DRAFT Proposed Revisions

Texas Essential Knowledge and Skills

Technology Applications Computer Science

Prepared by the State Board of Education TEKS Review Committees

First Draft—July 2010

These documents have been combined from grade-level team drafts and formatted for consistency and ease of review.

Proposed additions are shown in greed font with underlines and proposed deletions are shown in red font with strike throughs.

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 21st century technology trends, applications, and uses

Table of Contents

Technology Applications, Fundamentals of Computing (One Credit)

(a) General requirements. This course is recommended for students in Grades 6-8.

(b) Introduction.

(1) The technology applications curriculum has six strands: I. Vision (Creativity and Innovation), II. Integrity (Digital Citizenship), III. Collaboration (Teamwork and Communication), IV. Data Analysis (Research and Information Fluency), V. Reasoning (Critical Thinking, Problem Solving, and Decision Making), VI. Principles (Technology Operations, Systems, and Concepts).

(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. . Key elements include pattern recognition, Interpretation, model building, model checking and data representation.

Comment [A1]: MV, CRS, AV Data Analysis Content

(c) Knowledge and skills.

(1) Principles. The student demonstrates knowledge and appropriate use of hardware components, software programs, and their connections, and use data input skills appropriate to the task. The student is expected to:

Comment [A2]: MV: & AV: Information combined from previous c2.

(A) demonstrate knowledge and appropriate use of operating systems, software applications, and communication and networking components;

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

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

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

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

(F) differentiate among the levels of programming languages including machine, assembly, high-level compiled and interpreted languages;

(G) demonstrate coding proficiency in a contemporary programming language;

(H) 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

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

Comment [A3]: MV: Added from 2 below.

(2) Integrity. The student complies with the laws and examines the issues regarding the use of technology in society. 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 etiquette and knowledge of acceptable use policies when using networks, especially resources on the Internet and intranet;

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

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

(3) Data Analysis. The student uses a variety of strategies to acquire and evaluate electronic information in a variety of formats. The student is expected to:

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

(B) construct appropriate electronic search strategies in the acquisition of information including keyword and Boolean search strategies.

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

(D) 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;

(E) design and document sequential search algorithms for digital information storage and retrieval;

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

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

Comment [A4]: MV: From (6) below.

(4) Data Analysis. The student uses data to create and modify solutions to problems in manner that is most appropriate.

(A) identify *classes* of patterns, e.g. – linear, exponential, etc;

(B) identify dependent and independent factors in a system;

Comment [A5]: MV, ,VA, Data Analysis Content

(C) identify the limitations of a particular representation, and give examples;

(D) code using various data types;

Comment [A6]: MV: from (7), Also in Vision and Reasoning

(E) identify actual and formal parameters and use value and reference parameters; and

Comment [A7]: MV: also in (7)

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

Comment [A8]: MV: also in (7)

(5) Reasoning. The student uses appropriate computer-based productivity tools to create and modify solutions to problems. The student is expected to:

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

(B) express patterns and complex functional relationships (multiple inputs, conditional behavior, different input and output types, etc.) as a program.

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

(D) apply pattern recognition strategies to differentiate patterns from random noise (in numbers, letters, images, etc.), explain the rule that produces a pattern and its significance.

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

(F) compose functions to create more complicated behaviors, by writing programs with multiple procedures.

(G) prove that two (simple) functions are or are not equivalent.

(H) create test cases for programs; prove a model incorrect through counterexamples; determine if a model is well-tested by examining the model and the test cases.

(I) code using various data types;

(J) express a functional relationship by writing a computer program, drawing a table of inputs and outputs, a formula, or a graph.

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

(L) develop 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;

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

(N) identify actual and formal parameters and use value and reference parameters;

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

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

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

(R) debug and solve problems using reference materials and effective strategies.

(6) Vision. The student uses research skills and electronic communication to create new knowledge. The student is expected to:

Comment [A9]: VA, MV, CRSData Analysis content including additions below in this section,

Comment [A10]: VA, MV, CRS

Comment [A11]: VA, CRS, AV

Comment [A12]: VA, CRS, AV Modeling Checking content

Comment [A13]: MV: also under data analysis and vision

Comment [A14]: VA, CRS, AV math content using a variety of tools.

Comment [A15]: MV: also under data analysis

Comment [A16]: MV: also under data analysis

Comment [A17]: MV: from (12)

(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.

(E) create interactive documents using modeling, simulation, and hypertext;

(F) code using various data types;

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

(7) Collaboration. The student uses technology applications and formats digital information 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;

(B) use correct programming style to enhance the readability and functionality of the code such as spacing, descriptive identifiers, comments, or documentation;

(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;

(E) create technology specifications for tasks/evaluation rubrics and demonstrate that products/product quality can be evaluated against established criteria;

(F) annotate coding properly with comments, indentation, and formatting;

(G) write technology specifications for planning/evaluation rubrics documenting variables, prompts, and programming code internally and externally; and

(H) seek and respond to advice from peers and professionals in evaluating the product.

(8) Vision. The student creates a product to deliver electronically in a variety of media. 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.

Comment [A18]: MV: from (10)

Comment [A19]: MV: also under data analysis. And Reasoning. Originally from (7)

Comment [A20]: MV: also under Reasoning. Originally from (7)

Comment [A21]: MV: from (10)

Technology Applications, Fundamentals of Computer Science (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) Introduction to Computer Science is intended as a first course for those students just beginning the computer science studies. There are two goals for the course. First the course is meant to stand alone as a means to teach students about the computing tools that they already use every day and no doubt will use for the rest of their lives. The second goal of this course is to introduce those students that wish to continue their studies of computer science to the problem solving and reasoning skills they will need to be successful in subsequent computer science courses.

(2) The technology applications curriculum has six strands: Vision (Creativity and Innovation), Collaboration (Teamwork and Communication), Data Analysis (Research and Information Fluency), Reasoning (Critical Thinking, Problem Solving, and Decision Making), Integrity (Digital Citizenship), and Principles (Technology Operations, Systems, and Concepts).

(3) Introduction to Computer Science will foster students creativity and innovation by presenting opportunities to design, implement and present meaningful programs through a variety of media. Students will collaborate with one another, their instructor and with 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 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Introduction to Computer Science course. Students will gain and understanding of the principles of computer science through the study of technology operations, systems and concepts.

(c) Knowledge and skills.

(1) Vision (ISTE - Creativity and Innovation). The student will create new ideas and products and share those ideas and products through various media. 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) use HTML code to create useful web pages; and

(G) design creative and effective user interfaces.

(2) Collaboration (ISTE - Communication and Collaboration). The student will collaborate with their peers to create, testand publish new products and knowledge. The student is expected to:

(A) publish information in a variety of ways including, but not limited to, printed copy and via the monitor;

(B) publish information in a variety of ways including but not limited to software, web pages, and video;

(C) seek and respond to advice from peers and professionals in evaluating problem solutions; and

(D) debug and solve problems using reference materials and effective strategies.

(3) Data Analysis (ISTE - Research and Information Fluency). The student will use a variety of strategies to acquire information from electronic resources. The student is expected to:

(A) construct appropriate electronic search strategies in the acquisition of information including keyword and Boolean search strategies; and

(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.

(4) Reasoning (ISTE – Critical Thinking, Problem Solving, and Decision Making). The student uses appropriate productivity tools to create and modify solutions to problems. The student is expected to:

(A) demonstrate the ability to insert applications into web pages using HTML code;

(B) find, download and insert JavaScript code into web pages to enhance interactivity and usefulness; and

(C) recognize and understand the importance of the binary and hexadecimal numbering systems.

(5) Reasoning (ISTE – Critical Thinking, Problem Solving, and Decision Making). The student uses appropriate computer-based productivity tools to create and modify solutions to problems. The student is expected to:

(A) using a visual, drag and drop type programming language design and create programs that create stories, games and animations;

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

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

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

(E) demonstrate an understanding of and use variableswithin a programmed story, game or animation;

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

(G) demonstrate an understanding of and use keyboard inputwithin a programmed story, game or animation;

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

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

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

(6) Integrity (ISTE – Digital Citizenship). The student complies with the laws and examines the issues regarding the safe, legal, and responsible use of technology in society. The student is expected to:

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

(B) demonstrate proper 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;and

(D) understand safety risks associated with the use of social networking sites.

(7) Principles (ISTE – Technology Operations and Concepts). The student will understand the nature, design and operation of the computing systems used to solve technology based problems. The student is expected to:

(A) demonstrate knowledge about a central processing unit (CPU);

(B) compare, contrast, and appropriately use the various input, processing, output, and primary/secondary storage devices (all peripherals) including how to acquire and install the proper device drivers;

(C) demonstrate knowledge and understanding of both wired and wireless connectivity;

(D) demonstrate knowledge and appropriate use of different operating systems;

(E) demonstrate an understanding of the difference between open source and proprietary operating systems;

(F) demonstrate the ability to use the command prompt;

(G) demonstrate knowledge and appropriate use of software applications;

(H) compare and contrast the difference between an application and an operating system;

(I) describe how and application works with an operating system;

(J) demonstrate knowledge and appropriate use of networking components such as cables, switches, servers and routers;

(K) demonstrate knowledge and appropriate use of local area networks (LANs) and wide area networks (WANs);

(L) understand the difference between an intranet and the Internet;

(M) understand and explain the physical infrastructure of the Internet;

(N) demonstrate knowledge and appropriate use of Internet protocols including IP addresses and domain names;

(O) recognize the various web browsers available; and

(P) define what a web browser is and how it works.

Technology Applications, Principles of Computer Science

(a) General requirements. This course is recommended for students in Grades 9-12

(b) Introduction.

(1) Principles of Computer Science is intended as a first course for students interesting in learning about computing and computer science. Explicitly demonstrating the relevance and impact of computer science in today's world, students are expected to utilize critical thinking skills while engaging in the innovative aspects of the field such as developing relevant computational products. The focus is on providing students with the principles, concepts, and methodologies which are required in order to understand how computer science impacts every aspect of our world.

(2) The technology applications curriculum has six strands: I. Vision (Creativity and Innovation), II. Integrity (Digital Citizenship), III. Collaboration (Teamwork and Communication), IV. Data Analysis (Research and Information Fluency), V. Reasoning (Critical Thinking, Problem Solving, and Decision Making), VI. Principles (Technology Operations, Systems, and Concepts).

(3) Principles of Computer Science introduces students to computing as a creative activity. Students learn how computer systems and networks can facilitate communication, collaboration, and problem solving not only in computer science but in other fields including business, science, social science, arts, medicine, engineering, and the humanities. Analyzing the effects of computation, students will use abstractions on multiple levels. Using their own algorithms and programs, data will be processed for information and models developed to create simulations.

(c) Knowledge and skills.

(1) Integrity (Digital citizenship): The student understands legal and ethical issues associated with the use and development of intellectual property, digitalized data / information, and technologies. The student is expected to:

(A) identify and appropriately use digital information including the correct citation of sources;

(B) identify use appropriate methods for citing sources;

(C) identify and discuss intellectual property laws, issues, and use;

(D) identify and discuss legal and ethical issues that result from the creation of technological innovations;

(E) identify, compare, and contrast the significance of technological innovations and how technological innovations can enable as well as be enabled by the creation of new technologies; and

(F) identify, compare, and contrast the effects of technology on individuals, organizations, and communities and how they can differ.

(2) Collaboration (Teamwork and Communication): The student communicates individually and in groups. The student is expected to:

(A) work effectively in teams;

(B) communicate the details of an algorithm using appropriate format; and

(C) communicate processes and results using appropriate model.

(3) Data Analysis (Research and Information Fluency): The student manipulates, stores, and transmits data. The student is expected to:

(A) use programs to manipulate information;

(B) use finite binary sequences to represent data, information, and knowledge;

(C) use abstractions including models to translate information into digital representations; and

(D) identify, compare, and contrast informationrepresentation, storage, and transmission including aspects related to security.

(4) Reasoning (Critical Thinking, Problem Solving, and Decision Making): The student analyzes problems and develops solutions. The student is expected to:

(A) analyze in order to solve problems using multiple levels of abstraction;

(B) analyze algorithms and programs; and

(C) develop a computational solution to a problem by selecting appropriate approach.

(5) Vision (Creativity and Innovation): The student creates abstractions, algorithms, and programs to solve problems. The student is expected to:

(A) create and develop abstractions including models and simulations to analyze and solve problems;

(B) create and develop algorithms using appropriate abstractions and constructs to solve a specific problem; and

(C) create and develop executable programs.

(6) Principles (Technology Operations, Systems, and Concepts): The student is expected to:

(A) describe computer systems composed of input, output, storage, and processing including their relations and interactions;

(B) identify, compare, and contrast design principles used in the development of systems and networks; and

(C) identify, compare, and contrast computational problems by their complexity.

§126.22. Computer Science I (One Credit).

(a) General requirements. ~~The prerequisite for this course is proficiency in the knowledge and skills described in §126.12(c) 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 concurrent enrollment in Geometry 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 (c) of this section, the computer science course descriptions for the College Board Advanced Placement or International Baccalaureate programs, or a combination thereof.~~

(b) Introduction.

(1) Computer Science I is the first course in the one-to-one corresponding course sequence between High School Computer Science and College readiness. A programming language should be selected that is used by all courses in the sequence. This is referred to as the prevailing computer language. This course is mapped with the common course numbering system COSC 1426 Programming Fundamentals I. This course is designed for duel credit and college readiness. Computer science I is an academically challenging course.

(2)~~(1)~~ The technology applications curriculum has ~~four~~ six strands: ~~foundations, information acquisition, work in solving problems, and communication~~ I. Vision (Creativity and Innovation), II. Integrity (Digital Citizenship), III. Collaboration (Teamwork and Communication), IV. Data Analysis (Research and Information Fluency), V. Reasoning (Critical Thinking, Problem Solving, and Decision Making), VI. Principles (Technology Operations, Systems, and Concepts).

~~(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.~~

(3) Introduction to Computer Science will foster students' creativity and innovation by presenting opportunities to design, implement and present meaningful programs through a variety of media. Students will collaborate with one another, their instructor and with 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 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Introduction to Computer Science course. Students will gain and understanding of the principles of computer science through the study of technology operations, systems and concepts.

(c) Knowledge and skills.

(1) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student uses computer technology and a defined programming language to facilitate problem solving though program design. The student is expected to:

(A) use program design problem solving strategies to create program solutions, including:

(i) read and understand a problem's description, purpose, and goals;

(ii) specify the purpose and goals for a problem;

(iii) identify the subtasks needed to solve a problem;

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

(v) identify reusable components from existing code using predefined classes, class libraries, and student generated classes.

(B) use the principles of Software Engineering to design a solution to a problem, code a solution from a program design, identify errors and bugs, and test the solution for correctness.

(C) 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.

(D) use top-down development methodology and implementation techniques, including the use of program stubs in program testing and development.

(E) analyze and modify existing code by evaluating classic algorithms, such as finding the biggest number out of three, making change, and finding the average.

(F) test program solutions by investigating boundary conditions, testing methods in isolation and generating appropriate valid and invalid test data.

(G) implement program debugging strategies including:

(i) identifying, categorizing and correcting compile, syntax, run-time and logic errors;

(ii) program tracing; and

(iii) incorporating temporary output tracking statements.

(H) create program solutions that exhibit robust behavior by understanding and avoiding runtime exceptions, such as division by zero, and handling anticipated errors, such as input type mismatch, using the throws operation.

(I) analyze and identify algorithms by demonstrating effective algorithm selection.

(J)~~(12)(C)~~ debug and solve problems using error messages, reference materials, language documentation, and effective strategies.

(2)~~(3)~~ Integrity (Digital Citizenship) ~~Foundations~~. The student complies with the laws and examines the ethical issues and social issues regarding the use of technology in society. The student is expected to:

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

(B)~~(3)(A)~~ model ethical acquisition and use of digital information ~~, citing sources using established methods~~;

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

(C)~~(B)~~ demonstrate proper 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) compare and contrast the impact of current legislation topersonal privacy and intellectual property to fair use; and

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

(3)~~(1)~~ Principles (Technology Operations, Systems, and Concepts) ~~Foundations~~. The student demonstrates knowledge and appropriate use of major hardware components, system software ~~programs~~, programming languages, and types of computer systems. ~~their connections, and use data input skills appropriate to the task.~~ 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, central processing unit, and peripherals; and

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

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

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

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

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

(4) Collaboration (Teamwork and Communication) ~~Information acquisition~~. The student uses a variety of strategies to acquire and display electronic information in a variety of formats ~~and evaluate, with appropriate supervision~~. The student is expected to:

~~(A) use local area networks (LANs) and wide area networks (WANs), including the Internet and intranet, in research and resource sharing; and;~~

(A) create and properly display meaningful output using both continuous and line feed output statements, such as print and println;

~~(B) construct appropriate electronic search strategies in the acquisition of information including keyword and Boolean search strategies;~~

(B) create interactive console display interfaces, with appropriate user prompts, to acquire data from a user using a library class, such as Scanner();

(C) create interactive human interfaces to acquire data from a user and display program results using a Graphical User Interface (GUI) incorporating widgets, including Buttons, Text Fields, and Text Areas;

(D) properly apply escape sequence characters to improve the meaning of output, including carriage return line feed (\n) and non-printable quotes (\");

(E)~~(9)(B)~~ write programs and communicate with proper programming ~~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~~;

(F)~~(10)(A)~~ annotate coding properly with comments, indentation, and formatting;

(G) improve numeric display by optimizing data visualization using String formatting, using specifiers that control numeric place values;

(H) acquire and process information from text files, of a known fixed size and of a single data type;

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

(J) display simple bit map images using a utility class, such as ImageIcon.

~~(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~~

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

(5) Data Analysis (Research and Information Fluency). The student uses data structures and data analysis to create and modify solutions to problems. The student is expected to:

(A) understand the binary representation of numeric data in computer systems and the finite numerical limits;

(B) understand the binary representation of boolean values, and binary switching;

(C) demonstrate the ability to count in the binary number system;

(D) compute numerical conversions between the decimal and binary number systems;

(E) use the primitive data types int, double, and boolean when writing program solutions;

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

(G) represent text data using a string object data type;

(H) use concatenation to combine strings;

(I) use string library routines to determine length, substrings, and equality;

(J) understand and implement access scope modifiers of public and private;

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

(L) identify and use a list object data structure, such as Arraylist, to traverse, insert, and delete object data; and

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

~~(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)~~(7)~~ Vision (Creativity and Innovation) ~~Solving problems~~. The student uses appropriate ~~computer-based productivity tools~~ programming language statements to create and modify solutions to problems. 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) demonstrate proficiency in the use of the arithmetic operators addition (+), subtraction (-), multiplication (*), real division (/), integer division (/) and modulus division (%);

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

(B)~~(F)~~ develop ~~coding~~ program solutions with correct and efficient use of mathematical expressions and the assignment operator (=); ~~statements including the use of standard/user-defined functions, data structures, operators/proper operator precedence, and sequential/conditional/repetitive control structures;~~

(C) develop sequential ~~and~~ algorithms to decision making problems using branching control statements, including if, if-else, and cascading if-else;

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

(D) demonstrate proficiency in the use of the relational operators equal (= =), not equal (!=), less than (<), less than or equal (<=), greater than (>), greater than or equal (>=);

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

(E) demonstrate proficiency in the use of the logical operators NOT (!), OR(||), and AND (&&);

(F)~~(7)(C)~~ develop iterative algorithms and code~~s~~ programs ~~in prevailing computer languages~~ to solve practical problems using pre-test whileand for loops ~~modeled from school and community~~;

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

(G) write standard algorithms that perform traversal and search operations on a data structure; and

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

(H) use libraries of standardized algorithms to perform operations on data structures, including Arrays.sort and Collections.sort.

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

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

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

(7)~~(2)~~ Principles (Technology Operations, Systems, and Concepts) ~~Foundations~~. The student uses basic programming constructs and fundamental object oriented programming design to write program algorithms and solutions to problems ~~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~~

(A) understand and use packages and library classes by using fully qualified library access, such as import x.y.Z, and global library access, such as import x.y.*;

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

(B) create program solutions to problems using the mathematics library class, including absolute value (abs), round (round), power (pow), square (sqr) and square root (sqrt);

(C) understand object oriented design concepts of attributes, behaviors and instantiation;

(D) create object oriented definitions using class declarations, variable declarations, constant declarations, method declarations, and parameter declarations;

(E) create objects from class definitions through instantiation;

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

(G) write computer programs using sound class design, using:

(i) simple structured (in main) and single class implementations;

(ii) class membership of variables, constants and methods;

(iii) accessors and modifiers to examine and mutate the properties of an object; and

(iv) inheritance to extend, modify and improve reusable code.

(H) create classes that encapsulate data and the methods that operate on that data by using the fundamental components of constructor, toString, accessors, and modifiers;

(I) create void methods with and without the use of arguments and parameters;

(J) create return typed methods with and without the use of arguments and parameters;

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

(L) compare objects using reference values and a comparison routine, such as equal(), including equals (= =) and not equals (!=); and

(M) use inheritance to extend a given class to design, modify and implement subclasses.

~~(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;~~

~~(B) use correct programming style to enhance the readability and functionality of the code such as spacing, descriptive identifiers, comments, or documentation;~~

~~(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:~~

~~(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.

(1) The prerequisite for this course is proficiency in the knowledge and skills for Computer Science I as identified in §126.22(c) of this title (relating to Computer Science I (One Credit)). This course is recommended for students in Grades 10-12. ~~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.~~

(b) Introduction.

(1) Computer Science II is the second course in the 1 to 1 corresponding course sequence between High School Computer Science and College readiness. The prevailing programming language selected for computer science I should be continued.This course is mapped with the common course numbering system COSC 1437 Programming Fundamentals II. This course is designed for duel credit adoption and college readiness. Computer Science II is an academically rigorous and challenging course.

~~(1) The technology applications curriculum has four strands: foundations, information acquisition, work in solving problems, and communication.~~

(2) The technology applications curriculum has six strands: I. Vision (Creativity and Innovation), II. Integrity (Digital Citizenship), III. Collaboration (Teamwork and Communication), IV. Data Analysis (Research and Information Fluency), V. Reasoning (Critical Thinking, Problem Solving, and Decision Making), VI. Principles (Technology Operations, Systems, and Concepts).

~~(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.~~

(3) Introduction to Computer Science will foster students creativity and innovation by presenting opportunities to design, implement and present meaningful programs through a variety of media. Students will collaborate with one another, their instructor and with 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 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Introduction to Computer Science 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) 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)~~(9)~~ Reasoning (Critical Thinking, Problem Solving, and Decision Making) ~~Solving problems~~. The student uses computer technology and a defined programming language to facilitate problem solving though program design, process and product. The student is expected to:

(A) use program design problem solving strategies to create program solutions, including:

(i) identifying the subtasks needed to solve a problem;

(ii) identifying the data types and objects needed to solve a problem;

(iii) appling data abstraction and encapsulation;

(iv) read and understand class specifications and relationships among classes, including IS-A and HAS-A relationships;

(v) decompose a problem into classes, define relationships and responsibilities of those classes; and

(vi) understand and implement a student created class hierarchy.

(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 use a group oral presentation to report the solutions findings;

~~(D) analyze models used in development of software including software life cycle models, design objectives, documentation, and support; and~~

(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) seek and respond to advice from peers and professionals in delineating technological tasks.~~

(E)~~(C)~~ compare and contrast top-down and bottom-up design methodologies and implementation techniques;

(D) use object oriented programming development methodology, encapsulation with information hiding, and procedural abstraction in program development and testing;

(E) analyze, modify, and evaluate existing code by performing a case study on a large program;

(F) use inheritance and black box programming to extend existing code to change the behavior in a case study program;

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

(H) implement program debugging strategies including:

(i) identifying, categorizing and correcting compile, syntax, run-time and logic errors;

(ii) perform program tracing using a debugger; and

(iii) incorporating temporary output tracking statements.

(I) create program solutions that exhibit robust behavior by understanding and avoiding runtime exceptions, and handling anticipated errors by implementing try catch code blocks;

(J) compare and contrast algorithm efficiency by using informal run time 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; and

(K)~~(12)(C)~~ debug and solve problems using error messages, reference materials, language documentation, and effective strategies.

~~(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)~~(3)~~ Integrity (Digital Citizenship) ~~Foundations~~. The student complies with the laws and examines the ethical issues and social issues regarding the use of technology in society. 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 etiquette, responsible use of software, and knowledge of acceptable use policies ~~when using networks, especially resources on the Internet and intranet~~; and

(C) investigate digital rights management (DRM). ~~measures, such as passwords or virus detection/prevention, to protect computer systems and databases from unauthorized use and tampering; and~~

~~(D) code modules for the World Wide Web (WWW) community.~~

(3)~~(1)~~ Principles (Technology Operations, Systems, and Concepts) ~~Foundations~~. The student demonstrates knowledge and appropriate use of major hardware components, system software ~~programs~~, programming languages, and types of computer systems ~~their connections, and use data input skills appropriate to the task~~. The student is expected to:

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

(B) demonstrate knowledge of major hardware components including; primary and secondary memory, 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-host, and networked;

(E) demonstrate knowledge of computer addressing systems, including ip address and MAC address; and

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

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

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

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

(F)~~(F)~~ differentiate among the levels of programming languages including machine, assembly, ~~and~~ high-level compiled, high-level interpreted and scripted. ~~and~~

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

(4) ~~Information acquisition~~ Collaboration (Teamwork and Communication). The student uses a variety of strategies to acquire and display electronic information in a variety of formats ~~and evaluate, with appropriate supervision~~. 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 ~~and resource sharing~~; ~~and~~

(B) create and properly display meaningful output using both continuous and line feed output statements, such as print, println and printf;

(C) ~~construct appropriate electronic search strategies in the acquisition of information including keyword and Boolean search strategies.~~ create interactive console display interfaces, with appropriate user prompts, to acquire data from a user using a library class, such as Scanner();

(D) create interactive human interfaces to acquire data from a user and display program results using a Graphical User Interface (GUI) incorporating widgets, including Buttons, Text Fields, and Text Areas, Check Boxes, List Boxes, Choice Boxes, Radio Buttons, Mouse movement and Mouse selection;

(E) properly apply escape sequence characters to improve the meaning of output, including carriage return line feed (\n), non-printable quotes (\"), non-printable backslash (\\ ), tab (\t), and unicode (\ u n n n n n);

(F)~~(9)(B) use correct programming~~ 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; ~~such as spacing, descriptive identifiers, comments, or documentation~~;

~~(10)(A)(F) annotate coding properly with comments, indentation, and formatting;~~

(G) improve data display by optimizing data visualization using String formatting, using specifiers and associated flags, including. String, character, decimal integer, and floating point;

(H) acquire and process information from text files; of a known fixed size and an unknown size, and single data types and mixed data types;

(I) display simple vector graphics using lines, circles, rectangles, ovals, and polygons;

(J) display simple bit map images using a utility class, such as ImageIcon; and

(K) create, edit, and manipulate bit map images that are used to enhance user interfaces and program functionality.

~~(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~~

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

(5) Data Analysis (Research and Information Fluency). The student uses data structures and data analysis to create and modify solutions to problems. The student is expected to:

(A) demonstrate the ability to count in the binary and hexadecimal number system;

(B) compute numerical conversions between; the decimal and binary number systems, and the decimal and hexadecimal number systems;

(C) use the primitive data types char, int, double, float, and boolean when writing program solutions;

(D) demonstrate knowledge of the maximum integer boundary, minimum integer boundary, imprecision of real number representations, imprecision of floating-point representations, and round-off errors;

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

(F)~~(7)(B)~~ manipulate data structures using string processing using string library routines, including; length, substring, equals, equalsIgnoreCase, charAt, indexOf, lastIndexOf, compareTo, and valueOf;

(G) manipulate data values by casting between primitive data types;

(H) manipulate object reference scope by conversions to super types and casting to subtypes;

(I) understand and implement access scope modifiers of public, private, protected, default, static, final, and abstract;

(J) compare objects using a library method, such as compareTo, and implement a comparable interface for student defined objects;

(K) duplicate objects using the appropriate deep and/or shallow copy;

(L) understand and explain object relationships between defined classes, abstract classes and interfaces;

(M) define and implement abstract classes and interfaces in program problem solutions;

(N) use polymorphism to adjust the behavior of objects;

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

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

(Q) identify and use a list object data structure, such as Arraylist, to traverse, insert, search and delete object data;

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

(S) apply functional decomposition to a program solution.

~~(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)~~(7)~~ Vision (Creativity and Innovation). ~~Solving problems~~. The student uses appropriate ~~computer-based productivity tools~~ programming language statements to create and modify solutions to problems. The student is expected to:

(A) demonstrate proficiency in the use of the arithmetic operators addition (+), subtraction (-), multiplication (*), real division (/), integer division (/) and modulus division (%);

(B) develop program solutions with correct and efficient use of mathematical expressions, the assignment operator (=), and compound assignment operators, including;+=, -=, *=, /=, and %=;

~~(C) use notation for language definition such as syntax diagrams or Backus-Naur forms;~~

(C) demonstrate proficiency in the use of unary operators for incrementing (++) and decrementing (--), including pre and post operations;

(D) develop sequential algorithms using branching control statements, including if, if-else, cascading if-else, and nested if-else structures to create solutions to decision making problems;

~~(E) create robust programs with increased emphasis on design, style, clarity of expression and documentation for ease of maintenance, program expansion, reliability, and validity;~~

(E) develop choice algorithms using selection control statements, including switch, case and break;

~~(F) apply methods for computing iterative approximations and statistical algorithms;~~

(F) demonstrate proficiency in the use of the relational operators equal (= =), not equal (!=), less than (<), less than or equal (<=), greater than (>), greater than or equal (>=);

~~(G) define and develop code using the concepts of abstract data types including stacks, queues, linked lists, trees, graphs, and information hiding;~~

(G) demonstrate proficiency in the use of the logical operators NOT (!), OR(||), AND (&&), and XOR (^);

(H) demonstrate proficiency in the use short circuiting and De Morgan's Law;

~~(I) develop software to solve a school or community problem such as customer relations, design, modular programming, documentation, validation, marketing, or support; and~~

(I) develop iterative algorithms using while loops, for loops, enhanced for loops and nested loops;

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

(J)~~(7)(A)~~ identify, trace, and use recursion appropriately ~~and trace~~ in algebraic computations and in programming solutions ~~design comparing invariant, iterative, and recursive algorithms~~;

~~(7)(A) construct search algorithms including linear and binary searches; and~~

(K)~~(C)~~ design, ~~and document,~~ construct, evaluate, and compare standard ~~sequential~~ linear search algorithms that perform traversal and search operations on a data structure, including random searching, linear searching and binary searching ~~for digital information storage and retrieval.~~;

(L)~~(7)(D)~~ identify, describe, design, create, evaluate, and compare standard sorting algorithms that perform sorting operations on data structures, including; random sort, selection sort, bubble sort, insertion sort, and merge sort;

(M) ~~and~~ use sequential and non-sequential files~~; multidimensional arrays and arrays of records;~~ and ~~quadratic~~ sorting algorithms; ~~such as~~ including selection, bubble, ~~or~~ insertion, and ~~more efficient algorithms including~~ merge~~, shell, and quick~~ sorts to measure time/space efficiency;

(N)~~(7)(B)~~ compare and contrast search and sort algorithms including linear, quadratic, and recursive strategies ~~and binary searches for different purposes and~~ for time/space efficiency;and ~~search time.~~

(O)~~(9)(B)~~ analyze algorithms using "big-O" notation, for best, average, and worst case data patterns. ~~space techniques;~~

(7)~~(2)~~ Principles (Technology Operations, Systems, and Concepts) ~~Foundations Principles~~. The student uses basic programming constructs and fundamental object oriented programming design to write program algorithms and solutions to problems ~~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~~

(A) create program solutions to problems using the mathematics library class, including absolute value (abs), round (round), power (pow), square (sqr), square root (sqrt), minimum (min), maximum (max), ceiling (ceil), and floor (floor);

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

(B) use the mathematics library class random algorithm to create simulations that mirror the environment;

(C) use the mathematics library class standard constants of PI and E;

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

(E) understand programming file structure and file access; including class path, packages and projects;

(F) create object oriented definitions using class declarations, variable declarations, constant declarations, method declarations, parameter, and interface declarations;

(G) create simple and robust objects from class definitions through instantiation;

~~(H) identify and describe the correctness and complexity of algorithms such as divide and conquer, backtracking, or greedy algorithms;~~

(H) write computer programs using sound class design, using:

(i) simple structured (in main) and single class implementations;

(ii) class membership of variables, constants and methods;

(iii) accessors and modifiers to examine and mutate the properties of an object;

(iv) multiples class design;

(v) inheritance to extend, modify and improve existing code; and

(vi) polymorphism to create adaptive behaviors;

(H) create robust classes that encapsulate data and the methods that operate on that data and incorporate overloading to enrich the objects behavior:

(I) design and implement a set of interactive classes;

(J) compare objects using reference values and a comparison routine, such as equal(), including equals (= =) and not equals (!=);

(K) use wrapper classes to provide object functionality to primitive data types, including; Character, Integer, Double, and Graphics;

(L) understand and implement the “IS-A” class design of encapsulating composite data;

(M) understand and implement the “HAS-A” class design by extending class through inheritance;

(N) use inheritance to extend a given class to design, modify and implement subclasses;

(O) understand and evaluate the concept of abstract classes and interfaces;

(P) use the object reference identifiers null, this and super;

(Q) create multi-class programs that utilize abstract classes and interfaces;

(R) design and implement an interface;

(S) create multi-class programs that utilize polymorphism; and

(T) use various electronic media for data acquisition, including text, jpg, and gif.

~~(8) Solving problems. Vision. 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.~~

~~(10) Communication. The student formats digital information for appropriate and effective communication. The student is expected to:~~

~~(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~~

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

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

Computer Science III (One Credit).

(a) General requirements.

(1) 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 I (One Credit)) or proficiency in the knowledge and skills for Computer Science (AP) xxx. This course is recommended for students in Grades 11-12.

(b) Introduction.

(1) Computer Science III is the third course in the 1 to 1 corresponding course sequence between High School Computer Science and College readiness. The prevailing programming language selected for Computer Science I and II should be continued. This course is mapped with the common course numbering system COSC 2415 Programming Data Structures. This course is designed for duel credit adoption and college readiness. Computer Science III is an academically rigorous and challenging course.

(2) The technology applications curriculum has six strands: I. Vision (Creativity and Innovation), II. Integrity (Digital Citizenship), III. Collaboration (Teamwork and Communication), IV. Data Analysis (Research and Information Fluency), V. Reasoning (Critical Thinking, Problem Solving, and Decision Making), VI. Principles (Technology Operations, Systems, and Concepts).

(3) Computer Science III will foster students creativity and innovation by presenting opportunities to design, implement and present meaningful programs through a variety of media. Students will collaborate with one another, their instructor and with 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 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Introduction to Computer Science 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) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student uses computer technology and a defined programming language to facilitate problem solving though program design, process and product. The student is expected to:

(A) use program design problem solving strategies to create program solutions, including:

(i) appling data abstraction and encapsulation;

(ii) reading and writing class specifications;

(ii) decompose a problem into classes, and

(iv) implement a student created class hierarchy.

(B) 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 use a group oral presentation to report the solutions 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) use object oriented programming development methodology, encapsulation with information hiding, and procedural abstraction in program development and testing;

(F) use inheritance and black box programming to extend existing code to change the behavior in a case study program;

(G) create program solutions that exhibit robust behavior by understanding and avoiding runtime exceptions, handling anticipated errors by implementing try catch code blocks, and finally/throws; and

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

(2) Integrity (Digital Citizenship). The student complies with the laws and examines the ethical issues and social issues regarding the use of technology in society. The student is expected to:

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

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

(3) Principles (Technology Operations, Systems, and Concepts). The student demonstrates knowledge and appropriate use of major hardware components, system software, programming languages, and types of computer systems. The student is expected to:

(A) compare and contrast high-level programming languages;

(B) create a small workgroup network;

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

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

(4) Collaboration (Teamwork and Communication). The student uses a variety of strategies to acquire and display electronic information in a variety of formats. 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) incorporating widgets, including Buttons, Text Fields, and Text Areas, Check Boxes, List Boxes, Choice Boxes, Radio Buttons, Mouse movement, Mouse selection, menu bars, scroll bars, and scroll panes;

(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;

(D) write program solutions using Assertions;

(E) write program solutions using Invariants;

(F) read, write, and modify information from text files; and

(G) understand end of file (eof) and file readiness.

(5) Data Analysis (Research and Information Fluency). The student uses data structures and data analysis to create and modify solutions to problems. The student is expected to:

(A) demonstrate the ability to count in the binary, octal and hexadecimal number system;

(B) compute numerical conversions between all numbering systems using radix;

(C) use the primitive data types char, int, double, float, Boolean, byte, short, and long when writing program solutions;

(D) understand and implement access scope modifiers of public, private, protected, default, static, final, and abstract;

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

(F) use polymorphism to adjust the behavior of objects;

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

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

(I) identify and use a list object data structure, such as Vector, to traverse, insert, search and delete object data;

(J) understand and trace the Linked list data structure;

(K) create program solutions using the linked list data structure, including: singled (unordered), single (ordered), double, and circular;

(L) understand the data structure array of linked list;

(M) understand the data structure linked list of linked list;

(N) understand and create program solutions using Stacks;

(O) understand and create program solutions using Queues;

(P) understand and create program solutions using Trees;

(Q) understand and create program solutions using Heaps;

(R) understand and create program solutions using Priority Queues;

(S) understand and create program solutions using Sets, including HashSet and TreeSet;

(T) understand and create program solutions using Maps, including HashMap and TreeMap;

(U) understand and create program solutions using String Buffer, including defined, setCharAt, reverse, append, insert, and delete;

(V) understand and create program solutions using Graph Theory;

(W) understand and create program solutions using Enumerated data types;

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

(Y) apply functional decomposition to a program solution.

(6) Vision (Creativity and Innovation). The student uses appropriate programming language statements to create and modify solutions to problems. The student is expected to:

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

(B) demonstrate proficiency in the use of the ternary operator (? :);

(C) demonstrate proficiency in the use of the logical operators NOT (!), OR(||), AND (&&), and XOR (^);

(D) demonstrate proficiency in the use of the bitwise operators, OR(|), AND (&), left shift (<<), right shift (>>) and OR (^);

(E) demonstrate proficiency in boolean algebra and the use of De Morgan's Law;

(F) develop iterative algorithms using while loops, for loops, enhanced for loops, nested loops, and do-while loops;

(G) create program solutions that us iterators;

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

(I) understand and create program solutions using hashing;

(J) use regular expressions to perform pattern recognition;

(L) identify, describe, design, create, evaluate, and compare standard sorting algorithms that perform sorting operations on data structures, including; random sort, selection sort, bubble sort, insertion sort, merge sort, quick sort, and heap sort; and

(M) analyze algorithms using "big-O" notation, for best, average, and worst case data patterns.

(7) Principles (Technology Operations, Systems, and Concepts). The student uses basic programming constructs and fundamental object oriented programming design to write program algorithms and solutions to problems.The student is expected to:

(A) use the mathematics library class random algorithm to create simulations that mirror the environment;

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

(C) understand and create jar programming file structures;

(D) create simple and robust objects from class definitions;

(E) write computer programs using sound class design, using:

(i) simple structured (in main) and single class implementations;

(ii) class membership of variables, constants and methods;

(iii) accessors and modifiers to examine and mutate the properties of an object;

(iv) multiples class design;

(v) inheritance to extend, modify and improve existing code; and

(vi) polymorphism to create adaptive behaviors;

(F) create robust classes that encapsulate data and the methods that operate on that data and incorporate overloading to enrich the objects behavior:

(G) design and implement a set of interactive classes;

(H) understand and implement the "IS-A" class design of encapsulating composite data;

(I) understand and implement the "HAS-A" class design by extending class through inheritance;

(J) use inheritance to extend a given class to design, modify and implement subclasses;

(K) understand and evaluate the concept of abstract classes and interfaces;

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

(Q) create multi-class programs that utilize abstract classes and interfaces;

(R) design and implement an interface;

(S) create multi-class programs that utilize polymorphism; and

(T) use various electronic media for data acquisition, including text, jpg, and gif.

Technology Applications, Data Mining (1/2 to 2 credits)

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

(b) Introduction.

(1) Data mining and data warehousing are information technologies that are in growing demand in a range of industries. Data Mining will introduce students to techniques used to obtain information by extracting patterns from data which are used by for a variety of purposes such as marketing, scientific discovery, gaming, and profiling.

(2) The technology applications curriculum has six strands: I. Vision (Creativity and Innovation), II. Integrity (Digital Citizenship), III. Collaboration (Teamwork and Communication), IV. Data Analysis (Research and Information Fluency), V. Reasoning (Critical Thinking, Problem Solving, and Decision Making), VI. Principles (Technology Operations, Systems, and Concepts).

(3) Data Mining will provide students with an overall understanding of these data mining and data warehousing technologies. Data Mining will expand upon traditional database concepts to include the concepts, design methods, physical implementations, and applications that form the foundations of data mining.

(c) Knowledge and skills.

(1) Vision (Creativity and Innovation). The student will design a database to fulfill desired need. The student is expected to:

(A) design database to support an organization's decision making;

(B) design database that optimizes storage and delivery of data; and

(C) design and describe steps required for construction of data warehouse.

(2) Integrity (Digital citizenship): The student understands legal and ethical concerns associated with intellectual property, digitalized data and information, and networks. The student is expected to:

(A) Identify and appropriately use digital information including the correct citation of sources;

(B) identify use appropriate methods for citing sources;

(C) identify and discuss intellectual property laws, issues, and use;

(D) identify and discuss privacy and security issues related to collection and processing of data related to data mining; and

(E) identify and discuss social impact of data mining.

(3) Collaboration (Teamwork and Communication): The student communicates about the characteristics of a data management system. The student is expected to:

(A) explain what is meant when describing the characteristics and behaviors of a given data set; and

(B) justify the appropriateness and describe the impact of selecting a given data set or subset to solve a problem or complete a task.

(4) Collaboration (Teamwork and Communication): The student works effectively in teams to efficiently perform tasks. The student is expected to:

(A) apply effective teamwork practices;

(B) collaborate with multiple participants;

(C) document use, functionality, and implementation;

(D) seek and respond to advice from peers and professionals in delineating technological tasks; and

(E) resolve information conflicts and validate information through accessing, researching, and comparing.

(5) Principles (Technology Operations, Systems, and Concepts). The student understands terminology associated with basic database concepts and how they function. The student is expected to:

(A) demonstrate knowledge, appropriate application, and use of basic database management system operations;

(B) compare and contrast relational databases, data warehouses, and transactional databases, the data associated with each as well as their use;

(C) demonstrate knowledge, appropriate application, and use of normalization theory;

(D) demonstrate knowledge, appropriate application, and use of relational data modeling;

(E) demonstrate knowledge, appropriate application, and use of normalization theory;

(F) demonstrate knowledge, appropriate application, and use of query design techniques; and

(G) compare and contrast a general database and a data warehouse.

(6) Principles (Technology Operations, Systems, and Concepts). The student understands the terminology and concepts related to data mining. The student is expected to:

(A) define data mining data mining and describe its purpose;

(B) define patterns as they relate to data mining and describe their purpose;

(C) compare and contrast patterns that are “interesting” and not interesting;

(D) describe how concept/class descriptions can be used to characterize or discriminate data;

(E) define frequent patterns and describe possible uses;

(F) define what is meant by classification and how classification is used for prediction; and

(G) compare and contrast data mining systems classifications.

(7) Principles (Technology Operations, Systems, and Concepts). The student understands the terminology and concepts related to data mining. The student is expected to:

(A) define multidimensional data model and describe purpose and construction;

(B) describe architecture of data warehouse and how it is implemented; and

(C) describe relation of data warehouse to data mining;

(8) Reasoning (Critical Thinking, Problem Solving, and Decision Making): The student applies data mining to solve problems. The student is expected to:

(A) applies appropriate data mining processes to analyze a variety of complex data objects;

(B) applies appropriate data mining processes to solve problems in a variety of industries; and

(C) identify trends associated with data mining including future applications.

(9) Data Analysis (Research and Information Fluency): The student understands the purpose of data pre-processing and associated methods. The student is expected to:

(A) measure central tendency and dispersion of data;

(B) identify and explain descriptive data summaries and graphic displays;

(C) describe and explain data cleaning as a process including procedures for handling missing values and noisy data;

(D) use data cleaning methods appropriately;

(E) describe and use appropriately data integration and transformation;

(F) describe and explain data reduction and its purpose; and

(G) describe and explain data discretization and concept hierarchy generation.

(10) Data Analysis (Research and Information Fluency): The student understands use of data cubes, data streams, and online analytical processing (OLAP). The student is expected to:

(A) define online analytical processing (OLAP) and data cubes and describe relation;

(B) compare and contrast OLAP and online transactional process (OLTP);

(C) select appropriate strategy for efficient computation of data cube;

(D) describe process of analysis using data cubes; and

(E) compare and contrast methodologies for processing stream data.

(11) Data Analysis (Research and Information Fluency): The student understands how to use classification and clustering algorithms to perform data mining activities. The student is expected to:

(A) describe and appropriately use classification strategies including Decision Tree Induction, Bayesian, and Associative Rule Analysis;

(B) describe and appropriately use prediction using regression-based methods;

(C) describe and appropriately use clustering including outlier analysis; and

(D) describe and appropriately use strategies to evaluate accuracy and measure error.

Technology Applications, Digital Forensics (1/2 to 1 credit)

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

(b) Introduction.

(1) Digital Forensics can be used for many purposes such as investigating crimes and internal policy violations, reconstructing computer security incidents, troubleshooting operational problems, and recovering from accidental system damage. Most organizations could benefit from the ability to perform digital forensics and the purpose of this course is to help students develop such a capability by learning forensic techniques to assist with computer security incident responses as well as other situations.

(2) The technology applications curriculum has six strands: I. Vision (Creativity and Innovation), II. Integrity (Digital Citizenship), III. Collaboration (Teamwork and Communication), IV. Data Analysis (Research and Information Fluency), V. Reasoning (Critical Thinking, Problem Solving, and Decision Making), VI. Principles (Technology Operations, Systems, and Concepts).

(3) Digital forensics, also known as computer and network forensics, is the application of science to the law. The computing and network systems may be standard computer systems, networking equipment, computing peripherals, personal digital assistants (PDAs), consumer electronic devices, various types of media, or other any other source that transmits or stores electronic data. 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.

(c) Knowledge and skills.

(1) Vision (Creativity and Innovation). The student will understand how to establish and organize a digital forensics capability. 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; and

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

(2) Integrity (Digital citizenship): The student understands legal and ethical concerns associated with intellectual property, digitalized data and information, and networks. The student is expected to:

(A) identify and appropriately use digital information including the correct citation of sources;

(B) identify use appropriate methods for citing sources;

(C) identify and discuss intellectual property laws, issues, and use; and

(D) identify intellectual property stakeholders, their needs, and perspectives within a given technological context.

(3) Integrity (Digital citizenship): The student understands legal and ethical concerns associated with digital forensics. The student is expected to:

(A) identify and describe the kinds of crimes investigated by digital forensics specialist;

(B) identify what aspects of information gathering can make the practice legal or illegal, ethical or unethical;

(C) compare and contrast information gathering methods that are legal or illegal, ethical or unethical and identify possible "gray areas";

(D) identify and describe ways in which technological changes affect applicable laws; and

(E) identify and describe ways in which developing laws and guidelines affect digital forensics practices.

(4) Collaboration (Teamwork and Communication): The student communicates about the characteristics of a system. The student is expected to:

(A) explain what is meant when describing the characteristics and behaviors of a given system including those of networks;

(B) describe the characteristics and behaviors of a given system (potential and extent) including those of networks; and

(C) justify the appropriateness and describe the impact of selecting a given system to solve a problem or complete a task including effects (beneficial and harmful).

(5) Collaboration (Teamwork and Communication): The student works effectively in teams to efficiently perform tasks. The student is expected to:

(A) apply effective teamwork practices;

(B) collaborate with multiple participants;

(C) document use, functionality, and implementation;

(D) seek and respond to advice from peers and professionals in delineating technological tasks; and

(E) resolve information conflicts and validate information through accessing, researching, and comparing.

6) Principles (Technology Operations, Systems, and Concepts). The student understands terminology associated with computing and network devices and networks and how they function. The student is expected to:

(A) demonstrate knowledge and appropriate use of operating systems, software applications, and communication and networking components;

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

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

(D) demonstrate knowledge of data formats; and

(E) demonstrate knowledge of networks including intranets and Internet and how they function.

(7) Reasoning (Critical Thinking, Problem Solving, and Decision Making): The student selects and demonstrates appropriate use of digital forensic processes to collect data. The student is expected to:

(A) identify possible sources of data;

(B) acquire data;

(C) describe considerations required for incident response; and

(D) examine, analyze, and report data collected.

(8) Data Analysis (Research and Information Fluency): The student understands how to obtain and use data from data files. The student is expected to:

(A) describe file basics including file storage media, file systems, and other data on media;

(B) collect files by copying files from media while maintaining data file integrity;

(C) describe how files can be modified, access and creation times, and other technical issues; and

(D) examine data files by locating files, extracting data, and using digital forensic toolkit.

(9) Data Analysis (Research and Information Fluency): The student understands how to obtain and use data from operating systems. The student is expected to:

(A) describe operating system basics including terminology and functions;

(B) compare and contrast non-volatile data and volatile data;

(C) collect volatile and non-volatile operating system data;

(D) describe technical issues related to collecting operating system data; and

(E) examine and analyze operating system data.

(10) Data Analysis (Research and Information Fluency): The student understands how to obtain and use data from network traffic. The student is expected to:

(A) describe TCP/IP basics including application, transport, IP, and hardware layers the layers' significance to digital network forensics;

(B) use traffic data sources including firewalls and routers, packet sniffers and protocol analyzers, intrusion detection systems, remote access, security event management software, network forensic analysis tools, as well as other sources;

(C) identify and describe legal considerations and technical issues related to collecting network traffic data; and

(D) examine and analyze network traffic data in order to be able to identify an event of interest, examine data sources, , identify attacker, as well as draw other conclusions.

(11) Data Analysis (Research and Information Fluency): The student understands how to obtain and use data from applications. The student is expected to:

(A) describe the function and use of application components including configurations settings, authentications, logs, application data, supporting files, application architecture;

(B) 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; and

(C) collect, examine, and analyze application data.

(12) Data Analysis (Research and Information Fluency): The student understands how to obtain and use data from multiple sources. The student is expected to:

(A) provide examples of how multiple data sources can be used during digital forensics such as when investigations suspected network service worm infection or investigating threatening e-mail; and

(B) collect, examine, and analyze data from multiple sources.

Technology Applications, Discrete Mathematics (1 to 2 credits)

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

(b) Introduction.

(1) 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.

(2) The technology applications curriculum has six strands: I. Vision (Creativity and Innovation), II. Integrity (Digital Citizenship), III. Collaboration (Teamwork and Communication), IV. Data Analysis (Research and Information Fluency), V. Reasoning (Critical Thinking, Problem Solving, and Decision Making), VI. Principles (Technology Operations, Systems, and Concepts).

(3) Discrete Mathematics 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 in order that the applications can 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 so that they will 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) Principles (Technology Operations, Systems, and Concepts): The student understands the discrete data structures: functions, relations, and sets and utilizes them to analyze problems and apply basic principles. The student is expected to:

(A) explain basic terminology of functions, relations, and sets;

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

(C) analyze practical examples using appropriate models of sets, functions, or relations;

(D) interpret associated operations and terminology in context; and

(E) apply basic counting principles including cardinality and the pigeonhole principle.

(2) Principles (Technology Operations, Systems, and Concepts): The student understands and appropriately uses propositional logic. The student is expected to:

(A) convert spoken language statements to appropriate statements in propositional logic;

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

(C) apply appropriate precedence when using logical operators;

(D) define and provide examples of logical equivalence, normal forms, validity, modus ponens/modus tolens;

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

(F) use truth tables to demonstrate propositional relations;

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

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

(I) Identify and use appropriately use predicates, quantifiers (existential and universal), and valid arguments.

(3) Principles (Technology Operations, Systems, and Concepts): The student understands and uses appropriately the formal tools, methods, and reasoning of symbolic propositional and predicate logic. The student is expected to:

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

(B) model algorithms and real-world situations using formal tools of symbolic logic;;

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

(D) compare and contrast the appropriate use and limitations of predicate logic.

(4) Principles (Technology Operations, Systems, and Concepts): The student understands and uses appropriately proof techniques. The student is expected to:

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

(B) compare and contrast the type of problem best satisfied by each of the following types of proofs: direct, indirect, contradiction, and induction, existence and constructive proofs;

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

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

(E) Identify possible applications such as evaluating algorithmic complexity.

(5) Principles (Technology Operations, Systems, and Concepts): The student understands and appropriately uses basic and advanced counting techniques. The student is expected to:

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

(B) compute permutations and combinations of a set;

(C) use the context of a particular application to interpret the meaning derived when computing the permutations and combinations of a set;

(D) state the definition of the Master theorem;

(E) solve a variety of basic recurrence equations;

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

(G) use counting techniques to analyze algorithms and other practical problems.

(6) Principles (Technology Operations, Systems, and Concepts): The student understands and appropriately uses discrete probability concepts. The student is expected to:

(A) calculate the probabilities of events and expectations of random variables for such problems as games of chance;

(B) compare and contrast dependent and independent events;

(C) apply the binomial theorem to independent events;

(D) apply Bayes theorem to dependent events; and

(E) apply probability tools to solve problems.

(7) Principles (Technology Operations, Systems, and Concepts): The student understands and appropriately applies concepts from graph theory. The student is expected to:

(A) define, compare, and contrastsimple graphs, multigraphs, and directed and undirected graphs using definitions, properties, and examples (including special cases);

(B) demonstrate transversal methods for trees and graphs;

(C) model computer science problems by using graphs and trees; and

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

Technology Applications, Game Programming (One Credit or two Credits).

(a) General requirements. The prerequisite for this course is completion of a high school computer science course as identified in this subchapter. This course may be taken at Grades 10-12.

(b) Introduction.

(1) Game Programming is designed for a student to explore the world of computer games. Upon completion of the course the student should have created a computer game that is presented to an evaluation panel.

(2) The technology applications curriculum has six strands: six strands: creativity and innovation, communication and collaboration , research and information fluency, critical thinking, problem solving, and decision making, digital citizenship, technology operations and concepts

(3) 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, their instructor and with 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 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Game Programming course.

(c) Knowledge and skills.

(1) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student uses appropriate computer-based productivity tools to create and modify solutions to problems and uses research skills and electronic communication, with appropriate supervision, to create new knowledge. The student is expected to:

(A) select and utilize appropriate productivity tools including, but not limited to, IDE, debugger, network, telecommunication, multimedia tools, and utility programs in the creation of games;

(B) evaluate game rules, instructions and licensing agreements;

(C) understand the user experience by comparing rules and psychology patterns;

(D) write game rules, instructions and licensing documents;

(E) create a design concept document;

(F) read and use technical documentation;

(G) develop game products, including; software, marketing and packaging; and

(H) create a story board.

(2) Integrity (Digital Citizenship). The student complies with the laws and examines the issues regarding the use of technology in society. The student is expected to:

(A) discuss 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, responsible use of software, and knowledge of acceptable use policies when using networks, especially resources on the Internet and intranet;

(D) model respect of intellectual property when manipulating, morphing, or editing graphics, video, text, and sound;

(E) demonstrate best practices in understanding and applying Information Security;

(F) discuss and evaluate the social issues in gaming; and

(G) evaluate game design fundamentals including, rationale for games, and types of games.

(3) Collaboration (Teamwork and Communication). The student uses technology applications to facilitate evaluation of communication, both process and product. The student is expected to:

(A) design and implement procedures to track trends, set timelines, and review and evaluate the product using technology tools;

(B) determine and employ technology specifications to evaluate projects for design, content delivery, purpose, and audience, demonstrating that process and product can be evaluated using established criteria or rubrics;

(C) seek and respond to input from peers and professionals in evaluating the product;

(D) make necessary revisions and/or proceed to the next stage of study;

(E) demonstrate knowledge and appropriate use of operating systems, software applications, communication and networking components;

(F) collaborate to research and understand the business of games, including; developer, marketing, publisher, and retail sales;

(G) create games that connect with others using client server architecture across a network;

(H) understand online technology and evaluate online interaction and massively multiplayer games; and

(I) understand the use of collaboration, cooperation, and emergent behaviors in Multiplayer game design.

(4) Data Analysis (Research and Information Fluency).The student utilizes electronic resources and print media to research, practice and test language content, with appropriate supervision. The student is expected to:

(A) identify, create, acquire, and use a variety of electronic file formats including but not limited to text, image, and audio files;

(B) identify and employ a method to evaluate the design, functionality, and accuracy of the computer game;

(C) play board games, analyze, collect data, evaluate and write-up an analysis and evaluation of the games;

(D) research the dramatic elements in games, including; kinds of fun, player types, and nonlinear storytelling; and

(E) implement a tile-based data structure, including; loading tile maps, draw tile maps, rendering a tile map, layering drawing sprites, and parallax scrolling.

(5) Vision (Creativity and Innovation). The student will create new ideas and products and share those ideas and products. The student is expected to:

(A) understand the fundamentals of game art, including; look and feel, graphics coordinate system, basics of color, device context, and color palettes;

(B) understand graphics images, including; bitmap images, reading images, manipulating multiple bitmaps, pixel color, and bit depth;

(C) design and create bitmap images;

(D) understanding and create backgrounds, such as; solid, image, animated, tiled, and scrolling;

(E) understand image modeling and texture, including; polygonal, procedural, polygon reduction, and level of detail;

(F) design, create and implement graphics textures, such as; seamless, alternating, transition, and multi-layer;

(G) understand image rendering, including; transparency, file formats, refresh rate, hardware acceleration, animation, and active rendering;

(H) design rendering plans using visibility, lighting, texturing and shading;

(I) create image special effects, such as; fog, particle systems, shadows, lens effects, and shaders;

(J) understand and create game audio sound effects and music, including; creating and editing sounds from a file and resource, finding sound and music on the internet, implementing sound mechanics; playing, stopping and looping sound;

(K) use local area networks and wide area networks to acquire, organize, maintain and evaluate information;

(L) receive, listen, attend to, interpret, and respond to verbal messages and other cues;

(M) use efficient learning techniques to acquire and apply new knowledge, enrichment curricular content and skills; and

(N) exert a high level of effort and persevere towards product development in the creation of computer game.

(6) Principles (Technology Operations, Systems, and Concepts). The student creates, delivers and presents a computer game product, with appropriate supervision. The student is expected to:

(A) understand the basic game design elements, including; conceptual ideas, storyline, visualization and story board, game effects, sound elements, game play, game controls, player tutorial and game modes;

(B) identify the components and game structure, including; major subsystems, data structures, models, design schemes and interfaces;

(C) understand the game design process, including; idea generation, brainstorming, and paper prototyping;

(D) develop an appropriate data model and database scheme;

(E) understand and apply object oriented design and programming to games;

(F) understand game programming essentials, including; event-driven programming, communicating with messages, device management, information and hardware resources;

(G) understanding the role of a game engine, game events, the animation loop and game timing;

(H) design and create game engines;

(I) understand double buffing, including; flicker free, page flipping, monitor refresh and tearing;

(J) create game user interfaces, including; screen design and layout, visual controls, UI, menus and options;

(K) use game control design to understand, access and control the input devices keyboard, mouse and joystick;

(L) understand and apply animation to a game, including; principles of animation, frame rate, frame-based animation, and cast-based animation;

(M) understand the animation concepts of key-frame, motion capture, and procedural;

(N) understand flow theory, including; decision-making and types of decisions;

(O) understand event triggers, including; touch and release notification, listeners, timed events, scripting, and delayed events;

(P) understand and implement collision detection, including; bounding boxes and sprite collisions;

(Q) implement collision handling, incorporating game geometry, collision testing, additive geometry, and area clipping;

(R) create games using sprites, evaluating the role of sprites, creating sprites, and managing sprites;

(S) understanding, develop and implement Artificial Intelligence (AI), including; path finding, roaming, behavioral, strategic, dodging, attacking, avoiding, targeting, and firing.;

(T) understand game balance and tuning, including; levels of AI, smoke, mirrors and cheating;

(U) implement game evolution techniques, including; regeneration, evolving, upgrades and enhancements;

(V) understand player progression, such as; levels, modeling, linear progression, storing and retrieving high score data;

(W) write the game code, perform unit testing, integrate subsystems, resolve defects and revise existing code;

(X) testing the finished product, using testing techniques, such as; designer testing, player testing, solo testing, team testing, critical analysis, and blind testing;

(Y) perform analysis by defining system and software requirements and developing test requirements; and

(Z) implement game persistence by using game saving techniques.

Technology Applications, Graphics Programming (One-half, One Credit, or One and One Half Credits).

(a) General requirements. The prerequisite for this course is completion of a high school computer science course as identified in this subchapter. This course may be taken at Grades 10-12.

(b) Introduction.

(1) Graphics Programming is designed for a student to explore the world of computer graphics. Graphics programming is used in geometry to study ways to represent and process surfaces. Graphics programming is used in animation to study the ways to represent and manipulate motion. Graphics programming is used in rendering to study algorithms that reproduce light transport. Graphics programming is used in imaging to study image acquisition or image editing design and formats. Upon completion of the course the student should have created computer generated graphics designs and presented the graphics designs to an evaluation panel.

(2) The technology applications curriculum has six strands: six strands: creativity and innovation, communication and collaboration , research and information fluency, critical thinking, problem solving, and decision making, digital citizenship, technology operations and concepts

(3) Computer Graphics can be defined as representation and modification of pictorial data using a computer. Computer Graphics lays emphasis on the mathematical and computational implications of image generation and processing. Computer Graphics Programming will foster student creativity and innovation by presenting opportunities to design, implement and present meaningful computer generated graphics designs through a variety of media. Students will collaborate with one another, their instructor and with various electronic communities to create graphics projects 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 create projects. By using computer science knowledge and skills 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Graphics Programming course.

(c) Knowledge and skills.

(1) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student uses appropriate computer-based productivity tools to create and modify solutions to problems and uses research skills and electronic communication, with appropriate supervision, to create new knowledge. The student is expected to:

(A) select and utilize appropriate productivity tools including, but not limited to, IDE, network, multimedia tools, and utility programs in the creation of graphics programs;

(B) read and use technical documentation;

(C) develop computer created graphics program products that meet standards identified by the selected profession;

(D) understand object transformations, including; modeling transformations and hierarchical modeling;

(E) apply matrix arithmetic when performing transformations, including; scalar multiplication, matrix addition, matrix subtraction, and matrix multiplication;

(F) implement 2D image blit transformations in project creations, including; color fills, scaling, mirroring, and raster operations;

(G) implement 2D geometrical transformations in project creations, including; translation, scaling, rotation, homogeneous coordinates, and concatenation; and

(H) implement 3D geometrical transformations in project creations, including; 3D translation, 3D scaling, 3D rotation, and rotation about an arbitrary axis.

(2) Integrity (Digital Citizenship). The student complies with the laws and examines the issues regarding the use of technology in society. The student is expected to:

(A) discuss 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, responsible use of software, and knowledge of acceptable use policies when using networks, especially resources on the Internet and intranet;

(D) model respect of intellectual property when manipulating, morphing, or editing graphics, video, text, and sound;

(E) demonstrate best practices in understanding and applying Information Security;

(F) investigate how computer generated graphics are being used in our legal system to apply data and simulate events; and

(G) discuss and evaluate computer graphics and the impact of computer generated imagery on society.

(3) Collaboration (Teamwork and Communication). The student uses technology applications to facilitate evaluation of communication, both process and product. The student is expected to:

(A) design and implement procedures to track trends, set timelines, and review and evaluate the product using technology tools;

(B) determine and employ technology specifications to evaluate projects for design, content delivery, purpose, and audience, demonstrating that process and product can be evaluated using established criteria or rubrics;

(C) seek and respond to input from peers and professionals in evaluating the product;

(D) make necessary revisions and/or proceed to the next stage of study;

(E) demonstrate knowledge and appropriate use of operating systems, software applications, communication and networking components;

(F) research and discuss computer animation, including; cartoon animation, real-time animation, frame-by-frame animation, and interactive animation;

(G) compare and contrast the various animation techniques, including; real-time, page flipping, background animation, panning animation, zoom animation, and sprite operations;

(H) discuss the demands that animation places on a computer system, including; clipping and visibility computations;

(I) research and discuss animation techniques, such as; tweening and morphing, path-of-motion calculation, color-shift animation, background animation, and xor animation; and

(J) collaborate in the creation and presentation of graphics simulations.

(4) Data Analysis (Research and Information Fluency).The student utilizes electronic resources and print media to research, practice and create graphics program projects, with appropriate supervision. The student is expected to:

(A) identify, create, and use available graphics and image file formats;

(B) identify and employ a method to evaluate the design, functionality, and appeal of the graphics program projects;

(C) research and report on the history and evolution of computer graphics, including;CRT technology, PC video technology, Super VGA technology, graphics coprocessors and graphics accelerators.

(D) research and report on graphics applications, such as; Computer Games, Art, Design, Business, Simulations, Virtual reality, Artificial Life, Fractals, Science, Engineering, and Technology.

(E) understand PC graphics hardware and software, including; graphics boards, graphics chips, graphics memory, graphics languages (such as; OpenGL and Directx);

(F) understand the basics of computer graphic systems, including;graphics API, graphics processing, and graphics data pipeline

(G) understand the basics of a computer graphics language, such as; OpenGL, DirectX and Python; and

(H) understand the basics of graphics programming.

(5) Vision (Creativity and Innovation).The student will create new ideas and products and share those ideas and products. The student is expected to:

(A) understand the fundamentals of computer graphics, including; look and feel, graphics coordinate systems, basics of color, device context, and color palettes;

(B) understand graphics image properties, including; bitmap images, file format, lighting, brightness, contrast, pixel color, bit depth, resolution, alpha value, and aspect ratio;

(C) design and create bitmap images;

(D) understand color theory, including; light and the eye, and color models (CIE XYZ, RGB/CMY, HLS);

(E) design and create backgrounds, such as; solid, image, tiled, and rendered;

(F) create graphic designs using image modeling and blending, including; shadows, textures, and wraps;

(G) create graphic designs using image lighting effects, including; illumination, shading, shadows and shaders;

(H) understand graphics artifacts, including; jagged profiles, disappearing detail,improperly rendered detail, disintegrating textures, pixel operation and aliasing;

(I) understand texture mapping rendering attributes, such as; textures, shading, transparency, and dithering;

(J) design rendering plans using visibility, lighting, texturing and shading;

(K) design, create and implement graphics textures, including; seamless, alternating, transition, multi-layer, from an image, and from a surface;

(L) design and create vector graphics, using; vectorized surface, raster operations, blitting, transparency and shadows;

(M) create image special effects using lighting and shading, such as; fog, particle systems, highlights and shadows, retained mode lights, scene lighting, lens effects, local illumination (Phong and Blinn model), Flat shading, Gouraud shading, and Phong shadingl;

(N) create polygonal meshes and faces, using; mesh data structures, vertex arrays and vertex buffer objects;

(O) create optical effects using image plane ray tracing, such as; reflection, refraction, scattering, and chromatic aberration;

(P) create and alter graphics masking, such as; by applying curved models, changing the contrast, spatial frequency modifications, orientation of the texture pattern, and changes in the tessellation of the surface;

(Q) use local area networks and wide area networks to acquire, organize, maintain and evaluate information; and

(R) exert a high level of effort and perseverance towards product development in the creation of computer graphics.

(6) Principles (Technology Operations, Systems, and Concepts). The student creates, delivers and presents computer generated graphics products, with appropriate supervision. The student is expected to:

(A) understand graphics programming fundamentals, including; display modes, setting viewport, cooperative levels, surfaces, palettes, and clipping regions;

(B) understand graphics surfaces, including; surface type, surface operations, surfaces restoration, surface manipulation, transparency and color key;

(C) understand frame buffer operations, such as; double buffering, front buffer, and back buffer;

(D) understand the basic types of graphics data, including; coordinate system, bit mapped, geometrical objects and vector;

(E) understand 2D image rasterization and bitmapped graphics.

(F) create 2D images using drawing primitives for pixels, lines, and curves;

(G) create 2D images using drawing primitives for closed regions, including; circles, rectangles, and polygons;

(H) create 2D images by rendering bitmaps, including; loading the bitmap, displaying the bitmap, and manipulating with sprites;

(I) understand 3D image modeling design, including the coordinate system, rendering pipeline and matrix representation;

(J) understand 3D image display mechanics, including; the rendering engine, mode elements, and file formats;

(K) create images using 3D rendering techniques and rasterization, including; clearing the viewport, rendering the viewport, updating the screen;

(L) create images using 3D projections and perspective, including; lighting, shading, Z-buffering, and textures;

(M) understand the image visible surface determination process, including; z-order, depth buffer, depth coordinates, visible surface algorithm, and compression;

(N) create images using 3D perspective field of view, including; mesh frame, camera position, lighting frame, camera lens, surface material, and mesh color;

(O) understand viewing transformations, including; viewport size, world-to-screen transformation, screen-to-world transformation, and vector array transformation; and

(P) understand projection transformations, including; orthographic, perspective, and clipping.

Technology Applications, Mobile Application Development (½ to 2 Credits)

(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) Students enrolled in this course will learn how to create, design, implement and deliver software for mobile platforms and devices. Students will learn how to use mobile platform development environments and the appropriate programming languages to code solutions to the problems presented throughout the course. Finally, students will communicate their solutions to their peers and the greater online community.

(2) The technology applications curriculum has six strands: Vision (Creativity and Innovation), Collaboration (Teamwork and Communication), Data Analysis (Research and Information Fluency), Reasoning (Critical Thinking, Problem Solving, and Decision Making), Integrity (Digital Citizenship), and Principles (Technology Operations, Systems, and Concepts).

(3) Mobile Application Development will foster students' creativity and innovation by presenting opportunities to design, implement and present meaningful projects mobile computing devices. Students will collaborate with one another, their instructor and with 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 software development concepts to access, analyze, and evaluate information needed to solve problems. By using software design knowledge and skills 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Mobile Application Development course. Students will gain and understanding of the principles of mobile application development through the study of development platforms, programming languages, and software design standards.

(C) Knowledge and skills.

(1) Vision (Creativity and Innovation). The student will understand the advantages and disadvantages of mobile computing. The student is expected to:

(A) understand the difference between desktop and mobile applications; and

(B) identify applications that are best suited for mobile phones.

(2) Vision (Creativity and Innovation). The student will identify and compare existing technologies enabling the development of mobile applications. The student is expected to:

(A) recognize and understand various development environments;

(B) understand the underlying structures (software and hardware) for the design of mobile applications;

(C) understand and use various programming languages to create mobile applications as appropriate;

(D) be familiar with the development workflow of mobile applications; and

(E) recognize and choose between various options for developing mobile applications.

(3) Collaboration (Communication and Collaboration). The student will collaborate with their peers to create, test and implement mobile applications. The student is expected to:

(A) understand and discuss how teams function;

(B) use teamwork to solve problems;

(C) serve as a team leader and a team member and demonstrate appropriate attitudes while serving in those roles;

(D) describe a problem and share it through various forms of media; and

(E) describe a solution to a problem and share it through various forms of media.

(4) Collaboration (Communication and Collaboration). The student will develop skills for managing a project. The student is expected to:

(A) use time-management techniques to develop and maintain work schedules and meet deadlines; and

(B) complete work according to established criteria.

(5) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student will design and create effective user interfaces for various mobile applications. The student is expected to:

(A) understand and create interface frameworks;

(B) understand and create gesture-based interfaces; and

(C) create effective user interfaces for mobile phones.

(6) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student will design, implement and deliver various mobile applications. The student is expected to:

(A) compare and contrast design decisions base on mobile technology, platform, and application purpose;

(B) use object-oriented languages to create various mobile applications;

(C) understand event-based programming and appropriate use;

(D) understand layer animation and appropriate use; and

(E) understand libraries for the design of mobile applications and rationale for appropriate use.

(7) Integrity (Digital Citizenship). The student complies with the laws and examines the issues regarding the safe, legal, and responsible use of technology in society. The student is expected to:

(A) Discuss copyright laws/issues and model ethical acquisition and use of digital information, citing sources and using established methods;

(B) Demonstrate proper 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; and

(D) Understand safety risks associated with the use of social networking sites.

(8) Principles (Technology Operations and Concepts). The student will develop the ability to use and maintain technological products, processes, and systems that enable the creation and distribution of mobile applications. The student is expected to:

(A) understand hardware and software requirements;

(B) recognize multiple platforms and understand associated requirements;

(C) compare and contrast available networks and their implications for mobile application development;

(D) compare and contrast available mobile technologies including platforms and their operating systems and relevance to mobile application development;

(E) compare and contrast available development approaches and how they might be applied to specific technologies and platforms

(F) recognize various program development environments;

(G) understand and use an appropriate program development environment for the various mobile application environments;

(H) understand and use a simulation tool to emulate a mobile device's functionality; and

(I) use actual mobile devices to test mobile applications.

Technology Applications, Programming for Social Networks (½ to 1 Credits)

(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) Students enrolled in this course will learn how to create, design, implement and deliver programming projects designed for social networks. Students will learn how to use appropriate development environments and the appropriate programming languages to code solutions to the problems presented throughout the course. Finally, students will communicate their solutions to their peers and the greater online community.

(2) The technology applications curriculum has six strands: Vision (Creativity and Innovation), Collaboration (Teamwork and Communication), Data Analysis (Research and Information Fluency), Reasoning (Critical Thinking, Problem Solving, and Decision Making), Integrity (Digital Citizenship), and Principles (Technology Operations, Systems, and Concepts).

(3) Programming for Social Networking will foster students' creativity and innovation by presenting opportunities to design, implement and present meaningful programming projects designed for social networks. Students will collaborate with one another, their instructor and with 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 software development concepts to access, analyze, and evaluate information needed to solve problems. By using software design knowledge and skills 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the course. Students will gain and understanding of the principles of programming project design through the study of development platforms, programming languages, interface design, and requirements analysis.

(c) Knowledge and skills.

(1) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student will understand the advantages and disadvantages of using particular models of programming project design for social networks. The student is expected to:

(A) understand the relationship between program design and client / user needs.

(2) Data Analysis (Research and Information Fluency). The student will identify and analyze project needs, requirements, and specifications that must be satisfied to produce the desired result. The student is to:

(A) analyze, identify, and describe social networking project stakeholders (client, user, developer) including their perspectives;

(B) collect and analyze available data to identify program requirements;

(C) analyze, identify, and describe input, output, and processing requirements; and

(D) analyze, identify, and define develop hardware and software specifications.

(3) Vision (Creativity and Innovation). The student will design and create components including interfaces required for successful project completion. The student is expected to:

(A) create appropriate interfaces best suited for identified program purpose; and

(B) create program components appropriate for project needs.

(4) Reasoning (Critical Thinking, Problem Solving, and Decision Making. The student will identify, contrast, and compare existing technologies, including network, that may enable the development of program components. The student is expected to:

(A) recognize and understand various development environments;

(B) identify and evaluate emerging and extant technologies and assess their applicability to current social networking project;

(C) analyze, identify, and define the underlying structures (software and hardware) that will be required by program components and how they will affect design decisions; and

(D) compare and contrast design decisions based on available technology, platform, and application purpose.

(5) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student will design, implement, and deliver required program components. The student is expected to:

(A) develop and evaluate effectiveness of a technology-based project plan to solve a problem using an appropriate software life cycle approach;

(B) analyze, identify, and justify selection of most appropriate programming language(s), including available resources and / or required interfaces;

(C) analyze, identify, and justify available libraries to aide in development of project programs; and

(D) evaluate and justify selection of appropriate project option(s) and program components necessary for successful social netwoking project completion.

(6) Collaboration (Communication and Collaboration). The student will collaborate with their peers to create, test and implement social networking programs. The student is expected to:

(A) understand and discuss how teams function;

(B) use teamwork to solve problems;

(C) serve as a team leader and a team member and demonstrate appropriate attitudes while serving in those roles;

(D) describe a problem and share it through various forms of media; and

(E) describe a solution to a problem and share it through various forms of media.

(7) Collaboration (Communication and Collaboration). The student will develop skills for managing a project. The student is expected to:

(A) use time-management techniques to develop and maintain work schedules and meet deadlines; and

(B) complete work according to established criteria.

(8) Integrity (Digital Citizenship). The student complies with the laws and examines the issues regarding the safe, legal, and responsible use of technology in society. The student is expected to:

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

(B) demonstrate proper 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; and

(D) understand potential risks and benefits associated with the project development and program use.

(9) Principles (Technology Operations and Concepts). The student will develop the ability to use and maintain technological products, processes, and systems that enable the creation and distribution of program(s). The student is expected to:

(A) understand hardware and software requirements;

(B) recognize multiple platforms and understand associated requirements;

(C) compare and contrast available networks and their implications for social network project development;

(D) compare and contrast available technologies including platforms and their operating systems and relevance to social network project development;

(E) compare and contrast available development approaches and how they might be applied to specific technologies and platforms;

(F) recognize various program development environments;

(G) use appropriate program development environment for developing social network project components;

(H) understand appropriate use of simulation tools to evaluate functionality and efficacy of program and its components; and

(I) test program and program components in appropriate environment.

Technology Applications, Project Design (½ to 2 Credits)

(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) Students enrolled in this course will learn how to create, design, implement and deliver technological projects. Students will learn how to use appropriate development environments and the appropriate programming languages to code solutions to the problems presented throughout the course. Finally, students will communicate their solutions to their peers and the greater online community.

(2) The technology applications curriculum has six strands: Vision (Creativity and Innovation), Collaboration (Teamwork and Communication), Data Analysis (Research and Information Fluency), Reasoning (Critical Thinking, Problem Solving, and Decision Making), Integrity (Digital Citizenship), and Principles (Technology Operations, Systems, and Concepts).

(3) Project Design will foster students' creativity and innovation by presenting opportunities to design, implement and present meaningful projects. Students will collaborate with one another, their instructor and with 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 software development concepts to access, analyze, and evaluate information needed to solve problems. By using software design knowledge and skills 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Project Design course. Students will gain and understanding of the principles of project design through the study of development platforms, programming languages, interface design, and requirements analysis.

(c) Knowledge and skills.

(1) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student will understand the advantages and disadvantages of using particular models of project design. The student is expected to:

(A) understand the relationship between the project design and client / user needs.

(2) Data Analysis (Research and Information Fluency). The student will identify and analyze project needs, requirements, and specifications that must be satisfied to produce the desired results. The student is to:

(A) analyze, identify, and describe project stakeholders (client, user, developer) including their perspectives;

(B) collect and analyze available data to identify project requirements;

(C) analyze, identify, and describe input, output, and processing requirements; and

(D) analyze, identify, and define develop hardware and software specifications.

(3) Vision (Creativity and Innovation). The student will design and create components including interfaces required for successful project completion. The student is expected to:

(A) create appropriate interfaces best suited for identified project purpose; and

(B) create project components appropriate for project needs.

(4) Reasoning (Critical Thinking, Problem Solving, and Decision Making. The student will identify, contrast, and compare existing technologies that may enable the development of project components. The student is expected to:

(A) recognize and understand various development environments;

(B) identify and evaluate emerging and extant technologies and assess their applicability to current project;

(C) analyze, identify, and define the underlying structures (software and hardware) that will be required by project applications and how they will affect design decisions; and

(D) compare and contrast design decisions based on available technology, platform, and application purpose.

(5) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student will design, implement and deliver required project components. The student is expected to:

(A) develop and evaluate effectiveness of a technology-based project plan to solve a problem using an appropriate software/system life cycle approach;

(B) analyze, identify, and justify selection of most appropriate programming language(s), including available resources and / or required interfaces;

(C) analyze, identify, and justify available libraries to aide in development of project; and

(D) evaluate and justify selection of appropriate project option(s) and components necessary for successful project completion.

(6) Collaboration (Communication and Collaboration). The student will collaborate with their peers to create, test and implement project and its components. The student is expected to:

(A) understand and discuss how teams function;

(B) use teamwork to solve problems;

(C) serve as a team leader and a team member and demonstrate appropriate attitudes while serving in those roles;

(D) describe a problem and share it through various forms of media; and

(E) describe a solution to a problem and share it through various forms of media.

(7) Collaboration (Communication and Collaboration). The student will develop skills for managing a project. The student is expected to:

(A) use time-management techniques to develop and maintain work schedules and meet deadlines; and

(B) complete work according to established criteria.

(8) Integrity (Digital Citizenship). The student complies with the laws and examines the issues regarding the safe, legal, and responsible use of technology in society. The student is expected to:

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

(B) demonstrate proper 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; and

(D) understand potential risks and benefits associated with the project development and use.

(9) Principles (Technology Operations and Concepts). The student will develop the ability to use and maintain technological products, processes, and systems that enable the creation and distribution of project. The student is expected to:

(A) understand hardware and software requirements;

(B) recognize multiple platforms and understand associated requirements;

(C) compare and contrast available networks and their implications for project development;

(D) compare and contrast available technologies including platforms and their operating systems and relevance to project development;

(E) compare and contrast available development approaches and how they might be applied to specific technologies and platforms;

(F) recognize various program development environments;

(G) use appropriate program development environment for developing project components;

(H) understand appropriate use of simulation tools to evaluate functionality and efficacy of project; and

(I) test project and project components in appropriate environment.

Technology Applications, Robotics and Automation (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) Students enrolled in this course will learn how to identify and describe problems. Students will learn both engineering and programming skills to enable them to brainstorm solutions to the problems they have identified. The students will design and build prototype robots that will solve the problems that have been identified. They will program their robot and test it and repeat the procedure until the problem has been solved. Finally, students will communicate their solutions to their peers and the greater online community.

(2) The technology applications curriculum has six strands: Vision (Creativity and Innovation), Collaboration (Teamwork and Communication), Data Analysis (Research and Information Fluency), Reasoning (Critical Thinking, Problem Solving, and Decision Making), Integrity (Digital Citizenship), and Principles (Technology Operations, Systems, and Concepts).

(3) Robotics and Automation will foster students creativity and innovation by presenting opportunities to design, implement and present meaningful projects through a variety of media. Students will collaborate with one another, their instructor and with 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 engineering concepts to access, analyze, and evaluate information needed to solve problems. By using engineering knowledge and skills 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Robotics and Automation course. Students will gain and understanding of the principles of robotics and automation through the study of physics, robotics, automation and engineering design concepts.

(c) Knowledge and skills.

(1) Vision (Creativity and Innovation). The student designs products using appropriate design processes and techniques. The student is expected to:

(A) identify areas where quality, reliability, and safety can be designed into a product;

(B) improve a product design to meet a specified need;

(C) understand use of sensors in a robotic or automated system;

(D) evaluate design solutions using conceptual, physical, and mathematical models at various times during the design process to check for proper functionality and to note areas where improvements are needed; and

(E) implement a system to identify and track all components of the robotic or automated system and all elements involved with the operation, construction, and manipulative functions.

(2) Vision (Creativity and Innovation). The student designs products using appropriate design processes and techniques. The student is expected to:

(A) identify areas where quality, reliability, and safety can be designed into a product;

(B) improve a product design to meet a specified need;

(C) understand use of sensors in a robotic or automated system;

(D) evaluate design solutions using conceptual, physical, and mathematical models at various times during the design process to check for proper functionality and to not areas where improvements are needed; and

(E) implement a system to identify and track all components of the robotic or automated system and all elements involved with the operation, construction, and manipulative functions.

(3) Collaboration (Communication and Collaboration). The student will collaborate with their peers to create, test and implement robotic and automation projects. The student is expected to:

(A) understand and discuss how teams function;

(B) use teamwork to solve problems;

(C) serve as a team leader and a team member and demonstrate appropriate attitudes while serving in those roles;

(D) describe a problem and share it through various forms of media; and

(E) describe a solution to a problem and share it through various forms of media.

(F) describe perceptions of the quality of products and how they affect engineering decisions;

(G) document prototypes, corrections, and or mistakes in the design process; and

(H) document the final design, construction, and manipulation of finished projects.

(4) Collaboration (Communication and Collaboration). The student will develop skills for managing a project. The student is expected to:

(A) use time-management techniques to develop and maintain work schedules and meet deadlines;

(B) complete work according to established criteria;

(C) participate in the organization and operation of a real or simulated engineering project; and

(D) develop a plan for production of an individual product.

(5) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student uses engineering design methodologies. The student is expected to:

(A) think critically, identify the system constraints, and make fact-based decisions;

(B) use rational thinking to develop or improve a product;

(C) apply decision-making strategies when developing solutions; and

(D) identify quality-control issues in engineering design and production.

(6) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student uses software design strategies to develop robot control programs. The student is expected to:

(A) demonstrate an understanding of and use the sequence programming structure;

(B) demonstrate an understanding of and use the jumps, loops and selection programming structures;

(C) demonstrate an understanding of and use variables;

(D) demonstrate an understanding of and use functions;

(E) demonstrate an understanding of and use output commands;

(F) demonstrate an understanding of and use a graphical programming environment;

(G) demonstrate an understanding of and use constants;

(H) demonstrate an understanding of and use modifiers;

(7) Integrity (Digital Citizenship) The student complies with the laws and examines the issues regarding the safe, legal, and responsible use of technology in society.

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

(B) demonstrate proper 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; and

(D) understand safety risks associated with the use of social networking sites.

(8) Principles (Technology Operations and Concepts) The student will develop the ability to use and maintain technological products, processes, and systems. The student is expected to:

(A) demonstrate the use of computers to manipulate a robotic or automated system and associated subsystems;

(B) troubleshoot and maintain systems and subsystems to ensure safe and proper function and precision operation;

(C) demonstrate knowledge of process control factors; and

(D) demonstrate knowledge of motors, gears, and gear trains used in the robotic or automated systems.

(9) Principles (Technology Operations and Concepts) The student develops an understanding of the advanced concepts of physics, robotics, and automation. The student is expected to:

(A) demonstrate knowledge of rotational dynamics, weight, friction, and traction factors required for the operation of robotic and automated systems;

(B) demonstrate knowledge of torque and power factors used in the operation of robotic systems;

(C) demonstrate knowledge of feedback control loops to provide information; and

(D) demonstrate knowledge of different types of sensors used in robotic or automated systems and their operations.

(10) Principles (Technology Operations and Concepts) The student develops an understanding of the characteristics and scope of manipulators and end effectors required for a robotic or automated system to function. The student is expected to:

(A) demonstrate knowledge of robotic or automated system arm construction;

(B) understand and discuss the relationship of torque, gear ratio, and weight of payload in a robotic or automated system operation; and

(C) demonstrate knowledge of end effectors and their use in linkages and the gearing of a robotic or automated system.

(11) Principles (Technology Operations and Concepts) The student learns the function and application of the tools, equipment, and materials used in robotic and automated systems through specific project-based assessments. The student is expected to:

(A) safely use tools and laboratory equipment to construct and repair systems;

(B) use precision measuring instruments to analyze systems and prototypes; and

(C) use multiple software applications to simulate robot behavior and present concepts.

(12) The student builds a prototype robot using the appropriate tools, materials, and techniques. The student is expected to:

(A) identify and describe the steps needed to produce a prototype;

(B) identify and use appropriate tools, equipment, Machines, and materials to produce the prototype;

(C) implement sensors in a robotic or automated system;

(D) construct a robotic or automated system to perform specified operations using the design process;

(E) test and evaluate the design in relation to pre-established requirements such as criteria and constraints and refine as needed;

(F) refine the design of a robotic or automated system to ensure quality, efficiency, and manufacturability of the final product; and

(G) present the prototype using a variety of media.

Technology Applications, Independent Study in Computer Science—Algorithms (One half or One Credit).

(a) General requirements. The prerequisite for this course is completion of a high school computer science course as identified in this subchapter and permission of the instructor/mentor for Independent Study in Computer Science. This course may be taken at Grades 10-12.

(b) Introduction.

(1) Independent Study in Computer Science - Algorithms is designed for a student to work on computer science algorithms that are not included in other computer science courses and to explore standard algorithms more in depth. The student collaborates with their instructor to identify the advanced algorithms that will be studied over a period of one half year to one year. Upon completion of the course the student should have created a product that is presented to an evaluation panel.

(2) The computer science curriculum has six strands: creativity and innovation, communication and collaboration , research and information fluency, critical thinking, problem solving, and decision making, digital citizenship, technology operations and concepts

(3) Independent Study in Computer Science - Algorithms 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, their instructor and with 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 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Independent Study in Computer Science - Algorithms course. Students will gain and understanding additional formal language(s) not previously learned in their study of computer science.

(c) Knowledge and skills.

(1) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student uses appropriate computer-based productivity tools to create and modify solutions to problems and uses research skills and electronic communication, with appropriate supervision, to create new knowledge. The student is expected to

(A) collaborate with a mentor to determine problem to be solved, hypotheses, and strategies to accomplish task;

(B) identify and solve problems, individually and with input from peers and professionals, utilizing research methods and advanced content development system(s);

(C) select and utilize appropriate productivity tools including, but not limited to, IDE, debugger, network, telecommunication, multimedia tools, and utility programs in the creation of solutions to problems;

(D) using enriched curricular content in the creation of products;

(E) read and use technical documentation;

(F) write simple technical documentation;

(G) develop solutions that meet standards identified by the instuctor; and

(H) produce original work to solve the identified problem and publish the product in electronic media and print;

(2) Integrity (Digital Citizenship). The student complies with the laws and examines the issues regarding the use of technology in society. The student is expected to:

(A) discuss 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, responsible use of software, and knowledge of acceptable use policies when using networks, especially resources on the Internet and intranet;

(D) model respect of intellectual property when manipulating, morphing, or editing graphics, video, text, and sound;

(E) demonstrate best practices in understanding and applying Information Security; and

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

(3) Collaboration (Teamwork and Communication). The student uses technology applications to facilitate evaluation of communication, both process and product. The student is expected to:

(A) design and implement procedures to track trends, set timelines, and review and evaluate the product using technology tools;

(B) determine and employ technology specifications to evaluate projects for design, content delivery, purpose, and audience, demonstrating that process and product can be evaluated using established criteria or rubrics;

(C) seek and respond to input from peers and professionals in evaluating the product;

(D) make necessary revisions and/or proceed to the next stage of study;

(E) develop and maintain a technical documentation library;

(F) demonstrate knowledge and appropriate use of operating systems, software applications, communication and networking components; and

(G) use appropriate technology terminology and syntax in the chosen languages

(4) Data Analysis (Research and Information Fluency).The student utilizes electronic resources and print media to research, practice and test algoritms, with appropriate supervision. The student is expected to:

(A) acquire information using appropriate research strategies and a variety of electronic formats, including interactive components, text, audio, video, and graphics, citing the source; and simulations;

(B) identify, create, and use available file formats including but not limited to text, image, video (analog and digital), and audio files;

(C) identify and employ a method to evaluate the design, functionality, and accuracy of the program solutions; and

(D) analyze information for validity and relevance in the confirmation, testing, and solution of the hypotheses and questions.

(5) Vision (Creativity and Innovation). The student demonstrates an in-depth working understanding of writing program solutions using various advanced algorithms. The student is expected to:

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

(B) apply appropriate search strategies in the acquisition of information from the Internet;

(C) pose hypotheses/questions related to a selected problem;

(D) develop and apply advanced programming skills;

(E) identify and solve problems, individually and with input from peers and professionals, utilizing research methods and advanced programming skills;

(F) use foundation and enrichment curricular content in the creation of program solutions; and

(G) synthesize and generate new information from data gathered from electronic and telecommunications resources.

(6) Principles (Technology Operations, Systems, and Concepts). The student delivers the product electronically in a variety of media, with appropriate supervision. The student is expected to:

(A) collaborate with a mentor to determine a problem to be solved, hypotheses, and strategies to accomplish task;

(B) develop program solutions that meet standards identified by the selected profession or discipline;

(C) format the developed projects according to defined output specifications including target audience and viewing environment;

(D) produce original work to solve the identified problem and publish the product in electronic media and print;

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

(F) present findings to a panel for comment and professional response.

Technology Applications.-Independent Study in Computer Science— Languages (One half or One Credit).

(a) General requirements. The prerequisite for this course is completion of a high school computer science course as identified in this subchapter and permission of the instructor/mentor for Independent Study in Computer Science. This course may be taken at Grades 10-12.

(b) Introduction.

(1) Independent Study in Computer Science - Languages is designed for a student to work on computer science languages that are not included in other computer science courses. The student chooses the programming language or languages that will be studied over a period of one half year to one year. Upon completion of the course the student should have created a product that is presented to an evaluation panel.

(2) The technology applications curriculum has six strands: six strands: creativity and innovation, communication and collaboration , research and information fluency, critical thinking, problem solving, and decision making, digital citizenship, technology operations and concepts

(3) Independent Study in Computer Science - Languages 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, their instructor and with 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 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 will learn to become good digital citizens by researching current laws and regulations and by practicing integrity and respect throughout the Introduction to Computer Science course. Students will gain and understanding additional formal language(s) not previously learned in their study of computer science.

(c) Knowledge and skills.

(1) Reasoning (Critical Thinking, Problem Solving, and Decision Making). The student uses appropriate computer-based productivity tools to create and modify solutions to problems and uses research skills and electronic communication, with appropriate supervision, to create new knowledge. The student is expected to:

(A) colaborate with a mentor to determine problem to be solved, hypotheses, and strategies to accomplish task;

(B) identify and solve problems, individually and with input from peers and professionals, utilizing research methods and advanced content development system(s);

(C) select and utilize appropriate productivity tools including, but not limited to, IDE, debugger, network, telecommunication, multimedia tools, and utility programs in the creation of solutions to problems;

(D) using enriched curricular content in the creation of products;

(E) read and use technical documentation;

(F) write simple technical documentation;

(G) develop products that meet standards identified by the selected profession or discipline; and

(H) produce original work to solve the identified problem and publish the product in electronic media and print.

(2) Integrity (Digital Citizenship). The student complies with the laws and examines the issues regarding the use of technology in society. The student is expected to:

(A) discuss 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, responsible use of software, and knowledge of acceptable use policies when using networks, especially resources on the Internet and intranet;

(D) model respect of intellectual property when manipulating, morphing, or editing graphics, video, text, and sound;

(E) demonstrate best practices in understanding and applying Information Security; and

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

(3) Collaboration (Teamwork and Communication). The student uses technology applications to facilitate evaluation of communication, both process and product. The student is expected to:

(A) design and implement procedures to track trends, set timelines, and review and evaluate the product using technology tools;

(B) determine and employ technology specifications to evaluate projects for design, content delivery, purpose, and audience, demonstrating that process and product can be evaluated using established criteria or rubrics;

(C) seek and respond to input from peers and professionals in evaluating the product;

(D) make necessary revisions and/or proceed to the next stage of study;

(E) develop and maintain a technical documentation library;

(F) demonstrate knowledge and appropriate use of operating systems, software applications, communication and networking components; and

(G) use appropriate technology terminology and syntax in the chosen languages

(4) Data Analysis (Research and Information Fluency).The student utilizes electronic resources and print media to research, practice and test language content, with appropriate supervision. The student is expected to:

(A) acquire information using appropriate research strategies and a variety of electronic formats, including Interactive components, text, audio, video, and graphics, citing the source; and simulations;

(B) identify, create, and use available file formats including but not limited to text, image, video (analog and digital), and audio files;

(C) identify and employ a method to evaluate the design, functionality, and accuracy of the program solutions; and

(D) analyze information for validity and relevance in the confirmation, testing, and solution of the hypotheses and questions.

(5) Vision (Creativity and Innovation). The student demonstrates an in-depth working understanding of a chosen programming language. The student is expected to:

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

(B) apply appropriate search strategies in the acquisition of information from the Internet including keyword and Boolean search strategies;

(C) pose hypotheses/questions related to a selected problem;

(D) develop and apply advanced programming skills;

(E) identify and solve problems, individually and with input from peers and professionals, utilizing research methods and advanced programming skills;

(F) use foundation and enrichment curricular content in the creation of products; and

(G) synthesize and generate new information from data gathered from electronic and telecommunications resources.

(6) Principles (Technology Operations, Systems, and Concepts). The student delivers the product electronically in a variety of media, with appropriate supervision. The student is expected to:

(A) collaborate with a mentor to determine a problem to be solved, hypotheses, and strategies to accomplish task;

(B) develop products that meet standards identified by the selected profession or discipline;

(C) format the developed projects according to defined output specifications including target audience and viewing environment;

(D) produce original work to solve the identified problem and publish the product in electronic media and print;

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

(F) present findings to a panel for comment and professional response.