East Penn School District
Secondary Curriculum

A Planned Course Statement
for
AP Computer Science A

Course #365 Grade(s) 10-12

Department: Computer Science

Length of Period (mins.) 41 Total Clock Hours 123

Periods per Week 5 Length of Course (yrs.) 1

Type of Offering: _____required __x__ elective

CREDIT 1

Adopted 4/27/09

Developed by:
Carlen Blackstone
Beth Stoudt

Description of Course #365
Course Title: AP Computer Science A

Description: This course is designed for the college-bound student who wishes to pursue college-level studies while still in high school. This course will cover all of the curriculum as prescribed by the College Board for a one semester college course in computer science. Upon completion of this course, students will be prepared to take the AP Computer Science A level exam. Students will be expected to engage in rigorous problem solving activities and utilize computer resources outside of class.

Goals:

- To develop good programming methodology using control structures, object-oriented design, independent modules, recursion when needed, array structures, string processing, input and output files, and efficient means of searching for and sorting data
- To solve interesting problems by using computer algorithms
- To gain exposure to college-level computer science material and possibly gain college credit by performing well on the corresponding AP exam that is given in May

Requirements:

Prerequisite: Programming Foundations (recommended 84% or better), Algebra II (recommended 84% or better), or permission of instructor.

Text:

Lewis, John, Loftus, William, Cocking, Cara. Java Software Solutions for AP* Computer Science A, Pearson Education, Inc., c2006..

Key to Levels of Achievement (listed with each learning objective)

Awareness (A) Students are introduced to concepts, forms, and patterns.

Learning (L) Students are involved in a sequence of steps and practice activities which involve further development and allow for evaluation of process.

Understanding (U) Students demonstrate ability to apply acquired concepts and skills to individual assignments and projects on an independent level.

Reinforcement (R) Students maintain and broaden understanding of concepts and skills to accomplish tasks at a greater level of sophistication.

Unit	Num	Objective	Level	Content	Evaluation	Standard
I. Computer Science, Architecture, and Languages	1	Students will identify the various aspects of computer science and know what it means to study the field of computer science.	L	● Definition of Computer Science ● Ethical use of computers ● Scope of computer science field	Unit Test	ISTE 5.a ISTE 6.a
	2	Students will identify the various parts of computer architecture, explain their function, and understand how computers represent data and instructions.	L	● Configuration of hardware and functionality of each part ● Understanding of networks ● Binary representation of data, addresses, and instructions	Unit Test	ISTE 6.a
	3	Students will analyze high-level language development historically, understand the nature of assembly language, and know why to study Java.	R	● Development and characteristics of various high-level languages ● Writing simple assembly language code ● Euclid’s Algorithm	Homework assignments Written paper Unit Test	ISTE 6.a
	4	Students will be introduced to Java and object-oriented programming through role-play and modifying existing classes	R	● Structured programming concepts ● Object-oriented programming concepts ● Structure of Java programming language	Role-play in class Programming project - Picture class using shapes Unit Test	ISTE 1.b ISTE 4.b
II. Introduction to Java Control Structures	5	Students will know the difference between primitives and objects in Java and be able to understand pre-written Java code.	L	● Karel the Robot – already created programs ● Using basic robot methods ● Creating worlds	Classroom activities Unit Test	ISTE 1.a ISTE 4.d
	6	Students will know and use the three basic control structures of any language and see how to implement them in Java using blocks of code.	U	● Karel the Robot – creating your own programs ● Sequential statements ● Loop structures ● Decision structures	Classroom activities Programming project - Karel SquareMaker - Karel Harvester Unit Test	ISTE 1.b ISTE 4.b
	7	Students will devise an intelligent task for a robot to complete and write the code to complete it using control structures and new methods if needed.	R	● RacerRobot class example ● Defining new methods such as turnRight, faceNorth, isBlockedRight, raceStride	Classroom activities Programming project - Karel Steeple Chaser Unit Test	ISTE 1.b ISTE 4.b
III. Sequential Structures in Java	8	Students will learn how to declare variables and do input and output in Java.	U	● Declaring primitive types ● Using the Scanner class to input data ● Using System.out.println with formatting	Unit Test	ISTE 3.b ISTE 4.a
	9	Students will evaluate and write assignment statements with arithmetic expressions using	U	● Arithmetic operators (+,-,*, /, %) ● Boolean comparisons	Unit Test	ISTE 3.b ISTE 4.a

		various operators and functions in Java with type-casting when needed.		(<,<=,>,>=, ==,!=).and operators (&&,||,!) ● Short-cut assignment operators (=, +=, -=, *=, /=, %=, ++, --) ● Math functions (abs, sqrt, pow) ● Mixed-mode operations and type-casting		
	10	Students will know when and how to use constants and the difference between integer and real data in terms of exactness and storage.		● Using MAX_INT as a limit ● Using **final** to represent a user-defined constant ● Round-off error with **double**	Unit Test	ISTE 3.b ISTE 4.a
	11	Students be able to write a Java program that inputs, calculates, and outputs information using correct data types, formatting, prompts, type-casting, spacing, and indentation.	U	● Creating a main method to run a program ● Designing an algorithm using pseudo-code ● Editing, debugging, and testing a program	Programming project - Geometric Figures Take-home Program - Simple GPA Calculation	ISTE 1.b ISTE 4.b

Unit	Num	Objective	Level	Content	Evaluation	Standard
IV. Loop and Decision Structures in Java	12	Students will use the **for** statement for definite loops as well as the **while** and **do…while** statements for indefinite pre-test and post-test loops respectively in a Java program.	U	● The **for** statement for definite loop structures ● The **while** and **do…while** statements ● Counter-controlled, sentinel-controlled, and query-controlled loops for more than one data set	Programs to Present - Compound Interest - Grade Report - Pascal’s Triangle Unit Test	ISTE 3.b ISTE 4.a
	13	Students will use one-way, two-way, or multi-way **if. . .then . . . else** statements as well as the **switch, case,** and **break** statements to make decisions in a Java program.	U	● One-way, two-way, or multi-way **if. . .then . . . else** statements ● **switch, case,** land **break** statements ● writing blocks of code for multiple statements	Programs to Present - Calculator Program - Payroll Report - GPA Report Unit Test	ISTE 3.b ISTE 4.a
	14	Students will use nested selection and repetition statements in writing Java programs in order to create a larger project with well-structured code.	R	● Designing the structure for a larger program ● Designing the algorithms needed from pseudo-code and flowcharts ● Editing, debugging, and testing a larger program in an efficient way	Programming project - Factoring, Primes, Prime Factorization, and GCF/LCM Take-home Program - Quadratic Formula	ISTE 1.b ISTE 4.b
V. Defining Your Own Classes in Java	15	Students will know the concepts of encapsulation, public methods, private methods, constructors,	L	● Encapsulation and information hiding ● Private attributes and public	Unit Test	ISTE 3.b ISTE 4.a

		accessors and mutators, final and static, this and null, both call-by-value and call-by-reference parameters, and overloading		methods ● Constructors and overloading ● Accessors and mutators ● Call-by-value and call-by-reference parameters		
	16	Students will write a Driver program to use existing classes and create random numbers.	U	● Generating random numbers ● Using static variables and methods ● Declaring objects in a client program ● Calling methods from a client program	Programming project - Dice Take-home Program - Bank Account	ISTE 1.b ISTE 4.b
	17	Students will modify methods of an existing class to change functionality.	U	● Changing private attributes ● Changing public methods ● Using parameters when needed ● Using the **this** and **null** keywords ● Creating overloaded constructors and methods	Programming project - Frog1D becomes Frog2D Take-home Program - Bank Account	ISTE 1.b ISTE 4.b
	18	Students will design their own class for scratch.	R	● Creating private attributes ● Creating public methods ● Creating overloaded and static constructors and methods ● Using the **this** and **null** keywords	Programming project - Complex Numbers Take-home Program - Bank Account	ISTE 1.b ISTE 4.b
VI. One-dimensional Arrays, Strings, and ArrayLists	19	Students will input, output, and process one-dimensional arrays while understanding how they are stored internally.	U	● Dr. Seuss' "Too Many Daves" ● Inputting values into an array ● Printing a table from an array ● Accumulating and reversing	Programs to Present - Standard Deviation - The Game of Reverse Unit Test	ISTE 1.b ISTE 4.b
Unit	Num	Objective	Level	Content	Evaluation	Standard
	20	Students will use arrays to solve problems including table-look up and frequency distributions.	U	● Table-lookup algorithms ● Frequency distribution algorithms	Programs to Present - Calculating Postal Rates - Calculating Wages - Tallying Percentages - Tallying Voting Results Unit Test	ISTE 1.b ISTE 4.b
	21	Students will process either unordered or ordered safe arrays by searching, sorting, merging, inserting, and deleting and reversing.	R	● Linear and binary searching ● Max-min method of sorting ● Adding or removing from an unordered vs. an ordered array ● Reversing an array ● Merging two arrays into one ● Exception handling for too many array elements	Programs to Present - Reverse Game - Merging Runs of Data Group Programming Project - School Store Inventory Unit Test	ISTE 1.b ISTE 4.b

	22	Students will use arrays to process String objects.	U	● Using the **length, charAt, toUpperCase**, and **substring** methods ● Inputting and traversing a string of characters	Programs to Present - Letters in a Sentence - Unique Words in a String Unit Test	ISTE 1.b ISTE 4.b
	23	Students will understand the nature of collections of objects and be able to use the **ArrayList** collection's methods to accomplish the same tasks as a static array.	U	● Java Collections ● Declaring **ArrayList** objects ● Using the **get, set, remove, isEmpty, size**, and **add** methods in the **ArrayList** class	Programs to Present - Student phone numbers - Vectors of Frog2D Group Programming Project - School Store Inventory Unit Test	ISTE 1.b ISTE 2.d ISTE 4.b
VII. Sorting Algorithms	24	Students will know how to trace, analyze, and use the Selection, Insertion, and Bubble sorts and understand $O(n^2)$ time.	U	● Selection sort ● Insertion sort ● Bubble sort	Student presentations and programs Unit Test	ISTE 2.b
	25	Students will know how to trace, analyze, and use the Mergesort, Quicksort, and Heapsort and understand $O(n*log_2 n)$ time.	U	● Merge sort ● Quick sort ● Heap sort	Student presentations and programs Unit Test	ISTE 2.b
	26	Students will know the differences between the above sorts and the Shell sort, Radix sort, Bucket sort, Pointer sort, Merge Bubble sort, and External File sort.	U	● Shell sort ● Radix sort ● Bucket sort ● Pointer sort ● Merge Bubble sort ● External File sort	Student presentations and programs Unit Test	ISTE 2.b
VIII. Interfaces, Inheritance, and Abstract Classes	27	Students will understand the purpose of interfaces, inheritance, and abstract classes and know the concepts and Java keywords needed to implement them.	L	● Interfaces – contract with implementing classes ● Inheritance – Using **super** for an "is a" relationship ● Abstract classes – using inheritance to define more general methods and defining abstract methods at a lower level	Homework Assignment - Create a hierarchical design - Read the GridWorld case study student manual Unit Test	ISTE 3.b ISTE 4.a ISTE 6.d
	28	Students will use existing **Comparable**, **List**, and **ListIterator** interfaces and create an interface of their own.	U	● Implementing the **Comparable** interface with a **compareTo** method ● Iterating through an **ArrayList** and using enhanced For loops ● GridWorld related code	Programming Project - Modify the Inventory project Unit Test	ISTE 2.d ISTE 3.b ISTE 4.a ISTE 6.d
Unit	Num	Objective	Level	Content	Evaluation	Standard
	29	Students will write inherited classes from an existing project with already-defined base classes.	U	● Defining a Wheel class to extend a Circle class ● Defining different kinds of	Programming Project - Creating a Triangle, RegularPolygon, Square1,	ISTE 1.b ISTE 4.b ISTE 6.d

				Bugs to extend the Bug class	and Square2 class Unit Test	
	30	Students will override methods of “super” classes and define abstract methods of “super” abstract classes.	U	● Abstract classes defined methods ● Abstract methods to be defined by sub-classes	Programming Project - Creating a Triangle, RegularPolygon, Square1, and Square2 class Unit Test	ISTE 1.b ISTE 4.b ISTE 6.d
	31	Students will add objects to the GridWorld case study by using inheritance and overriding methods.	R	● Creating a Critter class and different kinds of Critters ● Knowing how all of the classes in GridWorld relate to each other	Programming Project - Add a new class of objects to the GridWorld case study Unit Test	ISTE 1.b ISTE 4.b ISTE 6.d
IX. Two-dimensional Arrays, Files, and Recursion	32	Students will input, output, and process two-dimensional arrays while understanding how they are stored internally.		● Inputting two-dimensional arrays ● Outputting a table of two-dimensional array data ● Looking up, tallying, or accumulating results in a two-dimensional array	Programs to Present - Magic Square - On-line Voting Take-home Program - Weighted Grades - Gauss-Jordan Method for Solving NxN Systems Unit Test	ISTE 1.b ISTE 4.b ISTE 6.d
	33	Students will read data from a text file using the Scanner and File classes with a **try {} catch {}** block of code for exception handling.		● Using an input file with the Scanner class ● Inputting the name of the file ● Making sure the file exists ● Checking for the end of the file	Programs to Present - Bowling Scores - Survey Results Unit Test	ISTE 1.b ISTE 4.b ISTE 6.d
	34	Students will know how to write some common math functions recursively.		● Tracing recursive functions ● Writing recursive functions from the definition ● Creating recursive definitions	Programs to Present - Factorial, Euclid’s GCF, Fibonacci, Power, Newton’s Root, ConvertToBase, Permutations, and Continuous Fractions Unit Test	ISTE 1.b ISTE 4.b ISTE 6.d
	35	Students will know how to use recursion to solve problems involving backtracking and decomposition.		● Decomposition example - Tower of Hanoi ● Backtracking example – Eight Queens	Analysis Assignments Programming Project - Mouse in a Maze, Boggle, or Sudoku Unit Test	ISTE 1.b ISTE 4.b ISTE 6.d
X. ACSL	36	Students will participate in the ACSL competition throughout the year by learning the concepts and writing the programs required.		● Computer Number Systems, Recursive Functions, Lisp, Boolean Algebra, Bit String Flicking, Digital Electronics, Graph Theory, Data Structures, and Prefix/Infix/Postfix	Short Problems Test (4 per year) Programming Assignment (4 per year)	ISTE 1.a ISTE 3.d ISTE 4.b