improve data display by optimizing data visualization	Chapter 6, Lesson 3
(F) display simple vector graphics to interpret and display program results	Chapter 32, Lesson 2
(G) display simple bitmap images	Chapter 32, Lesson 3

Knowledge and Skills Statement: (3) Research and information fluency. The student locates, analyzes, processes, and organizes data. The student is expected to:

Student Expectation	Citation(s)
(A) use local area networks (LANs) and wide area networks (WANs), including the Internet and intranets, in research, file management, and collaboration	Supplemental Chapter 4
(B) understand programming file structure and file access for required resources	Chapter 25

(C) acquire and process information from text files, including files of known and an unknown sizes	Chapter 25
(D) manipulate data structures using string processing	Chapter 6, Lessons 1, 2 Chapter 13, Lesson 4
(E) manipulate data values by casting between data types	Chapter 4, Lesson 3
(F) identify and use the structured data type of one- dimensional arrays to traverse, search, modify, insert, and delete data	Chapter 18 Chapter 20, Lesson 4
(G) identify and use the structured data type of two- dimensional arrays to traverse, search, modify, insert, and delete data	Chapter 21
(H) identify and use a list object data structure to traverse, search, insert, and delete data	Chapter 19

Knowledge and Skills Statement: (4) 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)
(A) develop sequential algorithms using branching control statements, including nested structures, to create solutions to decision-making problems	Chapters 8, 9 Chapter 12, Lesson 4
(B) develop choice algorithms using selection control statements based on ordinal values	Chapter 8, Lesson 4
(C) demonstrate proficiency in the use of short-circuit evaluation	Chapter 9, Lesson 2
(D) demonstrate proficiency in the use of Boolean algebra, including De Morgan's Law	Chapter 9, Lesson 3
(E) develop iterative algorithms using nested loops	Chapter 12, Lesson 4 Chapter 20, Lessons 1, 2, 3
(F) identify, trace, and appropriately use recursion in programming solutions, including algebraic computations	Chapter 24
(G) design, construct, evaluate, and compare search algorithms including linear searching and binary searching	Chapter 20, Lesson 4
(H) identify, describe, design, create, evaluate, and compare standard sorting algorithms, including selection sort, bubble sort, insertion sort, and merge sort	Chapter 20, Lessons 1, 2, 3
(I) measure time/space efficiency of various sorting algorithms	Chapter 20, Lessons 1, 2, 3 Chapter 33
(J) compare and contrast search and sort algorithms, including linear, quadratic, and recursive strategies, for time/space efficiency	Chapter 20, Lessons 1, 2, 3 Chapter 24, Lessons 2, 3 Chapter 33
(K) analyze algorithms using "big-O" notation, for best, average, and worst-case data patterns	Chapter 33
(L) develop algorithms to solve various problems, including factoring, summing a series, finding the roots of a quadratic equation, and generating Fibonacci numbers	Chapter 13, Lesson 3

(M) test program solutions by investigating boundary conditions; testing classes, methods, and libraries in isolation; and performing stepwise refinement	Chapter 11, Lesson 1 Chapter 27, Lesson 3 / Activity 3
(N) identify and debug compile, syntax, runtime, and logic errors	Chapters 10, 11
(O) compare and contrast algorithm efficiency by using informal runtime comparisons, exact calculation of statement execution counts, and theoretical efficiency values using "big-O" notation, including worst-case, best-case, and average-case time/space analysis	Chapter 20 Chapter 33
(P) demonstrate the ability to count, convert, and perform mathematical operations in the binary and hexadecimal number systems	Chapter 7, Lesson 2
(Q) demonstrate knowledge of the maximum integer boundary, minimum integer boundary, imprecision of real number representations, and round-off errors	Chapter 3, Lesson 1 Chapter 4, Lesson 3 Chapter 7, Lesson 4
(R) create program solutions to problems using the mathematics library class	Chapter 7, Lesson 3
(S) use random algorithm to create simulations that model the real world	Chapter 13 Activity Chapter 21 Activity
(T) identify, understand, and create class specifications and relationships among classes, including composition and inheritance relationships	Chapter 14, Lesson 1 Chapters 22, 23 Chapter 26, Lesson 2
(U) understand and explain object relationships among defined classes, abstract classes, and interfaces	Chapter 14, Lesson 1 Chapter 15, Lesson 5 Chapter 22, Lesson 1 Chapter 23, Lesson 2
(V) create object-oriented definitions using class declarations, variable declarations, constant declarations, method declarations, parameter declarations, and interface declarations	Chapters 14, 15
(W) create robust classes that encapsulate data and the methods that operate on that data and incorporate overloading to enrich the object's behavior	Chapters 14, 15
(X) design and implement a set of interactive classes	Chapters 14 - 17 Chapters 22, 23, 27
(Y) design, create, and evaluate multiclass programs that use abstract classes and interfaces	Chapters 15, 23
(Z) understand and implement a student-created class hierarchy	Chapter 27
(AA) extend, modify, and improve existing code using inheritance	Chapters 22, 23
(BB) create adaptive behaviors, including overloading, using polymorphism	Chapter 15, Lesson 4 Chapter 23
(CC) understand and use reference variables for object and string data types	Chapter 5, Lessons 1, 2 Chapter 14, Lesson 3
(DD) understand and implement access scope modifiers	Chapter 14, Lesson 3 Chapter 15, Lesson 3

(EE) understand and demonstrate how to compare objects	Chapter 6, Lesson 2 Chapter 8, Lesson 1 Chapter 23, Lesson 4
(FF) duplicate objects using the appropriate deep and/or shallow copy	Chapter 26, Lesson 3
(GG) define and implement abstract classes and interfaces in program problem solutions	Chapters 15, 23, 27
(HH) apply functional decomposition to a program solution	Chapter 26, Lesson 1
(II) create simple and robust objects from class definitions through instantiation	Chapters 15, 16, 17
(JJ) apply class membership of variables, constants, and methods	Chapters 14, 15
(KK) examine and mutate the properties of an object using accessors and modifiers	Chapter 15, Lesson 3
(LL) understand and implement a composite class	Chapter 26, Lesson 2
(MM) design and implement an interface	Chapter 15, Lesson 5

Knowledge and Skills Statement: (5) Digital citizenship. The student explores and understands safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to:

Student Expectation	Citation(s)
(A) model ethical acquisition and use of digital information	Chapter 1, Lessons 4, 5
(B) demonstrate proper digital etiquette, responsible use of software, and knowledge of acceptable use policies	Chapter 1, Lessons 4, 5
(C) investigate digital rights management	Chapter 1, Lessons 4, 5

Knowledge and Skills Statement: (6) Technology operations, systems, and concepts. The student understands technology concepts, systems, and operations as they apply to computer science. The student is expected to:

Student Expectation	Citation(s)
(A) compare and contrast types of operating systems, software applications, hardware platforms, and programming languages	Chapter 1, Lessons 2, 3 Chapter 2, Lesson 1
(B) demonstrate knowledge of major hardware components, including primary and secondary memory, a central processing unit (CPU), and peripherals	Chapter 1, Lesson 2
(C) demonstrate knowledge of major networking components, including hosts, servers, switches, and routers	Supplemental Chapter 4, Lesson 1
(D) demonstrate knowledge of computer communication systems, including single-user, peer-to-peer, workgroup, client-server, and networked	Supplemental Chapter 4, Lesson 2
(E) demonstrate knowledge of computer addressing systems, including Internet Protocol (IP) address and Media Access Control (MAC) address	Supplemental Chapter 4, Lesson 3
(F) differentiate among the categories of programming languages, including machine, assembly, high-level compiled, high-level interpreted, and scripted	Chapter 2, Lesson 1