

# **Proposed Revisions**

## ***Texas Essential Knowledge and Skills***

### ***Technology Applications, High School Computer Science***

**Prepared by the State Board of Education TEKS Review Committees**

**November 2010**

These documents have been formatted for consistency and ease of review.

Proposed additions are shown in green font with underlines and proposed deletions are shown in red font with strikethroughs.

Comments in the margin provide explanations for proposed changes. The following notations were used as part of the explanations:

**CRS**—information added or changed to align with College Readiness Standards

**ER**—information added, changed, or deleted based on expert reviewer feedback

**MV**—multiple viewpoints from within the committee

**VA**—information added, changed, or deleted to increase vertical alignment

**21st**—information updated to 21<sup>st</sup> century technology trends, applications, and uses

#### **Table of Contents**

Fundamentals of Computer Science .....	1
Computer Science I .....	4
Computer Science II .....	11
Computer Science III .....	18
Digital Forensics .....	21
Discrete Mathematics .....	24
Game Programming and Design .....	27
Mobile Application Development .....	31
Robotics Programming and Design .....	34

**§126.xx. Fundamentals of Computer Science (One-half to One Credit).**

(a) General requirements. The prerequisite for this course is proficiency in the knowledge and skills relating to Technology Applications Grades 6-8. This course is recommended for students in Grades 9-12.

(b) Introduction.

(1) The technology applications curriculum has six strands based on the National Educational Technology Standards (NETS•S) and Performance Indicators for Students 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.

Comment [A1]: VA

(2) Fundamentals of Computer Science is intended as a first course for those students just beginning study of computer science. The focus of this course is to teach about the computing tools that are used every day. This course will provide students that continue their studies in computer science with the problem solving and reasoning skills that are the foundation of computer science. Fundamentals of Computer Science will foster student creativity and innovation by presenting 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. 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 a solution, 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 the principles of computer science through the study of technology operations and concepts.

Comment [A2]: VA

(c) Knowledge and skills.

(1) Creativity and innovation. The student will develop products and generate new understanding by extending existing knowledge. The student is expected to:

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

(B) create and publish interactive stories;

(C) create and publish interactive games;

(D) create and publish interactive animations;

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

(F) create web pages using a mark up language;

(G) use the internet to create and publish solutions; and

(H) design creative and effective user interfaces.

- (2) Communication and collaboration. The student will communicate and collaborate with their peers to contribute to the learning of themselves and others. The student is expected to:
  - (A) seek and respond to advice from peers and professionals in 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, web pages, and video.
- (3) Research and information fluency. The student will locate, analyze, process, and organize data. The student is expected to:
  - (A) construct appropriate electronic search strategies; and
  - (B) use a variety of resources, including other subject areas, together with various productivity tools to gather authentic data as a basis for individual and group programming projects.
- (4) Critical thinking, problem solving, and decision making. The student will use appropriate strategies to analyze problems and design algorithms. The student is expected to:
  - (A) demonstrate the ability to insert applets into web pages;
  - (B) find, download, and insert scripting code into web pages to enhance interactivity;
  - (C) understand binary representation of data in computer systems, perform conversions between decimal and binary number systems, and count in binary number systems;
  - (D) read and define a problem's description, purpose, and goals;
  - (E) demonstrate coding proficiency in a contemporary programming language by developing solutions that create stories, games, and animations;
  - (F) choose, identify, and use the appropriate data type to properly represent data in a problem solution;
  - (G) demonstrate an understanding of and use variables within a programmed story, game, or animation;
  - (H) demonstrate proficiency in the use of arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division;
  - (I) demonstrate an understanding of and use sequence within a programmed story, game, or animation;
  - (J) demonstrate an understanding of and use conditional statements within a programmed story, game, or animation;
  - (K) demonstrate an understanding of and use iteration within a programmed story, game, or animation;
  - (L) create an interactive story, game, or animation;

Comment [A3]: ER

Comment [A4]: VA

Comment [A5]: MV - To clarify terminology

- (M) use random numbers within a programmed story, game, or animation; and
- (N) test program solutions by investigating valid and invalid data.
- (5) Digital citizenship. The student will explore and understand safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to:
  - (A) discuss copyright laws/issues and model ethical acquisition of digital information, citing sources, and using established methods;
  - (B) demonstrate proper digital etiquette and knowledge of acceptable use policies when using networks, especially resources on the Internet and on intranets;
  - (C) investigate measures such as passwords or virus detection/prevention to protect computer systems and databases from unauthorized use and tampering;
  - (D) understand the safety risks associated with the use of social networking sites;
  - (E) discuss the impact of computing and computing related advancements on society; and
  - (F) determine the reliability of information available through electronic media.
- (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) demonstrate knowledge of the basic computer components, including the central processing unit (CPU), storage, and input/output;
  - (B) use operating system tools, including appropriate file management;
  - (C) demonstrate knowledge and appropriate use of different operating systems;
  - (D) demonstrate knowledge and understanding of basic network connectivity;
  - (E) describe, compare, and contrast the differences between an application and an operating system; and
  - (F) compare, contrast, and appropriately use various input, processing, output, and primary/secondary storage devices.

Comment [A6]: VA – covered in CS1

Comment [A7]: 21<sup>st</sup> - skills

Comment [A8]: 21<sup>st</sup> - skills

Comment [A9]: 21<sup>st</sup> - skills

Comment [A10]: 21<sup>st</sup> - skills

**§126.22. Computer Science I (One-Half to One Credit).**

(a) General requirements. ~~The prerequisite for this course is proficiency in the knowledge and skills described in §126.12(e) of this title (relating to Technology Applications (Computer Literacy), Grades 6-8). In addition, it is recommended that students have proficiency in the knowledge and skills for Algebra I identified in §111.32(b) of this title (relating to Algebra I (One Credit)) or the equivalent knowledge and skills. This course is recommended for students in Grades 9-12. School districts may use the knowledge and skills described in subsection (e) of this section, the computer science course descriptions for the College Board Advanced Placement or International Baccalaureate programs, or a combination thereof.~~

**Comment [A11]:** ER – recommended by reviewers

(b) Introduction.

**Comment [A12]:** AP and IB Courses have been moved and are offered under section D of these TEKS.

(1) ~~The technology applications curriculum has four six strands: foundations, information acquisition, work in solving problems, and communication based on the National Educational Technology Standards (NETS•S) and Performance Indicators for Students 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.~~

**Comment [A13]:** ER comments

~~(2) Through the study of technology applications foundations, including technology-related terms, concepts, and data input strategies, students learn to make informed decisions about technologies and their applications. The efficient acquisition of information includes the identification of task requirements; the plan for using search strategies; and the use of technology to access, analyze, and evaluate the acquired information. By using technology as a tool that supports the work of individuals and groups in solving problems, students will select the technology appropriate for the task, synthesize knowledge, create a solution, and evaluate the results. Students communicate information in different formats and to diverse audiences. A variety of technologies will be used. Students will analyze and evaluate the results.~~

**Comment [A14]:** Standardization

(2) Computer Science I is designed to 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. Data analysis will include the identification of task requirements, planning search strategies, and the use of 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 to become good digital citizens by practicing integrity and respect throughout the course. Students will gain an understanding of the principles of computer science through the study of technology operations, systems, and concepts.

(c) Knowledge and skills.

~~(1)(7)~~ Creativity and innovation Solving problems. The student ~~uses appropriate computer-based productivity tools to create and modify solutions to problems~~ develops products and generates new understanding by extending existing knowledge. The student is expected to:

~~(A) apply problem solving strategies such as design specifications, modular top-down design, step-wise refinement, or algorithm development;~~

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

~~(B) use visual organizers to design solutions such as flowcharts or schematic drawings;~~

(B) extend the learning environment beyond the school walls with digital products created to increase teaching and learning in the other subject areas; and

~~(C) develop sequential and iterative algorithms and codes program in prevailing computer languages to solve practical problems modeled from school and community;~~

(C) participate in relevant, meaningful activities in the larger community and society to create electronic projects.

~~(7)(D) code using various data types;~~

~~(7)(E) demonstrate effective use of predefined input and output procedures for lists of computer instructions including procedures to protect from invalid input;~~

~~(7)(G) create and use libraries of generic modular code to be used for efficient programming;~~

~~(7)(H) identify actual and formal parameters and use value and reference parameters;~~

~~(7)(I) use control structures such as conditional statements and iterated, pretest, and posttest loops;~~

~~(7)(J) use sequential, conditional, selection, and repetition execution control structures such as menu-driven programs that branch and allow user input; and~~

~~(7)(K) identify and use structured data types of one-dimensional arrays, records, and text files.~~

~~(2) Foundations.~~ The student uses data input skills appropriate to the task. The student is expected to:

~~(A) demonstrate proficiency in the use of a variety of input devices such as keyboard, scanner, voice/sound recorder, mouse, touch screen, or digital video by appropriately incorporating such components into the product; and~~

~~(B) use digital keyboarding standards for the input of data.~~

**Comment [A15]:** All strands have been reorganized to provide vertical alignment

**Comment [A16]:** standardization

**Comment [A17]:** based on feedback

**Comment [A18]:** based on feedback

**Comment [A19]:** based on feedback

**Comment [A20]:** moved

**Comment [A21]:** moved

**Comment [A22]:** moved

(2)(4) Communication and collaboration ~~Information acquisition.~~ The student communicates and collaborates with their peers to contribute to the learning of themselves and others ~~uses a variety of strategies to acquire information from electronic resources, with appropriate supervision.~~ The student is expected to:

(A) create and properly display meaningful output; use local area networks (LANs) and wide area networks (WANs), including the Internet and intranet, in research and resource sharing; and

(B) create interactive console display interfaces, with appropriate user prompts, to acquire data from a user; construct appropriate electronic search strategies in the acquisition of information including keyword and Boolean search strategies.

(C) use Graphical User Interfaces (GUIs) to create interactive interfaces to acquire data from a user and display program results;

(D)(9)(B) write programs with proper use correct programming style to enhance the readability and functionality of the code by using meaningful descriptive identifiers, internal comments, white space, spacing, indentation, and a standardized program style such as spacing, descriptive identifiers, comments, or documentation;

(E) improve numeric display by optimizing data visualization;

(F) display simple vector graphics using lines, circles and rectangles;

(G) display simple bit map images;

(H) seek and respond to advice from peers and professionals in evaluating quality and accuracy.

(3) Research and information fluency. The student will locate, analyze, process, and organize data. The student is expected to:

(A)(5)(B) use a variety of resources, including foundation and enrichment curricula, together with various productivity tools to gather authentic data as a basis for individual and group programming projects; and

(B) use various productivity tools to gather authentic data as a basis for individual and group programming projects.

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

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

(B) define and specify the purpose and goals for a problem;

(C) identify the 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;

Comment [A23]: Standardization

Comment [A24]: based on feedback

Comment [A25]: 21<sup>st</sup> - Rationale – GUIs are ubiquitous. To create a GUI is to solve a problem. GUIs are an essential 21<sup>st</sup> century skill for students to understand and create interfaces for humans to interact with computers.

Comment [A26]: ER

Comment [A27]: based on feedback

Comment [A28]: MV

Comment [A29]: VA; Consistency

Comment [A30]: MV - added

Comment [A31]: VA Standardization

Comment [A32]: MV - feedback

- (G) code a solution from a program design;
- (H) identify and debug errors;
- (I) test program solutions with appropriate valid and invalid test data for correctness;
- (J)(H)(C) debug and solve problems using error messages, reference materials, language documentation, and effective strategies;
- (K) explore common algorithms, including greatest common divisor, finding the biggest number out of three, finding primes, making change, and finding the average;
- (L) analyze and modify existing code to improve the underlying algorithm;
- (M) create program solutions that exhibit robust behavior by understanding, avoiding, and preventing runtime errors, including division by zero and type mismatch;
- (N) select the most appropriate algorithm for a defined problem;
- (O) demonstrate proficiency in the use of the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division;
- (P) create program solutions to problems using available mathematics libraries, including absolute value, round, power, square, and square root
- (Q)(7)(F) develop program solutions that use coding with correct and efficient use of expressions and assignment statements including the use of standard/user-defined functions, data structures, operators/proper operator precedence, and sequential/conditional/repetitive control structures;
- (R) develop sequential algorithms to solve non-branching and non-iterative problems;
- (S) develop algorithms to decision-making problems using branching control statements;
- (T)(7)(C) develop sequential and iterative algorithms and codes programs in prevailing computer languages to solve practical problems modeled from school and community;
- (U) demonstrate proficiency in the use of the relational operators;
- (V) demonstrate proficiency in the use of the logical operators; and
- (W) generate and use random numbers.

Comment [A33]: VA – cross curriculum, response to comments

Comment [A34]: MV – response to feedback

Comment [A35]: MV – response to feedback

Comment [A36]: based on feedback

Comment [A37]: based on feedback

Comment [A38]: based on feedback

Comment [A39]: based on feedback

Comment [A40]: based on feedback

Comment [A41]: based on feedback

Comment [A42]: based on feedback

(5) Information acquisition. The student acquires electronic information in a variety of formats, with appropriate supervision. The student is expected to:

- (A) acquire information in and knowledge about electronic formats including text, audio, video, and graphics;



~~(C) design and document sequential search algorithms for digital information storage and retrieval.~~

~~(5)(3)~~ Digital citizenship Foundations. The student ~~complies with the laws and examines the explores and understands safety, legal, cultural, and societal~~ issues ~~relating to regarding~~ the use of technology ~~in society and information~~. The student is expected to:

(A) discuss intellectual property, privacy, and sharing of information, copyright laws, ~~issues~~ and software licensing agreements ~~model ethical acquisition and use of digital information, citing sources using established methods;~~

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

~~(C)(B)~~ demonstrate proper digital etiquette, responsible use of software, and knowledge of acceptable use policies ~~when using networks, especially resources on the Internet and intranet;~~

~~(D) discuss the impact of computer programming on the World Wide Web (WWW) community.~~

~~(D)(C)~~ investigate measures, ~~such as including~~ passwords ~~or and~~ virus detection/prevention, to protect computer systems and databases from unauthorized use and tampering; and

(E) investigate how technology has changed and the social and ethical ramifications of computer usage.

~~(6) Information acquisition. The student evaluates the acquired electronic information. The student is expected to:~~

~~(A) determine and employ methods to evaluate the design and functionality of the process using effective coding, design, and test data; and~~

~~(B) implement methods for the evaluation of the information using defined rubrics.~~

~~(6)(4)~~ Technology operations, systems, and concepts Foundations. The student understands technology concepts, systems, and operations as they apply to computer science demonstrates knowledge and appropriate use of hardware components, software programs, and their connections. The student is expected to:

(A) compare and contrast types ~~demonstrate knowledge and appropriate use of~~ operating systems, software applications, and ~~communication and networking components~~ programming languages;

~~(B) compare, contrast, and appropriately use the various input, processing, output, and primary/secondary storage devices;~~

(B) demonstrate knowledge of major hardware components, including primary and secondary memory, the central processing unit, and peripherals;

~~(C) make decisions regarding the selection, acquisition, and use of software taking under consideration its quality, appropriateness, effectiveness, and efficiency;~~

~~(C)(E)~~ differentiate among current programming languages, discuss the use of those languages in other fields of study, and demonstrate knowledge of specific programming terminology and concepts;

Comment [A43]: VA Consistency and standardization

Comment [A44]: VA Consistency

~~(D) delineate and make necessary adjustments regarding compatibility issues including, but not limited to, digital file formats and cross-platform connectivity;~~

~~(D)(F)~~ differentiate between a among the levels of programming languages including machine, assembly; high-level compiled language and an interpreted languages; and

~~(G) demonstrate coding proficiency in a contemporary programming language.~~

(E) understand concepts of object-oriented design ;

Comment [A45]: based on feedback

(F) use local and global scope access variable declarations;

Comment [A46]: moved to CS2

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

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

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

(J) understand and identify the data-binding process between arguments and parameters;

(K) compare objects using reference values and a comparison routine;

Comment [A47]: based on feedback

(L) understand the binary representation of numeric and nonnumeric data in computer systems;

(M) understand the finite limits of numeric data;

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

Comment [A48]: based on feedback

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

Comment [A49]: based on feedback

(P) understand the concept of a variable.

Comment [A50]: MV - needed

(Q) understand and use reference variables for objects;

Comment [A51]: MV – found in multiple programming languages

(R) understand how to represent and manipulate text data, including concatenation and other string functions;

(S) demonstrate an understanding of the concept of scope;

Comment [A52]: based on feedback

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

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

(V) compare and contrast strongly typed and un-typed programming languages.

Comment [A53]: based on feedback; VA Moved from (2) for consistency

~~(8) Solving problems. The student uses research skills and electronic communication, with appropriate supervision, to create new knowledge. The student is expected to:~~

~~(A) participate with electronic communities as a learner, initiator, contributor, and teacher/mentor;~~

- ~~(B) demonstrate proficiency in, appropriate use of, and navigation of LANs and WANs for research and for sharing of resources;~~
- ~~(C) extend the learning environment beyond the school walls with digital products created to increase teaching and learning in the foundation and enrichment curricula; and~~
- ~~(D) participate in relevant, meaningful activities in the larger community and society to create electronic projects.~~
- ~~(9) Solving problems. The student uses technology applications to facilitate evaluation of work, both process and product. The student is expected to:~~
  - ~~(A) design and implement procedures to track trends, set timelines, and review/evaluate progress for continual improvement in process and product;~~
  - ~~(C) seek and respond to advice from peers and professionals in delineating technological tasks;~~
  - ~~(D) resolve information conflicts and validate information through accessing, researching, and comparing data; and~~
  - ~~(E) create technology specifications for tasks/evaluation rubrics and demonstrate that products/product quality can be evaluated against established criteria.~~
- ~~(10) Communication. The student formats digital information for appropriate and effective communication. The student is expected to:~~
  - ~~(A) annotate coding properly with comments, indentation, and formatting; and~~
  - ~~(B) create interactive documents using modeling, simulation, and hypertext.~~
- ~~(11) Communication. The student delivers the product electronically in a variety of media, with appropriate supervision. The student is expected to:~~
  - ~~(A) publish information in a variety of ways including, but not limited to, printed copy and monitor displays; and~~
  - ~~(B) publish information in a variety of ways including, but not limited to, software, Internet documents, and video.~~
- ~~(12) Communication. The student uses technology applications to facilitate evaluation of communication, both process and product. The student is expected to:~~
  - ~~(A) write technology specifications for planning/evaluation rubrics documenting variables, prompts, and programming code internally and externally;~~
  - ~~(B) seek and respond to advice from peers and professionals in evaluating the product; and~~

**§126.23. Computer Science II (One Credit).**

(a) General requirements. The prerequisite for this course is proficiency in the knowledge and skills for Algebra I identified in §111.32(b) of this title (relating to Algebra I (One half to One Credit)) and either Computer Science I as identified in §126.22(c) of this title (relating to Computer Science I (One Credit)) or Fundamentals of Computer Science as identified in §126.22(c) of this title (relating to Fundamentals of Computer Science (One half to One Credit)). This course is recommended for students in Grades ~~10~~11-12. ~~School districts may use the knowledge and skills described in subsection (c) of this section, the computer science course descriptions for the College Board Advanced Placement or International Baccalaureate programs, or a combination thereof.~~

Comment [A54]: ER – recommended by reviewers

(b) Introduction.

(1) The technology applications curriculum has ~~four~~ six strands: ~~foundations, information acquisition, work in solving problems, and communication based on the National Educational Technology Standards (NETS•S) and Performance Indicators for Students 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.~~

Comment [A55]: VA

~~(2) Through the study of technology applications foundations, including technology-related terms, concepts, and data input strategies, students learn to make informed decisions about technologies and their applications. The efficient acquisition of information includes the identification of task requirements; the plan for using search strategies; and the use of technology to access, analyze, and evaluate the acquired information. By using technology as a tool that supports the work of individuals and groups in solving problems, students will select the technology appropriate for the task, synthesize knowledge, create a solution, and evaluate the results. Students communicate information in different formats and to diverse audiences. A variety of technologies will be used. Students will analyze and evaluate the results.~~

(2) 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, with their instructor, and with 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 a solution, and evaluate the results. Students will learn to become responsible digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Computer Science II course. Students will gain an understanding of computer science through the study of technology operations, systems, and concepts.

(c) Knowledge and skills.

~~(1) Foundations. The student demonstrates knowledge and appropriate use of hardware components, software programs, and their connections. The student is expected to:~~

- ~~(A) identify object-oriented data types and delineate the advantages/disadvantages of object data;~~
- ~~(B) demonstrate coding proficiency in contemporary programming languages including an object-oriented language; and~~
- ~~(C) survey the issues accompanying the development of large software systems such as design/implementation teams, software validation/testing, and risk assessment.~~

(1) Creativity and innovation. The student will develop products and generate new understanding by extending existing knowledge. The student is expected to:

- (A) use program design problem-solving strategies to create program solutions;
- ~~(B)~~~~(A)~~ demonstrate the ability to read and modify large programs, including the design description and process development;
- (C) 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;
- (D) follow the systematic process to problem solving of identifying the specifications of purpose and goals, the data types and objects needed, and the subtasks to be performed;
- ~~(E)~~~~(C)~~ compare and contrast design methodologies including and implementation techniques such as top-down, and bottom-up, and black box;
- (F) analyze, modify, and evaluate existing code by performing a case study on a large program, including inheritance and black box programming;
- (G) identify the data types and objects needed to solve a problem;
- (H) 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;
- (I) use object-oriented programming development methodology, data abstraction, encapsulation with information hiding, and procedural abstraction in program development and testing; and
- (J) create, edit, and manipulate bitmap images that are used to enhance user interfaces and program functionality.

Comment [A56]: Clarification based on feedback

Comment [A57]: Clarification, any case study is appropriate

Comment [A58]: MV

~~(2) Foundations. The student uses data input skills appropriate to the task. The student is expected to:~~

- ~~(A) demonstrate proficiency in the use of a variety of input devices such as keyboard, scanner, voice/sound recorder, mouse, touch screen, or digital video by appropriately incorporating such components into the product; and~~
- ~~(B) use digital keyboarding standards for the input of data.~~

(2) Communication and collaboration. The student will communicate and collaborate with their peers to contribute to the learning of themselves and others. The student is expected to:

Comment [A59]: based on feedback

(A) create interactive console display interfaces with appropriate user prompts;

Comment [A60]: Covered in prerequisites

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

Comment [A61]: 21<sup>st</sup> - Rationale – GUI's are ubiquitous. To create a GUI is to solve a problem. Gui's are an essential 21<sup>st</sup> century skill for students to understand and create interfaces for humans to interact with computers.

(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;

Comment [A62]: Based on feedback

(D) improve data display by optimizing data visualization;

Comment [A63]: Based on feedback

(E) display simple vector graphics to interpret and display program results; and

Comment [A64]: MV

(F) display simple bitmap images.

Comment [A65]: Covered in CS1

(3) Research and information fluency. The student will locate, analyze, process, and organize data. The student is expected to:

(A) use Local Area Networks (LANs) and Wide Area Networks (WANs), including the Internet and intranet, in research, file management, and collaboration;

Comment [A66]: ER

(B) understand programming file structure and file access for required resources;

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

Comment [A67]: Based on feedback

(D) ~~(7)(B)~~ manipulate data structures using string processing;

Comment [A68]: Based on feedback

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

Comment [A69]: Based on feedback

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

Comment [A70]: Based on feedback

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

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

Comment [A71]: Based on feedback

~~(4) Information acquisition. The student uses a variety of strategies to acquire information from electronic resources, with appropriate supervision. The student is expected to:~~

~~(A) construct search algorithms including linear and binary searches; and~~

(4) Critical thinking, problem solving, and decision making. The student will use 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;

Comment [A72]: VA

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

Comment [A73]: Based on feedback

(C) demonstrate proficiency in the use of short-circuit evaluation;

- (D) demonstrate proficiency in the use Boolean algebra, including De Morgan's Law;
- (E) develop iterative algorithms using nested loops;
- (F) ~~(7)(A)~~ identify, trace, and use appropriately use and trace recursion in programming solutions, including algebraic computations design-comparing invariant, iterative, and recursive algorithms;
- (G) design, construct, evaluate, and compare search algorithms, including linear searching and binary searching;
- (H) identify, describe, design, create, 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;
- (J) ~~(4)(B)~~ compare and contrast search and sort algorithms, including linear, quadratic, and recursive strategies and binary searches for different purposes and search time. for time/space efficiency;
- (K) ~~(9)(B)~~ analyze algorithms using "big-O" notation, for best, average, and worst-case data patterns space-techniques;
- (L) develop algorithms to solve various problems, including factoring, summing a series, finding the roots of a quadratic equation, and generating Fibonacci numbers;
- (M) test program solutions by investigating boundary conditions, testing classes, methods, and libraries in isolation, and performing stepwise refinement;
- (N) identify and debug compile, syntax, runtime and logic errors;
- (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;
- (P) demonstrate the ability to count, convert, and perform mathematical operations in the binary and hexadecimal number system;
- (Q) demonstrate knowledge of the maximum integer boundary, minimum integer boundary, imprecision of real number representations, and round-off errors;
- (R) create program solutions to problems using the mathematics library class; (S) use random algorithms to create simulations that model the real world;
- (T) identify, understand, and create class specifications and relationships among classes, including composition and inheritance relationships;
- (U) understand and explain object relationships among defined classes, abstract classes, and interfaces;
- (V) create object-oriented definitions using class declarations, variable declarations, constant declarations, method declarations, parameter, and interface declarations;
- (W) create robust classes that encapsulate data and the methods that operate on that data and incorporate overloading to enrich the objects behavior;

Comment [A74]: VA

Comment [A75]: VA – type of loop not specified making this more inclusive.

Comment [A76]: VA quick sort is covered in CS3

Comment [A77]: MV

Comment [A78]: VA

Comment [A79]: MV

Comment [A80]: MV

- (X) design and implement a set of interactive classes;
- (Y) design, create, and evaluate multiclass programs that utilize abstract classes and interfaces;
- (Z) understand and implement a student-created class hierarchy;
- (AA) use inheritance to extend, modify, and improve existing code;
- (BB) use polymorphism to create adaptive behaviors, including overloading;
- (CC) understand and use reference variables for object and string data types;
- (DD) understand and implement access scope modifiers;
- (EE) understand and demonstrate how to compare objects;
- (FF) duplicate objects using the appropriate deep and/or shallow copy;
- (GG) define and implement abstract classes and interfaces in program problem solutions;
- (HH) apply functional decomposition to a program solution;
- (II) create simple and robust objects from class definitions through instantiation;
- (JJ) apply class membership of variables, constants, and methods;
- (KK) use accessors and modifiers to examine and mutate the properties of an object;
- (LL) understand and implement a composite class; and
- (MM) design and implement an interface.

Comment [A81]: Based on feedback

Comment [A82]: MV

- ~~(5) Information acquisition. The student acquires electronic information in a variety of formats, with appropriate supervision. The student is expected to:~~
  - ~~(A) acquire information in and knowledge about electronic formats including text, audio, video, and graphics;~~
  - ~~(B) use a variety of resources, including foundation and enrichment curricula, together with various productivity tools to gather authentic data as a basis for individual and group programming projects; and~~
- (5)(3) Digital citizenship Foundations. The student will explore and understand safety, legal, cultural, and societal ~~complies with the laws and examines the~~ issues relating to ~~regarding~~ the use of technology ~~in society and information~~. The student is expected to:
  - ~~(A) discuss copyright laws/issues and~~ model ethical acquisition and use of digital information, ~~citing sources using established methods;~~
  - (B) demonstrate proper digital etiquette, responsible use of software, and knowledge of acceptable use policies ~~when using networks, especially resources on the Internet and intranet; and~~ and
  - ~~(C) investigate measures, such as passwords or virus detection/prevention, to protect computer systems and databases from unauthorized use and tampering; and~~
  - (C) investigate digital rights management.
  - ~~(D) code modules for the World Wide Web (WWW) community.~~



- ~~(6) Information acquisition. The student evaluates the acquired electronic information. The student is expected to:
  - ~~(A) determine and employ methods to evaluate the design and functionality of the process using effective coding, design, and test data; and~~
  - ~~(B) implement methods for the evaluation of the information using defined rubrics.~~~~
- (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 types of operating systems, software applications, hardware platforms, and programming languages;
  - (B) demonstrate knowledge of major hardware components, including primary and secondary memory, the central processing unit, and peripherals;
  - (C) demonstrate knowledge of major networking components, including hosts, servers, switches, and routers;
  - (D) demonstrate knowledge of computer communication systems, including single-user, peer-to-peer, workgroup, client-server, and networked;
  - (E) demonstrate knowledge of computer addressing systems, including Internet Protocol (IP) address and Media Access Control (MAC) address; and
  - (F) differentiate among the categories of programming languages, including machine, assembly, high-level compiled, high-level interpreted, and scripted.
- ~~(7) Solving problems. The student uses appropriate computer based productivity tools to create and modify solutions to problems. The student is expected to:
  - ~~(C) use notation for language definition such as syntax diagrams or Backus-Naur forms;~~
  - ~~(D) identify, describe, and use sequential/non-sequential files; multidimensional arrays and arrays of records; and quadratic sort algorithms such as selection, bubble, or insertion, and more efficient algorithms including merge, shell, and quick sorts;~~
  - ~~(E) create robust programs with increased emphasis on design, style, clarity of expression and documentation for ease of maintenance, program expansion, reliability, and validity;~~
  - ~~(F) apply methods for computing iterative approximations and statistical algorithms;~~
  - ~~(G) define and develop code using the concepts of abstract data types including stacks, queues, linked lists, trees, graphs, and information hiding;~~
  - ~~(H) identify and describe the correctness and complexity of algorithms such as divide and conquer, backtracking, or greedy algorithms;~~
  - ~~(I) develop software to solve a school or community problem such as customer relations, design, modular programming, documentation, validation, marketing, or support; and~~
  - ~~(J) research advanced computer science concepts such as applied artificial intelligence, expert systems, robotics, depth first/breadth first and heuristic search strategies, multitasking operating systems, or computer architecture, such~~~~

Comment [A83]: ER

Comment [A84]: MV  
VA – covered in prerequisite

Comment [A85]: VA covered in CS1

~~as reduced instruction set computer (RISC) and complex instruction set computer (CISC).~~

- ~~(8) Solving problems. The student uses research skills and electronic communication, with appropriate supervision, to create new knowledge. The student is expected to:
  - ~~(A) participate with electronic communities as a learner, initiator, contributor, and teacher/mentor;~~
  - ~~(B) demonstrate proficiency in, appropriate use of, and navigation of local area networks (LANs) and wide area networks (WANs) for research and for sharing of resources;~~
  - ~~(C) extend the learning environment beyond the school walls with digital products created to increase teaching and learning in the foundation and enrichment curricula; and~~
  - ~~(D) participate in relevant, meaningful activities in the larger community and society to create electronic projects.~~~~
- ~~(9) Solving problems. The student uses technology applications to facilitate evaluation of work, both process and product. The student is expected to:
  - ~~(D) analyze models used in development of software including software life cycle models, design objectives, documentation, and support; and~~
  - ~~(E) seek and respond to advice from peers and professionals in delineating technological tasks.~~~~
- ~~(10) Communication. The student formats digital information for appropriate and effective communication. The student is expected to:
  - ~~(A) annotate coding properly with comments, indentation, and formatting;~~
  - ~~(B) create interactive documents using modeling, simulation, and hypertext.~~~~
- ~~(11) Communication. The student delivers the product electronically in a variety of media, with appropriate supervision. The student is expected to:
  - ~~(A) publish information in a variety of ways including, but not limited to, printed copy and monitor displays; and~~
  - ~~(B) publish information in a variety of ways including, but not limited to, software, Internet documents, and video.~~~~
- ~~(12) Communication. The student uses technology applications to facilitate evaluation of communication, both process and product. The student is expected to:
  - ~~(A) write technology specifications for planning and evaluation rubrics documenting variables, prompts, and program internally and externally;~~
  - ~~(B) seek and respond to advice from peers and professionals in evaluating the product; and~~
  - ~~(C) debug and solve problems using reference materials and effective strategies.~~~~

**§126.xx. Computer Science III (One Credit).**

(a) General requirements. The prerequisite for this course is proficiency in the knowledge and skills for Computer Science II as identified in §126.22(c) of this title (relating to Computer Science II (One Credit)), proficiency in the knowledge and skills for Computer Science AP §xxx or proficiency in the knowledge and skills for Computer Science IB §xxx. This course is recommended for students in Grades 11-12.

(b) Introduction.

(1) The technology applications curriculum has six strands based on the National Educational Technology Standards (NETS•S) and Performance Indicators for Students 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, with their instructor, and with 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 a solution, and evaluate the results. Students will learn to become digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Computer Science III course. Students will gain an understanding of advanced computer science data structures through the study of technology operations and concepts.

(c) Knowledge and skills.

(1) Creativity and innovation. The student will develop products and generate new understanding 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.

Comment [A86]: clarification

Comment [A87]: VA; clarification

- (2) Communication and collaboration. The student will communicate and collaborate with their peers to contribute to the learning of themselves and others. The student is expected to:
- (A) use local area networks (LANs) and wide area networks (WANs), including the Internet and intranet, in research, file management, and collaboration;
  - (B) create interactive human interfaces to acquire data from a user and display program results using a 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 will locate, analyze, process, and organize 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 2-D ragged arrays to traverse, search, modify, insert, and delete data;
  - (C) identify and use a list object data structure, including vector, to traverse, insert, search, 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 linked list of linked list;
  - (G) understand and create program solutions using Stacks;
  - (H) understand and create program solutions using Queues;
  - (I) understand and create program solutions using Trees;
  - (J) understand and create program solutions using Heaps;
  - (K) understand and create program solutions using Priority Queues;
  - (L) understand and create program solutions using Sets, including HashSet and TreeSet;
  - (M) understand and create program solutions using Maps, including HashMap and TreeMap;
  - (N) understand and create program solutions using Graph Theory;
  - (O) understand and create program solutions using enumerated data types; and
  - (P) write and modify text file data.

**Comment [A88]:** 21<sup>st</sup> - Rationale – GUI's are ubiquitous. To create a GUI is to solve a problem. Gui's are an essential 21<sup>st</sup> century skill for students to understand and create interfaces for humans to interact with computers.

**Comment [A89]:** VA

**Comment [A90]:** for clarity

**Comment [A91]:** Based on feedback

**Comment [A92]:** clarification

- (4) Critical thinking, problem solving, and decision making. The student will use 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 use recursion appropriately;
  - (G) understand and create program solutions using hashing;
  - (H) use regular expressions to perform pattern recognition;
  - (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 will explore and understand 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;
  - (D) create discovery programs in a low-level language, high-level language, and a scripting language.

Comment [A93]: ER

Comment [A94]: VA CS2; ER

Comment [A95]: ER

Comment [A96]: VA CS2

Comment [A97]: MV

Comment [A98]: MV

Comment [A99]: MV

**§126.xx. Digital Forensics (One-Half to One Credit).**

(a) General requirements. It is recommended that students have participated in a program that has stressed positive character traits as outlined in Texas Education Code §29.906. The prerequisite for this course is proficiency in the knowledge and skills relating to Technology Applications Grades 6 -8. This course is recommended for students in Grades 9 – 12.

(b) Introduction.

(1) The technology applications curriculum has six strands based on the National Educational Technology Standards (NETS•S) and Performance Indicators for Students 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) Digital Forensics will foster students' creativity and innovation by presenting opportunities to investigate simulations and case studies of crimes, reconstructing computer security incidents, troubleshooting operational problems, and recovering from accidental system damage. Students will collaborate to develop forensic techniques to assist with computer security incident responses. Students will solve problems as they study the application of science to the law. Students will gain an understanding of computing and network systems that transmit or store electronic data. Students will learn to become digital citizens by practicing integrity and respect throughout digital forensics. In this course, students learn methods to identify, collect, examine, and analyze data while preserving the integrity of the information and maintaining a strict chain of custody for data.

Comment [A100]: MV

(c) Knowledge and skills.

(1) Creativity and innovation. The student will develop products and generate new understanding by extending existing knowledge. The student is expected to

(A) explain the need for digitals forensics, staffing requirements, and team interactions;

(B) develop policies to define staff roles and responsibilities;

(C) develop guidelines, procedures, and recommendations for digital forensic tool use; and

(D) investigate simulations and case studies of crimes to reconstruct computer security incidents.

Comment [A101]: MV

(2) Communication and collaboration. The student will communicate and collaborate with their peers to contribute to the learning of themselves and others. The student is expected to:

(A) describe the characteristics and behaviors of a given system;

(B) justify and describe the impact of selecting a given system;

(C) apply effective teamwork practices;

(D) collaborate with multiple participants;

(E) document use, functionality, and implementation;

Comment [A102]: MV

Comment [A103]: MV

- (F) seek and respond to advice from peers and professionals; and
- (G) describe considerations required for incident response.
- (2) Research and information fluency. The student will locate, analyze, process, and organize data. The student is expected to
  - (A) identify possible sources of data;
  - (B) acquire data;
  - (C) analyze and report data collected;
  - (D) collect files by copying files from media while maintaining data file integrity;
  - (E) examine data files by locating files, extracting data, and using a digital forensic toolkit;
  - (F) examine and analyze operating system data;
  - (G) collect volatile and non-volatile operating system data;
  - (H) collect, examine, and analyze application data;
  - (I) use traffic data sources, including firewalls and routers, packet sniffers and protocol analyzers, intrusion detection systems, remote access, security event management software, and network forensic analysis tools;
  - (J) describe how a file scan can be accessed and modified;
  - (K) collect, examine, and analyze data from multiple sources; and
  - (L) provide examples of how multiple data sources can be used during digital forensics, including investigating worm infections, viruses, and e-mail threats.
- (4) Critical thinking, problem solving, and decision making. The student will use appropriate strategies to analyze problems and design algorithms. The student is expected to
  - (A) resolve information conflicts and validate information through data acquisition, research, and comparison; and
  - (B) examine and analyze network traffic data, including identifying events of interest, examining data sources, and identifying attacks.
- (5) Digital citizenship. The student will explore and understand safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to:
  - (A) identify and use digital information appropriately;
  - (B) identify and use appropriate methods for citing sources;
  - (C) identify and discuss intellectual property laws, issues, and use;
  - (D) identify intellectual property stakeholders, their needs, and perspectives;
  - (E) identify and describe the kinds of crimes investigated by digital forensics specialists;
  - (F) identify legal, illegal, ethical, and unethical aspects of information gathering;

Comment [A104]: MV

Comment [A105]: MV

Comment [A106]: MV

- (G) compare and contrast legal, illegal, ethical, and unethical information gathering methods and identify possible gray areas;
- (H) identify and describe ways in which developing laws and guidelines affect digital forensics practices;
- (I) identify and describe legal considerations and technical issues related to collecting network traffic data;
- (J) identify and describe ways in which technological changes affect applicable laws;
- (K) identify and describe ways in which developing laws and guidelines affect digital forensics practices; and
- (L) identify and describe businesses and government agencies that use digital forensics.

(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) demonstrate knowledge and appropriately use operating systems, software applications, and communication and networking components;
- (B) compare, contrast, and appropriately use the various input, processing, output, and primary and secondary storage devices;
- (C) make decisions regarding the selection, acquisition, and use of software, including quality, appropriateness, effectiveness, and efficiency;
- (D) demonstrate knowledge of data formats;
- (E) demonstrate knowledge of networks, including the Internet, intranets, and extranets;
- (F) compare and contrast non-volatile and volatile data;
- (J) describe file basics, including file storage, file systems, and other types of storage media;
- (K) describe file modification, including access and creation times;
- (L) describe operating systems, including terminology and functions;
- (M) describe technical procedures related to collecting operating system data;
- (N) describe the significance to digital forensics of the Transmission Control Protocol/Internet Protocol (TCP/IP) model including application, transport, IP, and hardware layers;
- (O) describe the function and use of application components including configurations settings, authentications, logs, application data, supporting files, application architecture; and
- (P) describe the function and use of application types, including e-mail, web usage, interactive communications, file sharing, document usage, security applications, and data concealment tools.

Comment [A107]: MV



**§126.xx. Discrete Mathematics (One-half to One Credit)**

(a) General requirements. The prerequisite for this course is Algebra II. This course is recommended for students in Grades 11-12

(b) Introduction.

(1) The technology applications curriculum has six strands based on the National Educational Technology Standards (NETS•S) and Performance Indicators for Students 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) Discrete Mathematics provides the tools used in most areas of computer science. Exposure to the mathematical concepts and discrete structures presented in this course is essential in order to provide an adequate foundation for further study. Discrete Mathematics is generally listed as a core requirement for Computer Science majors. The intention is that upon successful completion of this course, students will develop the ability to see computational problems from a mathematical perspective. Course topics are divided into six areas: functions, relations, and sets; basic logic; proof techniques; counting basics; graphs and trees; and discrete probability. Math topics are interwoven with computer science applications to enhance the students' understanding of the introduced mathematics. Introduced to a formal system (propositional and predicate logic) upon which mathematical reasoning is based, students will acquire the necessary knowledge to be able to both read and construct mathematical arguments (proofs), understand mathematical statements (theorems), use mathematical problem-solving tools and strategies as well as be introduced to discrete data structures such as sets, relations and discrete functions, graphs and trees. Students will also be introduced to discrete probability and expectations.

(c) Knowledge and skills.

(1) Creativity and innovation. The student will develop products and generate new understandings by extending existing knowledge. The student is expected to:

- (A) model algorithms and real-world situations using formal tools of symbolic logic;
- (B) model computer science problems by using graphs and trees; and
- (C) calculate the probabilities of events and expectations of random variables for such problems as games of chance.

(2) Communication and collaboration. The student will communicate and collaborate with peers to contribute to the learning of themselves and others. The student is expected to:

- (A) convert spoken language statements to appropriate statements in propositional logic;
- (B) explain basic terminology of functions, relations, and sets;
- (C) state the definition of the Master theorem;
- (D) use the context of a particular application to interpret the meaning derived when computing the permutations and combinations of a set;
- (E) interpret associated operations and terminology in context; and

Comment [A108]: VA

Comment [A109]: VA

(F) define and provide examples of logical equivalence, normal forms, validity, modus ponens/modus tollens.

(3) Research and information fluency. The student will locate, analyze, process, and organize data. The student is expected to:

(A) construct truth tables for negation, conjunction, disjunction, implication, biconditional, and bit operators; and

(B) use truth tables to demonstrate propositional relations.

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

(A) analyze practical examples using appropriate models of sets, functions, and relations;

(B) compare and contrast tautology, contradiction, and contingency as the terms relate to propositional equivalences;

(C) compare and contrast examples and use of counterexamples, contrapositions, and contradictions;

(D) describe the appropriate use and limitations of predicate logic;

(E) apply formal methods of symbolic propositional and predicate logic;

(F) use formal logic proofs and logical reasoning to solve problems;

(G) outline the basic structure of proofs, including direct, indirect, contradiction, induction, existence, and constructive proofs;

(H) compare and contrast the type of problem best satisfied by direct, indirect, contradiction, induction, existence, and constructive proofs;

(I) relate mathematical induction to recursion and recursively defined structures;

(J) compare and contrast weak, strong, and structural induction, including when each is most appropriately used and examples of each;

(K) compare and contrast dependent and independent events;

(L) use recurrence equations to analyze algorithms and other practical problems;

(M) use counting techniques to analyze algorithms and other practical problems;

(N) apply probability tools to solve problems; and

(O) define, compare, and contrast simple graphs, multigraphs, and directed and undirected graphs using definitions, properties, and examples including special cases.

(5) Digital Citizenship. The student will explore and understand 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;

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

Comment [A110]: VA

Comment [A111]: VA

Comment [A112]: VA

(C) investigate how the concepts of discrete mathematics are related to relevant problems and significant questions.

Comment [A113]: Relevance

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

Comment [A114]: VA

(A) perform operations associated with sets, functions, and relations;

(B) apply basic counting principles, including cardinality and the pigeonhole principle;

(C) apply appropriate precedence when using logical operators;

(D) use appropriate strategies, including De Morgan's Laws, to identify propositional equivalences;

(E) identify and appropriately use predicates, existential and universal quantifiers, and valid arguments;

(F) identify possible applications of proofs, including evaluating algorithmic complexity;

(G) state and appropriately use the product and sum rules;

(H) compute permutations and combinations of a set;

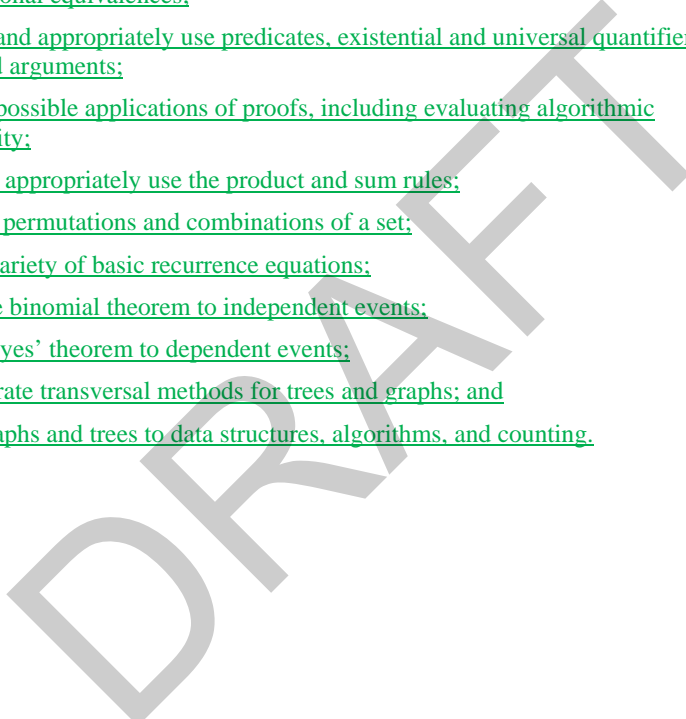
(I) solve a variety of basic recurrence equations;

(J) apply the binomial theorem to independent events;

(K) apply Bayes' theorem to dependent events;

(L) demonstrate transversal methods for trees and graphs; and

(M) relate graphs and trees to data structures, algorithms, and counting.



**§126.xx. Game Programming and Design (One-half to One Credit).**

- (a) General requirements. The prerequisite for this course is proficiency in the knowledge and skills for Algebra I identified in §111.32(b) of this title (relating to Algebra I (One Credit)). This course is recommended for students in Grades 9-12.
- (b) Introduction.
- (1) The technology applications curriculum has six strands based on the National Educational Technology Standards (NETS•S) and Performance Indicators for Students 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) Game programming will foster student creativity and innovation by presenting opportunities to design, implement, and present meaningful programs through a variety of media. Students will collaborate with one another, with their instructor, and with various electronic communities to solve gaming problems presented throughout the course. Data analysis will include the identification of task requirements, planning search strategies, and the use of programming concepts to access, analyze, and evaluate information needed to design games. By acquiring programming 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 a solution, and evaluate the results. Students will learn to become good digital citizens by practicing integrity and respect throughout the course. Upon completion of the course, the student will have created a computer game that is presented to an evaluation panel.
- (c) Knowledge and skills.
- (1) Creativity and innovation. The student will develop products and generate new understanding by extending existing knowledge. The student is expected to:
    - (A) understand the basic game design elements, including conceptual ideas, storyline, visualization, storyboard, game effects, sound elements, game play, game controls, and player tutorial;
    - (B) create a design concept document;
    - (C) create a storyboard;
    - (D) understand the fundamentals of game art, including the look and feel, the graphics coordinate system, the basics of color, and color palettes;
    - (E) use bitmap graphics images, including designing, creating, reading, and manipulating images ;
    - (F) create backgrounds, including solid, image, and tiled backgrounds;
    - (G) write programs creating images using geometric shapes;
    - (H) create games using sprites by evaluating the role of sprites, creating sprites, and managing sprites;
    - (I) create programs using sprite sheets;

- (J) understand image rendering, including transparency, refresh rate, hardware acceleration, and animation;
- (K) find, create, and edit game audio sound effects and music; and
- (L) implement game sound mechanics, including playing, pausing, and looping.
- (2) Communication and collaboration. The student will communicate and collaborate with their peers to contribute to the learning of themselves and others. The student is expected to:
  - (A) design and implement procedures to set timelines for, track the progress of, and evaluate a game product;
  - (B) seek and respond to input from peers and professionals in evaluating a game project;
  - (C) demonstrate knowledge and appropriate use of operating systems, program development tools, and networking resources;
  - (D) use network resources to acquire, organize, maintain, and evaluate information;
  - (E) collaborate to research the business of games, including the roles of developer, marketing, publisher, and retail sales; and
  - (F) understand and evaluate online technology, including online interaction and massive multiplayer games.
- (3) Research and information fluency. The student will locate, analyze, process, and organize data. The student is expected to:
  - (A) play board games to research and collect game play data;
  - (B) evaluate, analyze, and document game styles and playability; and
  - (C) research the dramatic elements in games, including kinds of fun, player types, and nonlinear storytelling.
- (4) Critical thinking, problem solving, and decision making. The student will use appropriate strategies to analyze problems and design algorithms. The student is expected to:
  - (A) understand the game design process, including generating ideas, brainstorming, and paper prototyping;
  - (B) write programs using variables of different data types;
  - (C) evaluate game rules and instructions;
  - (D) understand the user experience by comparing rules and game-play patterns;
  - (E) write game rules and instructions;
  - (F) develop game software;
  - (G) write computer game code, resolve game defects, and revise existing game code; and
  - (H) test a finished game product by implementing sound testing techniques.

- (5) Digital citizenship. The student will explore and understand safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to:
- (A) explore intellectual property, privacy and sharing of information, copyright laws, and software licensing agreements;
  - (B) model ethical acquisition and use of digital information;
  - (C) demonstrate proper etiquette when using networks, responsible use of software, and knowledge of acceptable use policies;
  - (D) model respect of intellectual property, including manipulating graphics, morphing graphics, editing graphics, and editing sound;
  - (E) discuss and evaluate the social issues surrounding gaming; and
  - (F) evaluate the cultural aspects of game design fundamentals, including rationale for games and types of games.
- (6) Technology operations and concepts. The student understands technology concepts, systems, and operations as they apply to game programming. The student is expected to:
- (A) identify basic game components, including the game engine, game play subsystems, data structures, models, and interfaces;
  - (B) generate random numbers in a program;
  - (C) create a program implementing conditional statements;
  - (D) develop an appropriate data model ;
  - (E) understand and apply object oriented when programming games;
  - (F) understand game programming essentials, including event-driven programming, communicating with messages, and device management;
  - (G) understand the role of game events, the animation loop, and game timing;
  - (H) understand the role of game engines;
  - (I) understand video display flicker and double buffering;
  - (J) apply basic game screen design and layout, including visual controls, user interfaces, menus, and options;
  - (K) use game control design to understand, access, and control input devices including, keyboard, mouse, and joystick;
  - (L) understand game animation and apply animation to a game, including the principles of animation and frame-based animation;
  - (M) understand decision making and types of decisions;
  - (N) understand game events, including listeners, triggers, and timed events;
  - (O) understand and implement collision detection, including bounding boxes and sprite collisions;

- (P) implement a tile-based game, including loading tile maps, drawing tile maps, rendering a tile map, and layering sprites;
- (Q) understand, develop, and implement artificial intelligence;
- (R) understand game balance and tuning; and
- (S) understand player progression, including leveling, linear progression, and maintaining high score data.

DRAFT

**§126.xx. Mobile Application Development (One-half to One Credit)**

(a) General requirements. The prerequisite for this course is proficiency in the knowledge and skills relating to Technology Applications Grades 6 -8 and that students have proficiency in the knowledge and skills for Algebra I identified in §111.32(b) of this title relating to Algebra I. This course is recommended for students in Grades 9 – 12.

Comment [A115]: VA

(b) Introduction.

Comment [A116]: VA

(1) The technology applications curriculum has six strands based on the National Educational Technology Standards (NETS•S) and Performance Indicators for Students 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) Mobile Application Development will foster students' creativity and innovation by presenting opportunities to design, implement, and deliver meaningful projects using mobile computing devices. Students will collaborate with one another, with their instructor, and with various electronic communities to solve problems presented throughout the course. Data analysis will include the identification of task requirements, planning search strategies, and the use of software development concepts to access, analyze, and evaluate information needed to program mobile devices. By using software design 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 a solution, and evaluate the results. Students will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect. Students will gain an understanding of the principles of mobile application development through the study of development platforms, programming languages, and software design standards.

(c) Knowledge and skills.

Comment [A117]: VA

(1) Creativity and innovation. The student will develop products and generate new understanding by extending existing knowledge. The student is expected to:

(A) create effective user interfaces appropriate for a specified mobile device that is best suited for an identified purpose;

(B) create effective user interfaces for browser-based, native, and hybrid mobile applications;

(C) create mobile application components appropriate for identified needs;

(D) create browser-based applications for mobile devices;

(E) create native applications that can reside on specified mobile devices; and

(F) create mobile applications that combine native and hybrid components.

Comment [A118]: MV

Comment [A119]: MV

Comment [A120]: MV

Comment [A121]: MV

Comment [A122]: VA

(2) Communication and collaboration. The student will communicate and collaborate with peers to contribute to the learning of themselves and others. The student is expected to:

(A) understand and discuss how teams function;

(B) use teamwork to solve problems;

Comment [A123]: MV

Comment [A124]: MV



- (C) describe the development workflow of mobile applications;
- (D) use time-management techniques to develop and maintain work schedules and meet deadlines and establish mobile application project criteria;
- (E) describe a problem solution; and
- (F) document and share problem solutions through various forms of media.
- (3) Research and information fluency. The student will locate, analyze, process, and organize data. The student is expected to:
  - (A) analyze, identify, and describe mobile application project stakeholders and their perspectives;
  - (B) collect and analyze available data to identify mobile application project requirements;
  - (C) analyze, identify, and describe input, output, and processing requirements; and
  - (D) analyze, identify, and define hardware and software specifications.
- (4) Critical thinking, problem solving, and decision making. The student will use appropriate strategies to analyze problems and design algorithms. The student is expected to:
  - (A) compare and contrast design decisions based on the hardware considerations of the mobile device;
  - (B) compare and contrast available mobile technologies, including platforms and their operating systems;
  - (C) compare and contrast available development approaches, including application to specific technologies and platforms;
  - (D) determine the most appropriate solution for the development of a given mobile application, including browser-based, native, and hybrid approaches;
  - (E) compare and contrast available programming languages and how their use might be applied to specific technologies and platforms;
  - (F) identify and justify the selection of an appropriate programming language, including available resources and required interfaces;
  - (G) select an appropriate program development environment;
  - (H) identify and use available libraries;
  - (I) evaluate and justify the selection of appropriate options and components;
  - (J) compare and contrast available networks and their implications for mobile application development; and
  - (K) compare and contrast design strategies related to mobile network and device security.
- (5) Digital citizenship. The student will explore and understand safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to:
  - (A) discuss copyright laws and issues;

Comment [A125]: MV

Comment [A126]: VA

Comment [A127]: VA

Comment [A128]: MV

Comment [A129]: MV

Comment [A130]: MV

Comment [A131]: MV

Comment [A132]: MV

Comment [A133]: MV

Comment [A134]: MV

Comment [A135]: MV

- (B) model ethical acquisition and use of digital information;
- (C) cite sources using established methods;
- (D) demonstrate proper digital etiquette and knowledge of acceptable use policies;
- (E) investigate mobile device security measures such as passwords, virus detection, and virus prevention;
- (F) describe potential risks and benefits associated with the use of the mobile application;
- (G) identify current and emerging technologies relating to mobile applications; and
- (H) evaluate technologies and assess their applicability to current mobile applications.

(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) understand the difference between desktop and mobile applications;
- (B) understand hardware and software structures and requirements in the design of mobile applications;
- (C) recognize multiple platforms and understand their associated requirements;
- (D) recognize various program development environments;
- (E) understand event-based programming and appropriate use;
- (F) describe how memory management affects mobile application design;
- (G) understand how low bandwidth and the mobility of the device affect the design of mobile applications;
- (H) identify applications that are best suited for mobile devices;
- (I) understand the use of libraries when designing mobile applications;
- (J) use a simulation tool to emulate a mobile device's functionality; and
- (M) use actual mobile devices to test mobile applications.

Comment [A136]: MV

Comment [A137]: VA

Comment [A138]: moved

Comment [A139]: MV

Comment [A140]: MV

**§126.xx. Robotics Programming and Design (One-half to One Credit)**

(a) General requirements. The prerequisite for this course is proficiency in the knowledge and skills relating to Technology Applications Grades 6 -8. This course is recommended for students in Grades 9–12.

(b) Introduction.

(1) The technology applications curriculum has six strands based on the National Educational Technology Standards (NETS•S) and Performance Indicators for Students 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.

Comment [A141]: VA

(2) Robotics Programming and Design will foster students' creativity and innovation by presenting opportunities to design, implement, and present meaningful robotic programs through a variety of media. Students will collaborate with one another, with their instructor, and with various electronic communities to solve problems in designing and programming robots. Data analysis will include the identification of task requirements, planning search strategies, and the use of robotic concepts to access, analyze, and evaluate information needed to solve problems. By using robotic 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 a solution, and evaluate results. Students will learn to become digital citizens by practicing integrity and respect. Students will gain an understanding of the principles of robotics through the study of physics, robotics, automation, and engineering design concepts.

Comment [A142]: MV

(c) Knowledge and skills.

(1) Creativity and innovation. The student will develop products and generate new understanding by extending existing knowledge. The student is expected to:

(A) produce a prototype;

(B) present a prototype using a variety of media;

(C) use the design process to construct a robot;

(D) refine the design of a robot;

(E) build robots of simple, moderate, and advanced complexity;

Comment [A143]: VA

(F) improve a robot design to meet a specified need;

(G) understand and create artificial intelligence in a robot; and

(H) create behavior-based control algorithms.

Comment [A144]: VA

(2) Communication and collaboration. The student will communicate and collaborate with peers to contribute to the learning of themselves and others. The student is expected to:

Comment [A145]: MV

(A) understand and implement design teams;

Comment [A146]: based on feedback

(B) use design teams to solve problems;

(C) serve as a team leader and a team member;

Comment [A147]: MV

- (D) describe a problem and identify design specifications;
- (E) design a solution to a problem and share a solution through various forms of media;
- (F) document prototypes, adjustments, and corrections in the design process; Comment [A148]: MV
- (G) document the final design and solution; and
- (H) present the final design, testing results, and solution.
- (3) Research and information fluency. The student will locate, analyze, process, and organize data. The student is expected to: Comment [A149]: VA
  - (A) test and evaluate the robot design;
  - (B) implement position tracking to complete assigned robot tasks; Comment [A150]: MV; VA
  - (C) develop solution systems and implement systems analysis;
  - (D) modify a robot to respond to a change in specifications; and
  - (E) implement a system to identify and track all components of the robot. Comment [A151]: MV
- (4) Critical thinking, problem solving, and decision making. The student will use appropriate strategies to analyze problems and design algorithms. The student is expected to:
  - (A) develop algorithms to control a robot, including applying instructions, collecting sensor data, and performing simple tasks;
  - (B) create maneuvering algorithms to physically move the location of a robot;
  - (C) create algorithms that provide interaction with a robot; Comment [A152]: VA
  - (D) demonstrate an understanding of and use output commands;
  - (E) demonstrate an understanding of and use variables;
  - (F) demonstrate an understanding of and use sequence programming structure;
  - (G) demonstrate an understanding of and use jumps, loops, and selection programming structures;
  - (H) demonstrate an understanding of and use subroutines;
  - (I) demonstrate an understanding of and use accessors; Comment [A153]: MV
  - (J) demonstrate an understanding of and use modifiers; and
  - (K) apply decision-making strategies when developing solutions.
- (5) Digital citizenship. The student will explore and understand safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to: Comment [A154]: MV
  - (A) discuss intellectual property, privacy and sharing of information, copyright laws, and software licensing agreements; Comment [A155]: MV
  - (B) demonstrate proper digital etiquette, responsible use of software, and knowledge of acceptable use policies; and
  - (C) explore the effects robots have on changing our culture and society. Comment [A156]: MV

- (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) safely use tools and laboratory equipment to construct and repair robots;
  - (B) identify and describe the steps needed to produce a prototype;
  - (C) use software applications to simulate robotic behavior, present design concepts, and test solution strategies;
  - (D) demonstrate the use of computers to manipulate a robot;
  - (E) demonstrate knowledge of process control design factors;
  - (F) demonstrate knowledge of different types of sensors used in robotics;
  - (G) demonstrate knowledge and use of effectors;
  - (H) implement multiple sensors in a robot;
  - (I) interpret sensor feedback and calculate threshold values;
  - (J) demonstrate knowledge of motors, gears, and gear trains used in the robot;
  - (K) implement infra-red range sensing;
  - (L) apply measurement and geometry to calculate robot navigation;
  - (M) implement movement control using shaft encoding;
  - (N) demonstrate robot navigation;
  - (O) implement path planning using geometry and multiple sensor feedback;
  - (P) program a robot to perform simple tasks, including following lines, moving objects, and avoiding obstacles;
  - (Q) demonstrate and implement a robotic task solution using robotic arm construction;
  - (R) demonstrate knowledge of feedback control loops to provide information;
  - (S) demonstrate knowledge of torque and power factors used in the operation of a robot servo; and
  - (T) troubleshoot and maintain robotic systems and subsystems.

Comment [A157]: MV

Comment [A158]: VA

Comment [A159]: MV

Comment [A160]: MV

Comment [A161]: MV

Comment [A162]: clarification

Comment [A163]: MV