§126.35. Computer Science III TEKS

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

(b) Introduction.

(1) The technology applications curriculum has six strands based on the National Educational Technology Standards for Students (NETS•S) and performance indicators developed by the International Society for Technology in Education (ISTE): creativity and innovation; communication and collaboration; research and information fluency; critical thinking, problem solving, and decision making; digital citizenship; and technology operations and concepts.

(2) 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 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 and regulations and by practicing integrity and respect. Students will gain an understanding of advanced computer science data structures through the study of technology operations, systems, and concepts.

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

(A) apply data abstraction and encapsulation to manage complexity;

(B) implement a student-created class hierarchy;

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

(D) use black box programming methodology;

(E) design, create, and use interfaces to apply protocols;

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

(G) select, 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

(H) manage complexity by using a systems approach.

(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) use local area networks (LANs) and wide area networks (WANs), including the Internet and intranets, in research, file management, and collaboration;

(B) create interactive human interfaces to acquire data from a user and display program results using an advanced Graphical User Interface (GUI);

(C) 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; and

(D) work in software design teams.

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

(A) identify and use the structured data type of arrays of objects to traverse, search, modify, insert, and delete data;

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

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

(D) understand and trace a linked-list data structure;

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

(F) understand composite data structures, including a linked list of linked lists;

(G) understand and create program solutions using stacks, queues, trees, heaps, priority queues, graph theory, and enumerated data types;

(H) understand and create program solutions using sets, including HashSet and TreeSet;

(I) understand and create program solutions using maps, including HashMap and TreeMap; and

(J) write and modify text file data.

(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) develop choice algorithms using selection control statements, including break, label, and continue;

(B) demonstrate proficiency in the use of the bitwise operators;

(C) develop iterative algorithms using do-while loops;

(D) demonstrate proficiency in the use of the ternary operator;

(E) create program solutions that use iterators;

(F) identify, trace, and appropriately use recursion;

(G) understand and create program solutions using hashing;

(H) perform pattern recognition using regular expressions;

(I) explore common algorithms, including matrix addition and multiplication, fractals, Towers of Hanoi, and magic square;

(J) create program solutions that exhibit robust behavior by understanding and avoiding runtime errors and handling anticipated errors;

(K) understand object-oriented design concepts of inner classes, outer classes, and anonymous classes;

(L) use object reference scope identifiers, including null, this, and super;

(M) provide object functionality to primitive data types;

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

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

(P) construct assertions to make explicit program invariants.

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

(B) demonstrate proper digital etiquette, responsible use of software, and knowledge of acceptable use policies.

(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) compare and contrast high-level programming languages;

(B) create a small workgroup network;

(C) create and apply a basic network addressing scheme; and

(D) create discovery programs in a low-level language, high-level language, and scripting language.