Computer Science - AP

HIGH WAGE
DC CAREER AND TECHNICAL EDUCATION

HIGH SKILL
DC CAREER AND TECHNICAL EDUCATION

IN-DEMAND
DC CAREER AND TECHNICAL EDUCATION

Updated August 31, 2022

DC·CTE
CAREER & TECHNICAL EDUCATION

OSSE

Office of the State Superintendent of Education
Postsecondary and Career Education Division
Career and Technical Education Department

Christina Grant, Ed.D.
State Superintendent of Education

Antoinette Mitchell, Ph.D.
Assistant Superintendent, Postsecondary and Career Education

Kilin Boardman-Schroyer
Deputy Assistant Superintendent, Postsecondary and Career Education

Richard W. Kincaid
State Director, Career and Technical Education

The purpose of this document is to communicate the required Career and Technical Education (CTE) academic standards for the Computer Science - AP Program of Study. The academic standards in this document are theoretical and performance-based. The standards contain content from Colorado, Maryland, Tennessee, and Texas and were validated by D.C. business and industry partners. All content is used with permission.

In addition to academic standards, OSSE has incorporated into this document Labor Market Information (LMI) definitions and explanations for the Program of Study; program aligned Industry Recognized Credentials; and Work-Based Learning resources and requirements by course level.

This document is intended for use by educational administrators and practitioners. A similar document is available for each state-approved CTE Program of Study.

Updated August 31, 2022

Table of Contents

Course Descriptions: Computer Science - AP		
Course Level	**Course Information**	**Description**
Level I	**Fundamentals of Computer Science** **OSSEID:** 10019G0.5015 **Grades:** 9-12 **Prerequisite:** None **Credit:** .5	Fundamentals of Computer Science is intended as a first course for those students just beginning the study of computer science. Students will learn about the computing tools that are used every day. Students will foster their creativity and innovation through opportunities to design, implement, and present solutions to real-world problems. Students will collaborate and use computer science concepts to access, analyze, and evaluate information needed to solve problems. Students will learn the problem-solving and reasoning skills that are the foundation of computer science. By using computer science knowledge and skills that support the work of individuals and groups in solving problems, students will select the technology appropriate for the task, synthesize knowledge, create solutions, and evaluate the results. Students will learn digital citizenship by researching current laws and regulations and by practicing integrity and respect. Students will gain an understanding of the principles of computer science through the study of technology operations and concepts.
Level I	**Computer Science I** **OSSEID:** 10901G0.5025 **Grades:** 9-12 **Prerequisite:** None **Credit:** .5	Computer Science I will foster students' creativity and innovation by presenting opportunities to design, implement, and present meaningful programs through a variety of media. Students will collaborate with one another, their instructor, and various electronic communities to solve the problems presented throughout the course. Through data analysis, students will identify task requirements, plan search strategies, and use computer science concepts to access, analyze, and evaluate information needed to solve problems. By using computer science knowledge and skills that support the work of individuals and groups in solving problems, students will select the technology appropriate for the task, synthesize knowledge, create solutions, and evaluate the results. Students will learn digital citizenship by researching current laws and regulations and by practicing integrity and respect. Students will gain an understanding of the principles of computer science through the study of technology operations, systems, and concepts.
Level II	**Computer Science II** **OSSEID:** 10901G1.0035 **Grades:** 10-12	Computer Science II will foster students' creativity and innovation by presenting opportunities to design, implement, and present meaningful programs through a variety of media.

	Prerequisite: Fundamentals of Computer Science **Credit:** 1	Students will collaborate with one another, their instructor, and various electronic communities to solve the problems presented throughout the course. Through data analysis, students will identify task requirements, plan search strategies, and use computer science concepts to access, analyze, and evaluate information needed to solve problems. By using computer science knowledge and skills that support the work of individuals and groups in solving problems, students will select the technology appropriate for the task, synthesize knowledge, create solutions, and evaluate the results. Students will learn digital citizenship by researching current laws and regulations and by practicing integrity and respect. Students will gain an understanding of computer science through the study of technology operations, systems, and concepts.
Level III	**Computer Science A** **OSSEID:** 10014G1.0045 **Grades:** 11-12 **Prerequisite: C**omputer Science II **Credit:** 1	Computer Science A – AP focuses on further developing computational-thinking skills through the medium of Android™ App development for mobile platforms. The course utilizes industry-standard tools such as Android Studio, Java™ programming language, XML, and device emulators. Students collaborate to create original solutions to problems of their own choosing by designing and implementing user interfaces and Web-based databases. PLTW is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CS A). This endorsement affirms that all components of PLTW CS A's offerings are aligned to the AP Curriculum Framework standards and the AP CS A assessment.
Level IV	**Computer Science III** **OSSEID:** 10901G1.0055 **Grades:** 12 **Prerequisite:** Computer Science A **Credit:** 1	Computer Science III will foster students' creativity and innovation by presenting opportunities to design, implement, and present meaningful programs through a variety of media. Students will collaborate with one another, their instructor, and various electronic communities to solve the problems presented throughout the course. Through data analysis, students will identify task requirements, plan search strategies, and use computer science concepts to access, analyze, and evaluate information needed to solve problems. By using computer science knowledge and skills that support the work of individuals and groups in solving problems, students will select the technology appropriate for the task, synthesize knowledge, create solutions, and evaluate the results. Students will learn digital citizenship by researching current laws and regulations and by practicing integrity and

		respect. Students will gain an understanding of advanced computer science data structures through the study of technology operations, systems, and concepts.

Industry Certifications

Work-Based Learning Examples and Resources

Level I Course	Level II Course	Level III Course	Level IV Course
Career Exploration Industry Visits Guest Speakers Participate in a CTSO	**Career Awareness** *All of Level I, plus:* Postsecondary Visits Program-Specific Site Tours Mock Interviews	**Career Preparation** *All of Level I and II, plus:* Job Shadow Paid/Unpaid Internships	**Career Preparation** Paid/Unpaid Internships Apprenticeships

Several resources are available to help instructors meet the Level I and Level II WBL requirements, including:

Career Coach DC (http://careercoachdc.emsicc.com). Online site designed to help students find and connect to a career pathway by providing the most current local data on wages, employment, job postings, and associated education and training. The resource includes a Career Assessment for students.

Nepris (https://dc.nepris.com/). Connects educators and learners with a network of industry professionals virtually, bringing real-world relevance and career exposure to all students. Nepris also provides a skills-based volunteering platform for business and industry professionals to extend their educational outreach.

Virtual Job Shadow (https://virtualjobshadow.com). Provides interactive tools which empower students to discover, plan, and pursue their dreams. Rich video library presents a "day in the life of" view for thousands of occupations.

Labor Market Information Definitions and Data

Career and Technical Education programs of study in the District of Columbia must meet at least one of the High Wage, High Skill, and In-Demand definitions below to be considered appropriate for our students and the regional labor market. These definitions were created in collaboration with Career and Technical Education leaders from District of Columbia LEAs, the University of the District of Columbia Community College, and national guidance from Research Triangle International (RTI) and Education Northwest. Additionally, previous work was consulted by researchers at MIT's Labor Wage Index Project and the DC CTE Task Force's 2012 Strategic Plan for the District of Columbia.

Indicator	Definition	Data for the Computer Science - AP Program of Study (source: EMSI, August 2022)
High Wage	Those occupations that have a 25th percentile wage equal to or greater than the most recent MIT Living Wage Index for one adult in the District of Columbia, and/or leads to a position that pays at least the median hourly or annual wage for the Washington, DC, metropolitan statistical area. *Note: A 25th percentile hourly wage of $23.13 or greater is required to meet this definition.*	**Standard Occupational Code (SOC):** 15-1251.00 Computer Programmers **Hourly Wages** **25th Percentile:** $38.84 **50th Percentile:** $49.50 **75th Percentile:** $63.48
High Skill	Those occupations located within the Washington, DC, metropolitan statistical area with the following education or training requirements: completion of an apprenticeship program; completion of an industry-recognized certification or credential; associate’s degree or higher.	**Typical Entry-Level Education:** Bachelor’s Degree
In-Demand	Those occupations in the Washington, DC, metropolitan statistical area having more than the median number of total **(growth plus replacement)** annual openings over a five-year period. *Note: An occupation is required to have an annual growth plus replacement rate of 105 openings, or greater, between 2021-2026 to meet this definition.*	**Annual Openings:** 688

DC★CTE
CAREER & TECHNICAL EDUCATION

Model Six-Year Plan: Computer Science - AP

College: University of the District of Columbia Community College
Program/CIP:
Plan:

Entity: Office of the State Superintendent of Education
Career Cluster: Information Technology
Program of Study: Computer Science - AP

High School					College			
Subject	**9th Grade**	**10th Grade**	**11th Grade**	**12th Grade**	**Semester I**	**Semester II**	**Semester III**	**Semester IV**
English (4)	English I	English II	English III	English IV				
Math (4)	Algebra I	Geometry	Algebra II	Math				
Science (4)	Biology	Lab Science	Anatomy and Physiology	Science				
Social Studies (4)	World History and Geography I: Middle Ages	World History and Geography II: Modern World	U.S. History	U.S. Government (.5) and D.C. History (.5)				
Health (.5) and Physical Ed (1)	Health (.5) Physical Ed (.5)	Physical Ed (.5)						
World Languages (2)			World Language I	World Language II				
Art (.5)		Art (.5)						
Music (.5)		Music (.5)						
Elective / Major Courses	Fundamentals of Computer Science (.5) and Computer Science I (.5)	Computer Science II	Computer Science A	Computer Science III				
Total possible college credits completed in high school: XX					***Credit hours required to complete the AAS program: XX***			

Course Standards

Fundamentals of Computer Science

1. **General requirements.** This course is recommended for students in Grades 9-12. Students shall be awarded one-half credit for successful completion of this course.

2. **Introduction.**
   - A. Career and technical education instruction provides content aligned with challenging academic standards and relevant technical knowledge and skills for students to further their education and succeed in current or emerging professions.

   - B. The Information Technology (I.T.) Career Cluster focuses on building linkages in I.T. occupations for entry-level, technical, and professional careers related to the design, development, support, and management of hardware, software, multimedia, and systems integration services.

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

   - D. Students will participate in at least two Career Exploration Work-Based Learning experiences in this course, which might include guest speakers and workplace tours relevant to the program of study.

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

3. **Knowledge and skills.**
   - A. **Creativity and innovation. The student develops products and generates new understanding by extending existing knowledge. The student is expected to:**
     1. investigate and explore various career opportunities within the computer science field and report findings through various media;
     2. create and publish interactive stories, games, and animations;
     3. create and publish interactive animations;
     4. create algorithms for the solution of various problems;
     5. create web pages using a mark-up language;
     6. use the Internet to create and publish solutions; and

7. design creative and effective user interfaces.

B. **Communication and collaboration. The student communicates and collaborates with peers to contribute to his or her own learning and the learning of others. The student is expected to:**

1. seek and respond to advice from peers and professionals in evaluating problem solutions;
2. debug and solve problems using reference materials and effective strategies; and
3. publish information in a variety of ways such as print, monitor display, web pages, and video.

C. **The student develops technology skills. The student is expected to:**

1. Use technology as a tool to research, organize, evaluate, and communicate information.
2. Use digital technologies (computers, PDAs, media players, GPSs, etc.), communication/networking tools, and social networks appropriately to access, manage, integrate, evaluate, and create information to successfully function in a knowledge economy.
3. Demonstrate utilizing current and new technologies specific to the program of study, course, and/or industry.
4. Apply a fundamental understanding of the ethical/legal issues surrounding the access and use of information technologies.

D. **Research and information fluency. The student locates, analyzes, processes, and organizes data. The student is expected to:**

1. construct appropriate electronic search strategies; and
2. use a variety of resources, including other subject areas, together with various productivity tools to gather authentic data as a basis for individual and group programming projects.

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

1. demonstrate the ability to insert applets into web pages;
2. find, download, and insert scripting code into web pages to enhance interactivity;
3. understand binary representation of data in computer systems, perform conversions between decimal and binary number systems, and count in binary number systems;
4. read and define a problem's description, purpose, and goals;
5. demonstrate coding proficiency in a contemporary programming language by developing solutions that create stories, games, and animations;
6. choose, identify, and use the appropriate data type to properly represent data in a problem solution;
7. demonstrate an understanding of and use variables within a programmed story, game, or animation;
8. demonstrate proficiency in the use of arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division;
9. demonstrate an understanding of and use sequence within a programmed story, game, or animation;
10. demonstrate an understanding of and use conditional statements within a programmed story, game, or animation;
11. demonstrate an understanding of and use iteration within a programmed story, game, or animation;
12. create an interactive story, game, or animation;
13. use random numbers within a programmed story, game, or animation; and
14. test program solutions by investigating valid and invalid data.

F. **Digital citizenship. The student explores and understands safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to:**

1. discuss copyright laws/issues and model ethical acquisition of digital information by citing sources using established methods;
2. demonstrate proper digital etiquette and knowledge of acceptable use policies when using networks, especially resources on the Internet and on intranets;
3. investigate measures such as passwords or virus detection/prevention to protect computer systems and databases from unauthorized use and tampering;
4. understand the safety risks associated with the use of social networking sites;
5. discuss the impact of computing and computing-related advancements on society; and
6. determine the reliability of information available through electronic media.

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

1. demonstrate knowledge of the basic computer components, including a central processing unit (CPU), storage, and input/output devices;
2. use operating system tools, including appropriate file management;
3. demonstrate knowledge and appropriate use of different operating systems;
4. demonstrate knowledge and understanding of basic network connectivity;
5. describe, compare, and contrast the differences between an application and an operating system; and
6. compare, contrast, and appropriately use various input, processing, output, and primary/secondary storage devices.

H. **The student develops technology skills. The student is expected to:**

1. use technology as a tool to research, organize, evaluate, and communicate information;
2. use digital technologies (computers, PDAs, media players, GPSs, etc.); communication/networking tools, and social networks appropriately to access, manage, integrate, evaluate, and create information to function in a knowledge economy successfully;
3. demonstrate using current and new technologies specific to the program of study, course, and/or industry; and
4. apply a fundamental understanding of the ethical/legal issues surrounding the access and use of information technologies.

Computer Science I

1. **General requirements.** This course is recommended for students in Grades 9-12. Prerequisite: Fundamentals of Computer Science. Students shall be awarded one-half credit for successful completion of this course.

2. **Introduction.**
   A. Career and technical education instruction provides content aligned with challenging academic standards and relevant technical knowledge and skills for students to further their education and succeed in current or emerging professions.

   B. The Information Technology (IT) Career Cluster focuses on building linkages in IT occupations for entry level, technical, and professional careers related to the design, development, support, and management of hardware, software, multimedia, and systems integration services.

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

   D. Students will participate in at least two Career Exploration Work-Based Learning experiences in this course, which might include guest speakers and work-place tours relevant to the program of study.

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

3. **Knowledge and skills.**
   A. **Creativity and innovation. The student develops products and generates new understandings by extending existing knowledge. The student is expected to:**
      1. participate with electronic communities as a learner, initiator, contributor, and teacher/mentor;
      2. extend the learning environment beyond the school walls with digital products created to increase teaching and learning in the other subject areas; and
      3. participate in relevant, meaningful activities in the larger community and society to create electronic projects.

   B. **Communication and collaboration. The student communicates and collaborates with peers to contribute to his or her own learning and the learning of others. The student is expected to:**
      1. create and properly display meaningful output;
      2. create interactive console display interfaces, with appropriate user prompts, to acquire data from a user;

3. use Graphical User Interfaces (GUIs) to create interactive interfaces to acquire data from a user and display program results;
4. write programs with proper 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;
5. improve numeric display by optimizing data visualization;
6. display simple vector graphics using lines, circles, and rectangles;
7. display simple bitmap images; and
8. seek and respond to advice from peers and professionals in evaluating quality and accuracy.

C. **Research and information fluency. The student locates, analyzes, processes, and organizes data. The student is expected to:**

1. use a variety of resources, including foundation and enrichment curricula, to gather authentic data as a basis for individual and group programming projects; and
2. use various productivity tools to gather authentic data as a basis for individual and group programming projects.

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

1. use program design problem-solving strategies to create program solutions;
2. define and specify the purpose and goals of solving a problem;
3. identify the subtasks needed to solve a problem;
4. identify the data types and objects needed to solve a problem;
5. identify reusable components from existing code;
6. design a solution to a problem;
7. code a solution from a program design;
8. identify and debug errors;
9. test program solutions with appropriate valid and invalid test data for correctness;
10. debug and solve problems using error messages, reference materials, language documentation, and effective strategies;
11. explore common algorithms, including finding greatest common divisor, finding the biggest number out of three, finding primes, making change, and finding the average;
12. analyze and modify existing code to improve the underlying algorithm;
13. create program solutions that exhibit robust behavior by understanding, avoiding, and preventing runtime errors, including division by zero and type mismatch;
14. select the most appropriate algorithm for a defined problem;
15. demonstrate proficiency in the use of the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division;
16. create program solutions to problems using available mathematics libraries, including absolute value, round, power, square, and square root;
17. develop program solutions that use assignment;
18. develop sequential algorithms to solve non-branching and non-iterative problems;
19. develop algorithms to decision-making problems using branching control statements;
20. develop iterative algorithms and code programs to solve practical problems

21. demonstrate proficiency in the use of the relational operators;
22. demonstrate proficiency in the use of the logical operators; and
23. generate and use random numbers.

E. **Digital citizenship. The student explores and understands safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to:**

1. discuss intellectual property, privacy, sharing of information, copyright laws, and software licensing agreements;
2. model ethical acquisition and use of digital information;
3. demonstrate proper digital etiquette, responsible use of software, and knowledge of acceptable use policies;
4. investigate measures, including passwords and virus detection/prevention, to protect computer systems and databases from unauthorized use and tampering; and
5. investigate how technology has changed and the social and ethical ramifications of computer usage.

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

1. compare and contrast types of operating systems, software applications, and programming languages;
2. demonstrate knowledge of major hardware components, including primary and secondary memory, a central processing unit (CPU), and peripherals;
3. differentiate among current programming languages, discuss the use of those languages in other fields of study, and demonstrate knowledge of specific programming terminology and concepts;
4. differentiate between a high-level compiled language and an interpreted language;
5. understand concepts of object-oriented design;
6. use local and global scope access variable declarations;
7. encapsulate data and associated subroutines into an abstract data type;
8. create subroutines that do not return values with and without the use of arguments and parameters;
9. create subroutines that return typed values with and without the use of arguments and parameters;
10. understand and identify the data-binding process between arguments and parameters;
11. compare objects using reference values and a comparison routine;
12. understand the binary representation of numeric and nonnumeric data in computer systems;
13. understand the finite limits of numeric data;
14. perform numerical conversions between the decimal and binary number systems and count in the binary number system;
15. choose, identify, and use the appropriate data types for integer, real, and Boolean data when writing program solutions;
16. demonstrate an understanding of the concept of a variable;
17. demonstrate an understanding of and use reference variables for objects;
18. demonstrate an understanding of how to represent and manipulate text data, including concatenation and other string functions;
19. demonstrate an understanding of the concept of scope;
20. identify and use the structured data type of one-dimensional arrays to traverse, search, and modify data;
21. choose, identify, and use the appropriate data type and structure to properly represent the data in a program problem solution; and

22. compare and contrast strongly typed and un-typed programming languages.

G. **The student develops technology skills. The student is expected to:**

1. use technology as a tool to research, organize, evaluate, and communicate information;
2. use digital technologies (computers, PDAs, media players, GPSs, etc.); communication/networking tools, and social networks appropriately to access, manage, integrate, evaluate, and create information to function in a knowledge economy successfully;
3. demonstrate using current and new technologies specific to the program of study, course, and/or industry; and
4. apply a fundamental understanding of the ethical/legal issues surrounding the access and use of information technologies.

Computer Science II

1. **General requirements.** This course is recommended for students in Grades 10-12. Students shall be awarded one credit for successful completion of this course. Prerequisites: Computer Science I and Fundamentals of Computer Science.

2. **Introduction.**

   A. Career and technical education instruction provides content aligned with challenging academic standards and relevant technical knowledge and skills for students to further their education and succeed in current or emerging professions.

   B. The Information Technology (IT) Career Cluster focuses on building linkages in IT occupations for entry level, technical, and professional careers related to the design, development, support, and management of hardware, software, multimedia, and systems integration services.

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

   D. Students will participate in at least two Career Awareness Work-Based Learning experiences in this course, which might include informational interviews or job shadowing relevant to the program of study.

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

3. **Knowledge and skills.**

   A. **Creativity and innovation. The student develops products and generates new understandings by extending existing knowledge. The student is expected to:**

      1. use program design problem-solving strategies to create program solutions;
      2. demonstrate the ability to read and modify large programs, including the design description and process development;
      3. follow the systematic problem-solving process of identifying the specifications of purpose and goals, the data types and objects needed, and the subtasks to be performed;
      4. compare and contrast design methodologies and implementation techniques such as top-down, bottom-up, and black box;
      5. analyze, modify, and evaluate existing code by performing a case study on a large program, including inheritance and black box programming;

6. identify the data types and objects needed to solve a problem;
7. 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;
8. use object-oriented programming development methodology, data abstraction, encapsulation with information hiding, and procedural abstraction in program development and testing; and
9. create, edit, and manipulate bitmap images that are used to enhance user interfaces and program functionality.

B. **Communication and collaboration. The student communicates and collaborates with peers to contribute to his or her own learning and the learning of others. The student is expected to:**

1. use the principles of software engineering to work in software design teams, break a problem statement into specific solution requirements, create a program development plan, code part of a solution from a program development plan while a partner codes the remaining part, team test the solution for correctness, and develop presentations to report the solution findings;
2. create interactive console display interfaces with appropriate user prompts;
3. create interactive human interfaces to acquire data from a user and display program results using an advanced Graphical User Interface (GUI);
4. 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;
5. improve data display by optimizing data visualization;
6. display simple vector graphics to interpret and display program results; an
7. display simple bitmap images.

C. **Research and information fluency. The student locates, analyzes, processes, and organizes data. The student is expected to:**

1. use local area networks (LANs) and wide area networks (WANs), including the Internet and intranets, in research, file management, and collaboration;
2. understand programming file structure and file access for required resources;
3. acquire and process information from text files, including files of known and unknown sizes;
4. manipulate data structures using string processing;
5. manipulate data values by casting between data types;
6. identify and use the structured data type of one-dimensional arrays to traverse, search, modify, insert, and delete data;
7. identify and use the structured data type of two-dimensional arrays to traverse, search, modify, insert, and delete data; and
8. identify and use a list object data structure to traverse, search, insert, and delete data.

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

1. develop sequential algorithms using branching control statements, including nested structures, to create solutions to decision-making problems;
2. develop choice algorithms using selection control statements based on ordinal values;
3. demonstrate proficiency in the use of short-circuit evaluation;

4. demonstrate proficiency in the use of Boolean algebra, including De Morgan's Law;
5. develop iterative algorithms using nested loops;
6. identify, trace, and appropriately use recursion in programming solutions, including algebraic computations;
7. design, construct, evaluate, and compare search algorithms, including linear searching and binary searching;
8. identify, describe, design, create, evaluate, and compare standard sorting algorithms, including selection sort, bubble sort, insertion sort, and merge sort;
9. measure time/space efficiency of various sorting algorithms;
10. compare and contrast search and sort algorithms, including linear, quadratic, and recursive strategies, for time/space efficiency;
11. analyze algorithms using "big-O" notation for best, average, and worst-case data patterns;
12. develop algorithms to solve various problems, including factoring, summing a series, finding the roots of a quadratic equation, and generating Fibonacci numbers;
13. test program solutions by investigating boundary conditions; testing classes, methods, and libraries in isolation; and performing stepwise refinement;
14. identify and debug compile, syntax, runtime, and logic errors;
15. compare and contrast algorithm efficiency by using informal runtime comparisons, exact calculation of statement execution counts, and theoretical efficiency values using "big-O" notation, including worst-case, best-case, and average-case time/space analysis;
16. demonstrate the ability to count, convert, and perform mathematical operations in the binary and hexadecimal number systems;
17. demonstrate knowledge of the maximum integer boundary, minimum integer boundary, imprecision of real number representations, and round-off errors;
18. create program solutions to problems using the mathematics library class;
19. use random algorithms to create simulations that model the real world;
20. identify, understand, and create class specifications and relationships among classes, including composition and inheritance relationships;
21. understand and explain object relationships among defined classes, abstract classes, and interfaces;
22. create object-oriented definitions using class declarations, variable declarations, constant declarations, method declarations, parameter declarations, and interface declarations;
23. create robust classes that encapsulate data and the methods that operate on that data and incorporate overloading to enrich the object's behavior;
24. design and implement a set of interactive classes;
25. design, create, and evaluate multiclass programs that use abstract classes and interfaces;
26. understand and implement a student-created class hierarchy;
27. extend, modify, and improve existing code using inheritance;
28. create adaptive behaviors, including overloading, using polymorphism;
29. understand and use reference variables for object and string data types;
30. understand and implement access scope modifiers;
31. understand and demonstrate how to compare objects;
32. duplicate objects using the appropriate deep and/or shallow copy;
33. define and implement abstract classes and interfaces in program problem solutions;
34. apply functional decomposition to a program solution;
35. create simple and robust objects from class definitions through instantiation;
36. apply class membership of variables, constants, and methods;

37. examine and mutate the properties of an object using assessors and modifiers;
38. understand and implement a composite class; and
39. design and implement an interface.

E. **Digital citizenship. The student explores and understands safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to:**

   1. model ethical acquisition and use of digital information;
   2. demonstrate proper digital etiquette, responsible use of software, and knowledge of acceptable use policies; and
   3. investigate digital rights management.

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

   1. compare and contrast types of operating systems, software applications, hardware platforms, and programming languages;
   2. demonstrate knowledge of major hardware components, including primary and secondary memory, a central processing unit (CPU), and peripherals;
   3. demonstrate knowledge of major networking components, including hosts, servers, switches, and routers;
   4. demonstrate knowledge of computer communication systems, including single-user, peer-to-peer, workgroup, client-server, and networked;
   5. demonstrate knowledge of computer addressing systems, including Internet Protocol (IP) address and Media Access Control (MAC) address; and
   6. differentiate among the categories of programming languages, including machine, assembly, high-level compiled, high-level interpreted, and scripted.

G. **The student develops technology skills. The student is expected to:**

   1. use technology as a tool to research, organize, evaluate, and communicate information;
   2. use digital technologies (computers, PDAs, media players, GPSs, etc.); communication/networking tools, and social networks appropriately to access, manage, integrate, evaluate, and create information to function in a knowledge economy successfully;
   3. demonstrate using current and new technologies specific to the program of study, course, and/or industry; and
   4. apply a fundamental understanding of the ethical/legal issues surrounding the access and use of information technologies.

Computer Science A – AP

1. **General requirements.** This course is recommended for students in Grades 11-12. Prerequisite: Computer Science Principles. Students shall be awarded one credit for successful completion of this course.

2. **Introduction.**
   A. Career and technical education instruction provides content aligned with challenging academic standards and relevant technical knowledge and skills for students to further their education and succeed in current or emerging professions.

   B. The Information Technology (I.T.) Career Cluster focuses on building linkages in I.T. occupations for entry-level, technical, and professional careers related to the design, development, support, and management of hardware, software, multimedia, and systems integration services.

   C. Computer Science A – AP focuses on further developing computational-thinking skills through the medium of Android™ App development for mobile platforms. The course utilizes industry-standard tools such as Android Studio, Java™ programming language, XML, and device emulators. Students collaborate to create original solutions to problems of their own choosing by designing and implementing user interfaces and Web-based databases. PLTW is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CS A). This endorsement affirms that all components of PLTW CS A's offerings are aligned with the AP Curriculum Framework standards and the AP CS A assessment.

   D. Students will participate in a Career Preparation Work-Based Learning experience in this course, which might include paid or unpaid internship experiences relevant to the program of study.

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

1. **Knowledge and Skills.** Content requirements. Content requirements for Advanced Placement (A.P.) Computer Science A is prescribed in the College Board Publication Advanced Placement Course in Computer Science A published by The College Board.

Computer Science III

1. **General requirements.** This course is recommended for students in Grade 12. Students shall be awarded one credit for successful completion of this course. Prerequisite: Computer Science A.

2. **Introduction.**
   A. Career and technical education instruction provides content aligned with challenging academic standards and relevant technical knowledge and skills for students to further their education and succeed in current or emerging professions.

   B. The Information Technology (IT) Career Cluster focuses on building linkages in IT occupations for entry level, technical, and professional careers related to the design, development, support, and management of hardware, software, multimedia, and systems integration services.

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

   D. Students will participate in a Career Preparation Work-Based Learning experience in this course, which includes paid or unpaid internship, pre-apprenticeship, or apprenticeship experiences relevant to the program of study.

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

3. **Knowledge and skills.**
   A. **Creativity and innovation. The student develops products and generates new understandings by extending existing knowledge. The student is expected to:**
      1. apply data abstraction and encapsulation to manage complexity;
      2. implement a student-created class hierarchy;
      3. read and write class specifications using visual organizers, including Unified Modeling Language;
      4. use black box programming methodology;
      5. design, create, and use interfaces to apply protocols;
      6. identify, describe, design, create, evaluate, and compare standard sorting algorithms that perform sorting operations on data structures, including quick sort and heap sort;
      7. select, identify, and use the appropriate abstract data type, advanced data structure, and supporting algorithms to properly represent the data in a program problem solution; and
      8. manage complexity by using a systems approach.

B. **Communication and collaboration. The student communicates and collaborates with peers to contribute to his or her own learning and the learning of others. The student is expected to:**

1. use local area networks (LANs) and wide area networks (WANs), including the Internet and intranets, in research, file management, and collaboration;
2. create interactive human interfaces to acquire data from a user and display program results using an advanced Graphical User Interface (GUI);
3. write programs and communicate with proper programming style to enhance the readability and functionality of the code by using meaningful descriptive identifiers, internal comments, white space, indentation, and a standardized program style; and
4. work in software design teams.

C. **Research and information fluency. The student locates, analyzes, processes, and organizes data. The student is expected to:**

1. identify and use the structured data type of arrays of objects to traverse, search, modify, insert, and delete data;
2. identify and use two-dimensional ragged arrays to traverse, search, modify, insert, and delete data;
3. identify and use a list object data structure, including vector, to traverse, search, insert, and delete object data;
4. understand and trace a linked-list data structure;
5. create program solutions using a linked-list data structure, including unordered single, ordered single, double, and circular linked;
6. understand composite data structures, including a linked list of linked lists;
7. understand and create program solutions using stacks, queues, trees, heaps, priority queues, graph theory, and enumerated data types;
8. understand and create program solutions using sets, including HashSet and TreeSet;
9. understand and create program solutions using maps, including HashMap and TreeMap; and
10. write and modify text file data.

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

1. develop choice algorithms using selection control statements, including break, label, and continue;
2. demonstrate proficiency in the use of the bitwise operators;
3. develop iterative algorithms using do-while loops;
4. demonstrate proficiency in the use of the ternary operator;
5. create program solutions that use iterators;
6. identify, trace, and appropriately use recursion;
7. understand and create program solutions using hashing;
8. perform pattern recognition using regular expressions;
9. explore common algorithms, including matrix addition and multiplication, fractals, Towers of Hanoi, and magic square;
10. create program solutions that exhibit robust behavior by understanding and avoiding runtime errors and handling anticipated errors;
11. understand object-oriented design concepts of inner classes, outer classes, and anonymous classes;

12. use object reference scope identifiers, including null, this, and super;
13. provide object functionality to primitive data types;
14. write program assumptions in the form of assertions;
15. write a Boolean expression to test a program assertion; and
16. construct assertions to make explicit program invariants.

E. **Digital citizenship. The student explores and understands safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to:**
   1. model ethical acquisition and use of digital information; and
   2. demonstrate proper digital etiquette, responsible use of software, and knowledge of acceptable use policies.

F. **Technology operations and concepts. The student understands technology concepts, systems, and operations as they apply to computer science. The student is expected to:**
   1. compare and contrast high-level programming languages;
   2. create a small workgroup network;
   3. create and apply a basic network addressing scheme; and
   4. create discovery programs in a low-level language, high-level language, and scripting language.

G. **The student develops technology skills. The student is expected to:**
   1. use technology as a tool to research, organize, evaluate, and communicate information;
   2. use digital technologies (computers, PDAs, media players, GPSs, etc.); communication/networking tools, and social networks appropriately to access, manage, integrate, evaluate, and create information to function in a knowledge economy successfully;
   3. demonstrate using current and new technologies specific to the program of study, course, and/or industry; and
   4. apply a fundamental understanding of the ethical/legal issues surrounding the access and use of information technologies.