Texas CTE: Science, Technology, Engineering, and Mathematics

Computer Science I (2022)

Implementation. The provisions of this section shall be implemented by school districts beginning with the 2024-2025 school year. A

1 **No later than August 1, 2024, the commissioner of education shall determine whether instructional materials funding has been made available to Texas public schools for materials that cover the essential knowledge and skills identified in this section.** A.1

2 **If the commissioner makes the determination that instructional materials funding has been made available this section shall be implemented beginning with the 2024-2025 school year and apply to the 2024-2025 and subsequent school years.** A.2

3 **If the commissioner does not make the determination that instructional materials funding has been made available under subsection (a) of this section, the commissioner shall determine no later than August 1 of each subsequent school year whether instructional materials funding has been made available. If the commissioner determines that instructional materials funding has been made available, the commissioner shall notify the State Board of Education and school districts that this section shall be implemented for the following school year.** A.3

General requirements. This course is recommended for students in Grades 9-12. Prerequisite or corequisite: Algebra I. Students shall be awarded one credit for successful completion of this course. B

b **General requirements. This course is recommended for students in Grades 9-12. Prerequisite or corequisite: Algebra I. Students shall be awarded one credit for successful completion of this course.** B

Introduction. C

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.** C.1

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.** C.2

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.** C.3

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

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.5

Knowledge and skills. D

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

A identify job and internship opportunities and accompanying job duties and tasks and contact one or more companies or organizations to explore career opportunities; D.1.A

B examine the role of certifications, resumes, and portfolios in the computer science profession; D.1.B

C employ effective technical reading and writing skills; D.1.C

D employ effective verbal and non-verbal communication skills; D.1.D

E solve problems and think critically; D.1.E

F demonstrate leadership skills and function effectively as a team member; D.1.F

G communicate an understanding of legal and ethical responsibilities in relation to the field of computer science; D.1.G

H demonstrate planning and time-management skills; and D.1.H

I compare university computer science programs. D.1.I

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:** D.2

- A participate in learning communities as a learner, initiator, contributor, and teacher/mentor; and D.2.A
- B seek and respond to advice from peers, educators, or professionals when evaluating quality and accuracy of the student's product. D.2.B

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:** D.3

- A create and properly label and display output; D.3.A
- B create interactive input interfaces, with relevant user prompts, to acquire data from a user such as console displays or Graphical User Interfaces (GUIs); D.3.B
- 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.3.C
- D format data displays using standard formatting styles; and D.3.D
- E display simple vector graphics using lines, circles, and rectangles. D.3.E

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

A use program design problem-solving strategies such as flowchart or pseudocode to create program solutions; D.4.A

B create a high-level program plan using a visual tool such as a flowchart or graphic organizer; D.4.B

C identify the tasks and subtasks needed to solve a problem; D.4.C

D identify the data types and objects needed to solve a problem; D.4.D

E identify reusable components from existing code; D.4.E

F design a solution to a problem; D.4.F

G code a solution from a program design; D.4.G

H identify error types, including syntax, lexical, run time, and logic; D.4.H

I test program solutions with valid and invalid test data and analyze resulting behavior; D.4.I

J debug and solve problems using error messages, reference materials, language documentation, and effective strategies; D.4.J

K 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; D.4.K

L create program solutions that address basic error handling such as preventing division by zero and type mismatch; D.4.L

M select the most appropriate construct for a defined problem; D.4.M

N create program solutions by using the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division; D.4.N

O create program solutions to problems using available mathematics library functions or operators, including absolute value, round, power, square, and square root; D.4.O

P develop program solutions that use assignment; D.4.P

Q develop sequential algorithms to solve non-branching and non-iterative problems; D.4.Q

R develop algorithms to decision-making problems using branching control statements; D.4.R

S develop iterative algorithms and code programs to solve practical problems; D.4.S

T demonstrate the appropriate use of the relational operators; D.4.T

U demonstrate the appropriate use of the logical operators; and D.4.U

V generate and use random numbers. D.4.V

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:** D.5

A discuss and explain intellectual property, privacy, sharing of information, copyright laws, and software licensing agreements; D.5.A

B practice ethical acquisition and use of digital information; D.5.B

C demonstrate proper digital etiquette, responsible use of software, and knowledge of acceptable use policies; D.5.C

D investigate privacy and security measures, including strong passwords, pass phrases, and other methods of authentication and virus detection and prevention; and D.5.D

E investigate computing and computing-related advancements and the social and ethical ramifications of computer usage. D.5.E

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:** D.6

A identify and describe the function of major hardware components, including primary and secondary memory, a central processing unit (CPU), and peripherals; D.6.A

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; D.6.B

C differentiate between a high-level compiled language and an interpreted language; D.6.C

D identify and use concepts of object-oriented design; D.6.D

E differentiate between local and global scope access variable declarations; D.6.E

F encapsulate data and associated subroutines into an abstract data type; D.6.F

G create subroutines that do not return values with and without the use of arguments and parameters; D.6.G

H create subroutines that return typed values with and without the use of arguments and parameters; D.6.H

I create calls to processes passing arguments that match parameters by number, type, and position; D.6.I

J compare data elements using logical and relational operators; D.6.J

K identify and convert binary representation of numeric and nonnumeric data in computer systems using American Standard Code for Information Interchange (ASCII) or Unicode; D.6.K

L identify finite limits of numeric data such as integer wrap around and floating point precision; D.6.L

M perform numerical conversions between the decimal and binary number systems and count in the binary number system; D.6.M

N choose, identify, and use the appropriate data types for integer, real, and Boolean data when writing program solutions; D.6.N

O analyze the concept of a variable, including primitives and objects; D.6.O

P represent and manipulate text data, including concatenation and other string functions; D.6.P

Q identify and use the structured data type of one-dimensional arrays to traverse, search, and modify data; D.6.Q

R choose, identify, and use the appropriate data type or structure to properly represent the data in a program problem solution; and D.6.R

S compare strongly typed and un-typed programming languages. D.6.S