Chapter 127. Texas Essential Knowledge and Skills for Career Development and Career and Technical Education

Subchapter O. Science, Technology, Engineering, and Mathematics

§127.788. Fundamentals of Computer Science (One Credit), Adopted 2021.

(a) General requirements. Students shall be awarded one credit for successful completion of this course. This course is recommended for students in Grades 7-12.

(b) Introduction.

(1) Career and technical education instruction provides content aligned with challenging academic standards, industry-relevant technical knowledge, and college and career readiness skills for students to further their education and succeed in current and emerging professions.

(2) The Science, Technology, Engineering, and Mathematics (STEM) Career Cluster focuses on planning, managing, and providing scientific research and professional and technical services such as laboratory and testing services and research and development services.

(3) Fundamentals of Computer Science is intended as a first course for those students just beginning the study of computer science. Students will learn about the computing tools that are used every day. Students will foster their creativity and innovation through opportunities to design, implement, and present solutions to real-world problems. Students will collaborate and use computer science concepts to access, analyze, and evaluate information needed to solve problems. Students will learn computational thinking, problem-solving, and reasoning skills that are the foundation of computer science. By using computer science knowledge and skills that support the work of individuals and groups in solving problems, students will select the technology appropriate for the task, synthesize knowledge, create solutions, and evaluate the results. Students will learn digital citizenship by researching current laws, regulations, and best practices and by practicing integrity and respect. Students will gain an understanding of the principles of computer science through the study of technology operations and concepts.

(4) Students are encouraged to participate in extended learning experiences such as career and technical student organizations and other leadership or extracurricular organizations.

(5) Statements that contain the word "including" reference content that must be mastered, while those containing the phrase "such as" are intended as possible illustrative examples.

(c) Knowledge and skills.

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

(A) identify job opportunities and accompanying job duties and tasks;

(B) examine the role of certifications, resumes, and portfolios in the computer science profession;

(C) employ effective technical reading and writing skills;

(D) employ effective verbal and non-verbal communication skills;

(E) solve problems and think critically;

(F) demonstrate leadership skills and function effectively as a team member;

(G) demonstrate an understanding of legal and ethical responsibilities in relation to the field of computer science;

(H) demonstrate planning and time-management skills; and

(I) compare university computer science programs.

(2) Creativity and innovation. The student develops products and generates new knowledge, understanding, and skills. The student is expected to:

(A) investigate and explore various career opportunities within the computer science field and report findings through various media;

(B) create algorithms for the solution of various problems;

(C) discuss methods and create and publish web pages using a web-based language such as HTML, Java Script, or XML; and

(D) use generally accepted design standards for spacing, fonts, and color schemes to create functional user interfaces, including static and interactive screens.

(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:

(A) seek and respond to advice or feedback from peers, educators, or professionals when evaluating problem solutions;

(B) debug and solve problems using reference materials and effective strategies; and

(C) publish information in a variety of ways such as print, monitor display, web pages, or video.

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

(A) demonstrate the ability to insert external standalone objects such as scripts or widgets into web pages;

(B) demonstrate understanding of binary representation of data in computer systems, perform conversions between decimal and binary number systems, and count in binary number systems;

(C) identify a problem's description, purpose, and goals;

(D) demonstrate coding proficiency in a programming language by developing solutions that create stories, games, and animations;

(E) identify and use the appropriate data type to properly represent the data in a program problem solution;

(F) demonstrate an understanding of and use variables within a programmed story, game, or animation;

(G) demonstrate proficiency in the use of arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division;

(H) demonstrate an understanding of and use sequence within a programmed story, game, or animation;

(I) demonstrate an understanding of and use conditional statements within a programmed story, game, or animation;

(J) demonstrate an understanding of and use iteration within a programmed story, game, or animation;

(K) use random numbers within a programmed story, game, or animation; and

(L) test program solutions by investigating intended outcomes.

(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:

(A) discuss privacy and copyright laws~~/issues~~ and model ethical acquisition of digital information by citing sources using established methods;

(B) compare various non-copyright asset sharing options such as open source, freeware, and public domain;

(C) demonstrate proper digital etiquette and knowledge of acceptable use policies when using networks;

(D) ~~discuss~~ explain the value of strong passwords, ~~and~~ virus detection~~/~~and virus~~-~~prevention for privacy and security;

(E) discuss and give examples of the impact of computing and computing-related advancements on society; and

(F) ~~discuss~~ analyze how electronic media can affect reliability of information.

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

(A) identify and explain the function ~~demonstrate knowledge~~ of ~~the~~ basic computer components, including a central processing unit (CPU), storage, and ~~input/output~~ peripheral devices;

(B) use system tools, including appropriate file management;

(C) ~~demonstrate knowledge of~~ compare different operating systems;

(D) describe the differences between an application and an operating system; and

(E) use various input, processing, output, and primary/secondary storage devices.

§127.789. Computer Science I (One Credit), Adopted 2021.

(a) General requirements. Students shall be awarded one credit for successful completion of this course. ~~Required~~ Prerequisite or corequisite: Algebra I. This course is recommended for students in Grades 8-12.

(b) Introduction.

(1) Career and technical education instruction provides content aligned with challenging academic standards, industry-relevant technical knowledge, and college and career readiness skills for students to further their education and succeed in current and emerging professions.

(2) The Science, Technology, Engineering, and Mathematics (STEM) Career Cluster focuses on planning, managing, and providing scientific research and professional and technical services such as laboratory and testing services and research and development services.

(3) Computer Science I will foster students' creativity and innovation by presenting opportunities to design, implement, and present meaningful programs through a variety of media. Students will collaborate with one another, their instructor, and various electronic communities to solve the problems presented throughout the course. Through computational thinking and data analysis, students will identify task requirements, plan search strategies, and use computer science concepts to access, analyze, and evaluate information needed to solve problems. By using computer science knowledge and skills that support the work of individuals and groups in solving problems, students will select the technology appropriate for the task, synthesize knowledge, create solutions, and evaluate the results. Students will learn digital citizenship by researching current laws, regulations, and best practices and by practicing integrity and respect. Students will gain an understanding of the principles of computer science through the study of technology operations, systems, and concepts.

(4) Students are encouraged to participate in extended learning experiences such as career and technical student organizations and other leadership or extracurricular organizations.

(5) Statements that contain the word "including" reference content that must be mastered, while those containing the phrase "such as" are intended as possible illustrative examples.

(c) Knowledge and skills.

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

(A) identify job opportunities and accompanying job duties and tasks;

(B) examine the role of certifications, resumes, and portfolios in the computer science profession;

(C) employ effective technical reading and writing skills;

(D) employ effective verbal and non-verbal communication skills;

(E) solve problems and think critically;

(F) demonstrate leadership skills and function effectively as a team member;

(G) demonstrate an understanding of legal and ethical responsibilities in relation to the field of computer science;

(H) demonstrate planning and time-management skills; and

(I) compare university computer science programs.

(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:

(A) participate in learning communities as a learner, initiator, contributor, and teacher/mentor; and

(B) seek and respond to advice from peers, educators, or professionals when evaluating quality and accuracy of the student's product.

(3) Programming style and presentation. The student utilizes proper programming style and develops appropriate visual presentation of data, input, and output. The student is expected to:

(A) create and properly label and display output;

(B) create interactive input interfaces, with relevant user prompts, to acquire data from a user such as console displays or Graphical User Interfaces (GUIs);

(C) write programs with proper programming style to enhance the readability and functionality of a code by using descriptive identifiers, internal comments, white space, spacing, indentation, and a standardized program style;

(D) format data displays using standard formatting styles; and

(E) display simple vector graphics using lines, circles, and rectangles.

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

(A) use program design problem-solving strategies such as flowchart or pseudocode to create program solutions

(B) create a high-level program plan using a visual tool such as a flow-chart or graphic organizer;

(C) identify the tasks and subtasks needed to solve a problem;

(D) identify the data types and objects needed to solve a problem;

(E) identify reusable components from existing code;

(F) design a solution to a problem;

(G) code a solution from a program design;

(H) identify error types, including syntax, lexical, run time, and logic;

(I) test program solutions with valid and invalid test data and analyze resulting behavior;

(J) debug and solve problems using error messages, reference materials, language documentation, and effective strategies;

(K) ~~explore~~ create and implement common algorithms such as finding greatest common divisor, finding the biggest number out of three, finding primes, making change, and finding the average;

(L) create program solutions that address basic error handling such as preventing division by zero and type mismatch;

(M) select the most appropriate construct for a defined problem;

(N) create program solutions by using the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division;

(O) create program solutions to problems using available mathematics library functions or operators, including absolute value, round, power, square, and square root;

(P) develop program solutions that use assignment;

(Q) develop sequential algorithms to solve non-branching and non-iterative problems;

(R) develop algorithms to decision-making problems using branching control statements;

(S) develop iterative algorithms and code programs to solve practical problems;

(T) demonstrate ~~proficiency in~~ the appropriate use of the relational operators;

(U) demonstrate ~~proficiency in~~ the appropriate use of the logical operators; and

(V) generate and use random numbers.

(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:

(A) discuss and explain intellectual property, privacy, sharing of information, copyright laws, and software licensing agreements;

(B) model ethical acquisition and use of digital information;

(C) demonstrate proper digital etiquette, responsible use of software, and knowledge of acceptable use policies;

(D) investigate *privacy and security* measures, including strong passwords, pass phrases, and other methods of authentication~~, as well as~~ and virus detection~~/~~and-virus-prevention ~~for *privacy and security*~~; and

(E) investigate computing and computing-related advancements and the social and ethical ramifications of computer usage.

(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:

(A) identify and describe the function ~~demonstrate knowledge~~ of major hardware components, including primary and secondary memory, a central processing unit (CPU), and peripherals;

(B) differentiate between current programming languages, discuss the general purpose for each language, and demonstrate knowledge of specific programming terminology and concepts and types of software development applications;

(C) differentiate between a high-level compiled language and an interpreted language;

(D) identify and use concepts of object-oriented design;

(E) differentiate between local and global scope access variable declarations;

(F) encapsulate data and associated subroutines into an abstract data type;

(G) create subroutines that do not return values with and without the use of arguments and parameters;

(H) create subroutines that return typed values with and without the use of arguments and parameters;

(I) create calls to processes passing arguments that match parameters by number, type, and position;

(J) compare data elements using logical and relational operators;

(K) identify and convert binary representation of numeric and nonnumeric data in computer systems using American Standard Code for Information Interchange (ASCII) or Unicode;

(L) identify finite limits of numeric data such as integer wrap around and floating point precision;

(M) perform numerical conversions between the decimal and binary number systems and count in the binary number system;

(N) choose, identify, and use the appropriate data types for integer, real, and Boolean data when writing program solutions;

(O) analyze the concept of a variable, including primitives and objects;

(P) represent and manipulate text data, including concatenation and other string functions;

(Q) identify and use the structured data type of one-dimensional arrays to traverse, search, and modify data;

(R) choose, identify, and use the appropriate data type or structure to properly represent the data in a program problem solution; and

(S) compare strongly typed and un-typed programming languages.

§127.790. Computer Science II (One Credit), Adopted 2021.

(a) General requirements. This course is recommended for students in Grades 10-12. Prerequisites: Algebra I and Computer Science I or AP Computer Science Principles. Students shall be awarded one credit for successful completion of this course.

(b) Introduction.

(1) Career and technical education instruction provides content aligned with challenging academic standards, industry-relevant technical knowledge, and college and career readiness skills for students to further their education and succeed in current and emerging professions.

(2) The Science, Technology, Engineering, and Mathematics (STEM) Career Cluster focuses on planning, managing, and providing scientific research and professional and technical services such as laboratory and testing services and research and development services.

(3) Computer Science II will foster students' creativity and innovation by presenting opportunities to design, implement, and present meaningful programs through a variety of media. Students will collaborate with one another, their instructor, and various electronic communities to solve the problems presented throughout the course. Through computational thinking and data analysis, students will identify task requirements, plan search strategies, and use computer science concepts to access, analyze, and evaluate information needed to solve problems. By using computer science knowledge and skills that support the work of individuals and groups in solving problems,

students will select the technology appropriate for the task, synthesize knowledge, create solutions, and evaluate the results. Students will gain an understanding of computer science through the study of technology operations, systems, and concepts.

(4) Students are encouraged to participate in extended learning experiences such as career and technical student organizations and other leadership or extracurricular organizations.

(5) Statements that contain the word "including" reference content that must be mastered, while those containing the phrase "such as" are intended as possible illustrative examples.

(c) Knowledge and skills.

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

(A) identify job opportunities and accompanying job duties and tasks;

(B) examine the role of certifications, resumes, and portfolios in the computer science profession;

(C) employ effective technical reading and writing skills;

(D) employ effective verbal and non-verbal communication skills;

(E) solve problems and think critically;

(F) demonstrate leadership skills and function effectively as a team member;

(G) identify legal and ethical responsibilities in relation to the field of computer science;

(H) demonstrate planning and time-management skills; and

(I) compare university computer science programs.

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

(A) use program design problem-solving strategies to create program solutions;

(B) read, analyze, and modify programs and their accompanying documentation such as an application programming interface (API), internal code comments, external documentation, or readme files;

(C) follow a systematic problem-solving process ~~of~~ which is identifying the purpose and goals, the data types and objects needed, and the subtasks to be performed;

(D) compare design methodologies and implementation techniques such as top-down, bottom-up, and black box;

(E) trace a program, including inheritance and black box programming;

(F) 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; and

(G) use object-oriented programming development methodology~~,~~ including data abstraction, encapsulation with information hiding, inheritance, and procedural abstraction in program development ~~and testing~~.

(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:

(A) use the principles of software development to work in software design teams;

(B) break a problem statement into specific solution requirements;

(C) create a program development plan;

(D) code part of a solution from a program development plan while a partner codes the remaining part;

(E) collaborate with a team to test a solution, including boundary and standard cases; and

(F) develop presentations to report the solution findings.

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

(A) utilize programming file structure and file access for required resources;

(B) acquire and process information from text files, including files of known and unknown sizes;

(C) manipulate data using string processing;

(D) manipulate data values by casting between data types;

(E) use the structured data type of one-dimensional arrays to traverse, search, modify, insert, and delete data;

(F) identify and use the structured data type of two-dimensional arrays to traverse, search, modify, insert, and delete data;

(G) identify and use a list object data structure to traverse, search, insert, and delete data; and

(H) differentiate between categories of programming languages, including machine, assembly, high-level compiled, high-level interpreted, and scripted.

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

(A) develop sequential algorithms using branching control statements, including nested structures, to create solutions to decision-making problems;

(B) develop choice algorithms using selection control statements based on ordinal values;

(C) demonstrate ~~proficiency in~~ the appropriate use of situations involving short-circuit evaluation;

(D) ~~demonstrate proficiency in the~~ use ~~of~~ Boolean algebra, including De Morgan's Law to evaluate and simplify logical expressions;

(E) develop iterative algorithms using nested loops;

(F) identify, trace, and appropriately use recursion in programming solutions, including algebraic computations;

(G) trace, construct, evaluate, and compare search algorithms, including linear searching and binary searching;

(H) identify, describe, trace, evaluate, and compare standard sorting algorithms, including selection sort, bubble sort, insertion sort, and merge sort;

(I) measure time/space efficiency of various sorting algorithms, including analyzing algorithms using "big-O" notation for best, average, and worst-case data patterns;

(J) develop algorithms to solve various problems such as factoring, summing a series, finding the roots of a quadratic equation, and generating Fibonacci numbers;

(K) test program solutions by investigating boundary conditions; testing classes, methods, and libraries in isolation; and performing stepwise refinement;

(L) identify and debug compile, syntax, runtime, and logic errors;

(M) compare ~~algorithm~~ efficiency of *search and sort algorithms* ~~linear, quadratic, and recursive strategies~~ 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 ~~of~~ ~~*search and sort algorithms*~~;

(N) ~~demonstrate the ability to~~ count, convert, and perform mathematical operations in the decimal, binary, octal, and hexadecimal number systems;

(O) ~~demonstrate knowledge of the~~ identify maximum integer boundary, minimum integer boundary, imprecision of real number representations, and round-off errors;

(P) create program solutions to problems using a mathematics library;

(Q) use random number generator algorithms to create simulations;

(R) use composition and inheritance relationships to identify and create class definitions and relationships;

(S) explain and use object relationships between defined classes, abstract classes, and interfaces;

(T) create object-oriented class definitions and declarations using variables, constants, methods, parameters, and interface implementations;

(U) create adaptive behaviors using polymorphism;

(V) use reference variables for object and string data types;

(W) use value and reference parameters appropriately in method definitions and method calls;

(X) implement access scope modifiers;

(Y) use object comparison for content quality;

(Z) duplicate objects using the appropriate deep or shallow copy;

(AA) apply functional decomposition to a program solution;

(BB) create objects from class definitions through instantiation; and

(CC) examine and mutate the properties of an object using accessors and modifiers.

§127.791. Computer Science III (One Credit), Adopted 2021.

(a) General requirements. This course is recommended for students in Grades 11 and 12. Prerequisite: Computer Science II, Advanced Placement (AP) Computer Science A, or International Baccalaureate (IB) Standard Level or Higher Level Computer Science. Students shall be awarded one credit for successful completion of this course.

(b) Introduction.

(1) Career and technical education instruction provides content aligned with challenging academic standards, industry-relevant technical knowledge, and college and career readiness skills for students to further their education and succeed in current and emerging professions.

(2) The Science, Technology, Engineering, and Mathematics (STEM) Career Cluster focuses on planning, managing, and providing scientific research and professional and technical services such as laboratory and testing services and research and development services.

(3) Computer Science III will foster students' creativity and innovation by presenting opportunities to design, implement, and present meaningful programs through a variety of media. Students will collaborate with one another, their instructor, and various electronic communities to solve the problems presented throughout the course. Through computational thinking and data analysis,

students will identify task requirements, plan search strategies, and use computer science concepts to access, analyze, and evaluate information needed to solve problems. By using computer science knowledge and skills that support the work of individuals and groups in solving problems, students will select the technology appropriate for the task, synthesize knowledge, create solutions, and evaluate the results. Students will gain an understanding of advanced computer science data structures through the study of technology operations, systems, and concepts.

(4) Statements that contain the word "including" reference content that must be mastered, while those containing the phrase "such as" are intended as possible illustrative examples.

(5) Students are encouraged to participate in extended learning experiences such as career and technical student organizations and other leadership or extracurricular organizations.

(c) Knowledge and skills.

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

(A) identify job opportunities and accompanying job duties and tasks;

(B) examine the role of certifications, resumes, and portfolios in the computer science profession;

(C) employ effective technical reading and writing skills;

(D) employ effective verbal and non-verbal communication skills;

(E) solve problems and think critically;

(F) demonstrate leadership skills and function effectively as a team member;

(G) demonstrate an understanding of legal and ethical responsibilities in relation to the field of computer science;

(H) demonstrate planning and time-management skills; and

(I) compare university computer science programs.

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

(A) apply object-oriented programming, including data abstraction, encapsulation, inheritance, and polymorphism, to manage the complexity of a project;

(B) design and implement a class hierarchy;

(C) read and write class specifications using visual organizers, including Unified Modeling Language;

(D) identify, describe, evaluate, compare, and implement standard sorting algorithms that perform sorting operations on data structures, including quick sort and heap sort; and

(E) identify and use the appropriate abstract data type, advanced data structure, and supporting algorithms to properly represent the data in a program problem solution.

(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:

(A) use networked tools for file management and collaboration; and

(B) work in software design teams.

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

(A) identify and use two-dimensional ragged arrays to traverse, search, modify, insert, and delete data;

(B) describe and demonstrate proper linked list management, including maintaining the head and safe addition and deletion of linked objects;

(C) create or trace program solutions using a linked-list data structure, including unordered single, ordered single, double, and circular linked;

(D) describe composite data structures, including a linked list of linked lists;

(E) create or trace program solutions using stacks, queues, trees, heaps, priority queues, graph theory, and enumerated data types;

(F) create or trace program solutions using sets, including hash and tree-based data structures;

(G) create or trace program solutions using map style data structures; and

(H) write and modify text file data.

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

(A) evaluate expressions using bitwise operators;

(B) evaluate expressions using the ternary operator;

(C) identify, trace, and appropriately use recursion in programming solutions, including processing binary trees;

(D) create or trace program solutions using hashing;

(E) explore common algorithms such as matrix addition and multiplication, fractals, Towers of Hanoi, and magic square; and

(F) create program solutions that exhibit robust behavior by recognizing and avoiding runtime errors and handling anticipated errors.

(6) Testing and documentation. The student demonstrates appropriate documentation and testing practices. The student is expected to:

(A) use appropriate formatting and write documentation to support code maintenance, including pre- and post-condition statements;

(B) write program assumptions in the form of assertions;

(C) write a Boolean expression to test a program assertion; and

(D) construct assertions to make explicit program invariants.

(7) Practical application of technology. The student utilizes technology concepts, systems, and operations as they apply to computer science. The student is expected to:

(A) analyze and create computer program workflow charts and basic system diagrams, documenting system functions, features, and operations;

(B) gather requirements, design, and implement a process by which programs can interact with each other such as using interfaces;

(C) create simple programs using a low-level language such as assembly;

(D) create discovery programs in a high-level language;

(E) create scripts for an operating system;

(F) explore industry best practices for secure programming; and

(G) explore emerging industry or technology trends.