AP Computer Science - A Overview

Content Area: **Computer Science & Business**
Course(s): **AP COMPUTER SCIENCE A**
Time Period:
Length: **180 Days**
Status: **Published**

Cover

EAST BRUNSWICK PUBLIC SCHOOLS

East Brunswick New Jersey

Superintendent of Schools

Dr. Victor P. Valeski

BOARD OF EDUCATION

Todd Simmens, President

Vicki Becker, Vice President

Susanna Chiu

Robert Cancro

Liwu Hong

Laurie Lachs

Barbara Reiss

Chad Seyler

Meredith Shaw

Course Adoption: 4/21/1986

Curriculum Adoption: 4/21/1986

Date of Last Revision Adoption: 9/1/2017

Course Overview

COURSE DESCRIPTION

The Advanced Placement Computer Science course is a full year introductory course in college level Computer Science. Students taking the course are required to have taken The Introduction to Computer Science course and math courses at the Algebra II level. Students taking the course will prepare to take the Advanced Placement Test in Computer Science. Practice exercises and problems from old and sample AP tests will be included a part of the course. The course centers on the development of computer programs to solve problems. The course emphasizes the design issues that make programs understandable, adaptable and reusable. The course uses an "object oriented" design paradigm. Important concepts include the development and analysis of algorithms, the development and use of data structures, the study of standard algorithms and typical applications. In addition, an understanding of the basic hardware and software components of computer systems and the responsible use of these systems are part of the course.

COURSE SCOPE AND SEQUENCE

Sequential Unit Description	Associated CPI's to be Achieved	Marking Period Guide	Other Pacing Guide References	Proficiency (Summative) Assessments
Unit 1 - Introduction to Computers and Java Language				
· **Components of a computer system (hardware and software)**	8.1.12.D.1			
· **A brief history of computing**	8.1.12.D.2			
· **Object Oriented Programming AP Computer Science and Java (an overview)**	8.1.12.D.58.2.2.E.1 8.2.2.E.2 8.2.8.E.2	1	6 days	- History report - Vocabulary assignment
1. encapsulation 2. inheritance 3. polymorphism .				
Unit 2 - Introduction to Program Development and Object Oriented Design Simple class design Introduction to standard Java classes (String class, Character class) ·Inputting data into a program(Scanner class) ·Using control structures to implement algorithms	N-Q (R) A-SSE (D) F-IF (D) S-IC (R) 8.2.2.E.1 8.2.2.E.2 8.2.2.E.3 8.2.2.E.4	1	25 days	- Programming projects - Online practice assignments - Unit tests

	8.2.5.E.2 8.2.5.E.3 8.2.8.E.3 8.2.12.E.3 N-Q (M) A-CED (M) A-REI(M) F-BF (D)			
Unit 3 -Using Numeric Data Types in a Program Use methods from the Integer and Double wrapper classes to convert String input to primitive numeric types Use the numeric operators and Math class methods and constants to perform calculations with numeric data Use the a Formatting class to format numeric data for output Design classes and algorithms to perform numeric calculations Use Math class random method to generate pseudo-random numeric values	8.2.5.E.1 8.2.5.E.2 8.2.5.E.3 8.2.2.E.3 8.2.2.E.4 8.2.5.E.1 8.2.5.E.2 8.2.8.E.3	1	16 days	- Programming projects - Online practice assignments - Unit tests
Unit 4- Use List interface and ArrayLists Class to Create and Process Linear Data Structures Use the Java library ArrayList class and methods to create and process a list of objects. Perform the standard operations (traversals, insertions, and deletions) on ArrayLists	8.2.12.E.3 N-Q 2 (D) N-VM(D) 8.2.2.E.1 8.2.2.E.2 8.2.2.E.3 8.2.2.E.4 8.2.5.E.3 8.2.8.E.3 8.2.12.E.3	1/2	13 days	- Programming projects - Unit tests

	N-Q 2 (I) A-SSE 3 (I) F-BF 1 (R) S-IC 2 (D) 8.2.2.E.1 8.2.2.E.2 8.2.2.E.3 8.2.2.E.4	2	20 days	- Programming projects - Unit tests
Unit 5- Writing Classes Using Inheritance and Interfaces. **Examine the use of inheritance in Java library classes.** **Examine a group of classes to demonstrate the concepts of inheritance.** **Have classes implement the Comparable interface to allow for comparisons of objects**				
	8.2.5.E.3 8.2.8.E.3 8.2.12.E.3 N-Q 1 (A) N-Q 2 (A) N-Q 3 (A) N-VM 6 (D) F-BF 2 (D) F-BF 3 (D) S-IC 2 (D) 8.2.2.E.1 8.2.2.E.2 8.2.2.E.3 8.2.2.E.4	2/3	19 days	- Programming projects - Unit tests
Unit 6- Using 1 and 2 Dimensional Arrays to Manipulate Data in Java **Use Java arrays to create linear lists of any data type.** Use 2-dimensional arrays to create tables/matrices of data **Perform the standard operations (traversals, insertions, and deletions) on one and two dimensional arrays.** Search and sort data in arrays-compare efficiencies of algorithms				
	8.2.5.E.3 8.2.8.E.3 8.2.12.E.3 A-SSE 1 (I) A-SSE 4 (I) F-BF 2 (I) F-BF 3 (I) 8.2.2.E.1 8.2.2.E.2 8.2.2.E.3	3	17 days	- Programming projects - Online practice assignments - Unit tests
Unit 7 -Using Recursive Algorithms **Simple recursive algorithms** **Recursive sorting and searching algorithms**				

	8.2.2.E.4 8.2.5.E.3 8.2.8.E.3 8.2.12.E.3 S-MD 1 (R) S-MD 5 (R) S-MD 6 (R) S-MD 7 (R)			
Unit 8- A.P. Computer Science Labs The Advanced Placement labs. The study will change over the years.	8.2.2.E.1 8.2.2.E.2 8.2.2.E.3 8.2.2.E.4 8.2.5.E.3 8.2.8.E.3 8.2.12.E.3 A-REI 1 (M)	3	18 days	- Programming projects - Unit tests
Unit 9 – Preparing for the Advanced Placement Computer Science test · Reviewing old AP Free response questions Reviewing available sample multiple choice questions	8.2.2.E.1 8.2.2.E.2 8.2.2.E.3 8.2.2.E.4 8.2.5.E.1 8.2.5.E.2 8.2.5.E.3 A-SSE 1 (I) A-SSE 4 (I) F-BF 2 (I) F-BF 3 (I)	3/4	12 days	- Programming projects - Practice AP tests - Unit tests
Unit 10 - Advanced Topics- (for after the AP Exam) **Student Projects**	8.2.2.E.1 8.2.2.E.2 8.2.2.E.3 8.2.2.E.4 8.2.5.E.1	4	25 days	- Programming projects - Online practice assignments - Unit tests

8.2.5.E.2

8.2.5.E.3

CONTENT FOCUS AREA AND COURSE NAME

Course Name: Advanced Placement Computer Science #1456

Course Number	School Numbers	Course Level	Grads(s)	Credits	Min. Per Week	Elective/Required	Initial Course Adopted
1456	050	AP	10-12	5.00	210	E	04/21/86

Textbooks and Other Resources

COURSE RESOURCES

Java an Introduction to Computer Science and Programming(Walter Savitch) Second Edition Prentice Hall

APCS Case Study and other materials posted on the AP website and other websites.

Be Prepared for the AP Computer Science Exam in Java (Maria Litvin) SkyLight Publishing

APCS review tests and test questions released from past years tests

Standards

MA.N-Q.A.1	Use units as a way to understand problems and to guide the solution of multi-step problems; choose and interpret units consistently in formulas; choose and interpret the scale and the origin in graphs and data displays.
MA.N-Q.A.2	Define appropriate quantities for the purpose of descriptive modeling.
MA.N-Q.A.3	Choose a level of accuracy appropriate to limitations on measurement when reporting quantities.
MA.F-BF.A.1	Write a function that describes a relationship between two quantities.
MA.F-BF.A.2	Write arithmetic and geometric sequences both recursively and with an explicit formula, use them to model situations, and translate between the two forms.

MA.F-BF.A.1a	Determine an explicit expression, a recursive process, or steps for calculation from a context.
MA.F-BF.A.1b	Combine standard function types using arithmetic operations.
MA.F-BF.A.1c	Compose functions.
MA.F-IF.A.1	Understand that a function from one set (called the domain) to another set (called the range) assigns to each element of the domain exactly one element of the range. If f is a function and x is an element of its domain, then $f(x)$ denotes the output of f corresponding to the input x. The graph of f is the graph of the equation $y = f(x)$.
MA.F-IF.A.2	Use function notation, evaluate functions for inputs in their domains, and interpret statements that use function notation in terms of a context.
MA.F-IF.A.3	Recognize that sequences are functions, sometimes defined recursively, whose domain is a subset of the integers.
MA.F-IF.C.8	Write a function defined by an expression in different but equivalent forms to reveal and explain different properties of the function.
MA.N-VM.C.6	Use matrices to represent and manipulate data, e.g., to represent payoffs or incidence relationships in a network.
MA.S-IC.A.1	Understand statistics as a process for making inferences about population parameters based on a random sample from that population.
MA.S-IC.A.2	Decide if a specified model is consistent with results from a given data-generating process, e.g., using simulation.
MA.S-IC.B.3	Recognize the purposes of and differences among sample surveys, experiments, and observational studies; explain how randomization relates to each.
MA.S-IC.B.4	Use data from a sample survey to estimate a population mean or proportion; develop a margin of error through the use of simulation models for random sampling.
MA.A-CED.A.1	Create equations and inequalities in one variable and use them to solve problems.
MA.A-CED.A.3	Represent constraints by equations or inequalities, and by systems of equations and/or inequalities, and interpret solutions as viable or nonviable options in a modeling context.
MA.A-CED.A.4	Rearrange formulas to highlight a quantity of interest, using the same reasoning as in solving equations.
MA.A-REI.A.1	Explain each step in solving a simple equation as following from the equality of numbers asserted at the previous step, starting from the assumption that the original equation has a solution. Construct a viable argument to justify a solution method.
MA.A-SSE.A.1	Interpret expressions that represent a quantity in terms of its context.
MA.A-SSE.A.2	Use the structure of an expression to identify ways to rewrite it. For example, see $x^4 - y^4$ as $(x^2)^2 - (y^2)^2$, thus recognizing it as a difference of squares that can be factored as $(x^2 - y^2)(x^2 + y^2)$.
MA.A-SSE.A.1a	Interpret parts of an expression, such as terms, factors, and coefficients.
MA.A-SSE.A.1b	Interpret complicated expressions by viewing one or more of their parts as a single entity.
MA.A-SSE.B.3	Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.
MA.A-SSE.B.4	Derive and/or explain the formula for the sum of a finite geometric series (when the common ratio is not 1), and use the formula to solve problems.
MA.A-SSE.B.3a	Factor a quadratic expression to reveal the zeros of the function it defines.
MA.A-SSE.B.3b	Complete the square in a quadratic expression to reveal the maximum or minimum value of the function it defines.
MA.A-SSE.B.3c	Use the properties of exponents to transform expressions for exponential functions.

TECH.8.1.12.D.1	Demonstrate appropriate application of copyright, fair use and/or Creative Commons to an original work.
TECH.8.1.12.D.2	Evaluate consequences of unauthorized electronic access (e.g., hacking) and disclosure, and on dissemination of personal information.
TECH.8.1.12.D.3	Compare and contrast policies on filtering and censorship both locally and globally.
TECH.8.1.12.D.4	Research and understand the positive and negative impact of one's digital footprint.
TECH.8.1.12.D.5	Analyze the capabilities and limitations of current and emerging technology resources and assess their potential to address personal, social, lifelong learning, and career needs.
TECH.8.1.12.E.1	Produce a position statement about a real world problem by developing a systematic plan of investigation with peers and experts synthesizing information from multiple sources.
TECH.8.1.12.E.2	Research and evaluate the impact on society of the unethical use of digital tools and present your research to peers.
TECH.8.1.12.E.CS1	Plan strategies to guide inquiry.
TECH.8.1.12.E.CS2	Locate, organize, analyze, evaluate, synthesize, and ethically use information from a variety of sources and media.
TECH.8.1.12.E.CS3	Evaluate and select information sources and digital tools based on the appropriateness for specific tasks.
TECH.8.1.12.E.CS4	Process data and report results.

Grading and Evaluation Guidelines

GRADING PROCEDURES

In terms of proficiency level the East Brunswick grades equate to:

A Excellent - Advanced Proficient

B Good Above Average - Proficient

C Fair - Proficient

D Poor - Minimally proficient

F Failing - Partially Proficient

COURSE EVALUATION

Each quarter students will be evaluated with tests and programming assignments using a total point basis to determine the quarter average. The semester average will be a weighted average of the 2 quarter averages (40% each) and a midterm exam (20%). The final average will be weighted of 4 quarters averages (20% each) the midterm (10%) and the final exam (10%)

Course achievement will be evaluated based on the percent of all pupils who achieve the minimum level of proficiency (final average grade) in the course. Student achievement levels above minimum proficiency will also be reported. Final grades, and where relevant mid-term and final exams, will be analyzed by staff for the total cohort and for sub-groups of students to determine course areas requiring greater support or modification.)

Other Details

10157 AP Computer Science A

Following the College Board's suggested curriculum designed to mirror college-level computer science courses, AP Computer Science A courses provide students with the logical, mathematical, and problem-solving skills needed to design structured, well-documented computer programs that provide solutions to real-world problems. These courses cover such topics as programming methodology, features, and procedures; algorithms; data structures; computer systems; and programmer responsibilities.