

Syllabus Development Guide: AP Computer Science A

To the AP teacher: Please take full advantage of this guide. It is designed to support you as you develop your syllabus for the AP Course Audit. The guide contains the following sections and information:

Curricular Requirements	The curricular requirements are the core elements of the course. Your syllabus must provide clear evidence that each requirement is fully addressed in your course.	Important Considerations	Aligned with the Evaluation Guidelines, these statements provide advice on the type of evidence your syllabus should include.
Scoring Components	Some curricular requirements consist of complex, multi-part statements. These particular requirements are broken down into their component parts and restated as “scoring components”. Reviewers will look for evidence that each scoring component is included in your course.	Reference	As appropriate, references to specific sections of the official AP Course Description or other pertinent publications are included here.
Key Terms	To ensure the clarity of certain terms or expressions that may have multiple meanings, each of these terms is clearly defined.	Samples of Evidence	For each scoring component, three separate samples of evidence are provided. These statements provide either verbatim samples from actual authorized syllabi or clear descriptions of what acceptable evidence should look like.
Evaluation Guidelines	These are the exact guidelines used by reviewers as they evaluate the evidence in your syllabus. Use these to interpret any requirement you may find ambiguous.		

Curricular Requirements	Scoring Components, Key Terms, Evaluation Guidelines, Important Considerations, References and Samples of Evidence
Curricular Requirement: The course includes all of the topics listed in the "Computer Science A" column of the Topic Outline in the AP Computer Science Course Description.	Scoring Component: Please see the scoring components below for detail.

Scoring Component 1*: The course teaches students to design and implement computer-based solutions to problems				
*Note Each Curricular Requirement may be subdivided into two or more distinct Scoring Components.				
Key Term(s)	Evaluation Guideline(s)	Important Consideration(s)	Reference	
<p>Curricular Requirement: The course teaches students to design and implement computer-based solutions to problems in a variety of application areas</p>	<p>Design: defining the class(es) with their respective attributes, constants and operations, the interactions among classes and the algorithms that are required to solve a specific problem.</p> <p>Implement: creating those classes by writing the Java code and running and testing the solution designed.</p>	<p>The syllabus must include explicit evidence that students engage in regular, frequent practice writing programs. Programming activity (assignments/labs/projects) must be included in most units, roughly 75% of the time.</p> <p>Representative examples of assignment titles and/or short descriptions must be included in the syllabus</p> <p>A list of textbook problem numbers alone is not sufficient evidence. For example, "Various assignments from chapter 3" is not sufficient evidence; likewise, "Problems 8.2, 8.3, 8.4" is not sufficient evidence.</p>	<p>The syllabus must demonstrate that students design solutions to problems using object-oriented constructs, data structures and algorithms; that students implement, document and test those solutions; and that students learn how to solve problems by creating working software.</p> <p>Evidence of design could include exercises in writing, documenting (for example, by using Javadoc, commenting, pre/post conditions, and/or assertions), and testing programs.</p> <p>Evidence of implementation could include exercises and problems that involve coding object-oriented constructs, data structures, and algorithms.</p>	<p>For additional information see "I. Object-Oriented Program Design" and "II. Program Implementation" in the Computer Science Course Description.</p>
	Samples of Evidence			
	Sample 1	Sample 2	Sample 3	
<p>The syllabus includes a variety of problems involving interacting classes under each unit of study, including Lottery, Slot Machine, Pac Fish, Asteroids, Spell Checker, and Google Billboard.</p>	<p>The syllabus includes a variety of projects involving interacting classes integrated throughout the course including a project where students design and implement an application for a registrar's office. This application has classes Student, Course, CourseSection and Instructor.</p>	<p>The syllabus includes a "Lab Assignment" section under each unit of the course which includes the name and short description of each lab.</p>		

Scoring Component 2: The course teaches students to use and implement commonly used algorithms.				
Curricular Requirement: The course teaches students to use and implement commonly used algorithms and data structures.	Key Term(s)	Evaluation Guideline(s)	Important Consideration(s)	Reference
	<p>Use: knowing how to integrate classes and methods written by someone else (the Java implementers, the teacher, or the textbook author, for example).</p> <p>Implement: knowing how to write and test Java classes and methods "from scratch" - translating algorithms described without Java (In English or mathematics, for example) into working Java code.</p>	<p>The syllabus must include evidence of instruction for</p> <ul style="list-style-type: none"> • Operations on one-dimensional arrays • Sequential and binary search • Insertion, selection, and merge sorts 	<p>Sorting algorithms might be individually listed, or they might be indicated by phrases such as "elementary and recursive sorting" or "quadratic and sub-quadratic sorts" or "O(n-squared) and O(n log n) sorts".</p>	<p>For additional information see "V. Standard Algorithms" in the AP Computer Science Course Description.</p>
	Samples of Evidence			
	Sample 1	Sample 2	Sample 3	
	<p>The syllabus:</p> <ul style="list-style-type: none"> • Includes a lesson on "Simple (iterative) sorts and recursive sorts" • Includes a lesson on "Searching in ordered and unordered lists" • Describes a lesson in which students learn about "Insertion and deletion of elements in arrays and ArrayLists. Traversals of arrays and ArrayLists." 	<p>The syllabus:</p> <ul style="list-style-type: none"> • Includes a lesson on "Quadratic sorts and n log n sorts." • Includes a lesson on "Linear and binary searches." • Describes a lesson in which students learn about "construction, modification, and comparison of arrays and ArrayLists." 	<p>The syllabus lists "Java Concepts, 4th edition AP Edition, Cay Horstmann, 2005, John Wiley & Sons" as the main text book and includes topic and chapter titles throughout the syllabus including Chapter 8: Array and Array lists and 19: Sorting and Searching.</p>	

<p>Curricular Requirement: The course teaches students to use and implement commonly used algorithms and data structures.</p>	Scoring Component 3: The course teaches students to use and implement commonly-used data structures.			
	Key Term(s)	Evaluation Guideline(s)	Important Consideration(s)	Reference
	Use and implement: using ArrayLists and implementing and using one- and two-dimensional arrays.	The syllabus must contain some study of (one-and two-dimensional) arrays and ArrayLists	Scoring Component is clear and explicit. No Important Considerations are needed.	For additional information, see “IV. Standard Data Structures” in the AP Computer Science Course Description.
	Samples of Evidence			
	Sample 1	Sample 2	Sample 3	
	The syllabus lists exercises labeled as follows: <ul style="list-style-type: none"> • Practice with arrays. • Instantiate ArrayLists. • Storing numerical data in ArrayLists. • Use loops to manipulate arrays and lists. 	The syllabus lists exercises labeled as follows: <ul style="list-style-type: none"> • Storing objects in arrays and ArrayLists. • Storing numbers in arrays. 	The syllabus includes a description under each unit of study. Unit 16, Arrays and Array Lists, includes: <ul style="list-style-type: none"> • Topics: One-dimensional arrays: Traversals, Insertions, Deletions; Array Lists: Discuss similarities and differences between Array Lists and Arrays. • Labs: Pascal’s Triangle, Student Roster, WordPatterns, Anagrams 	

Curricular Requirement: The course teaches students to develop and select appropriate algorithms and data structures to solve problems.	Scoring Component 4: The course teaches students to select appropriate algorithms and data structures to solve problems.			
	Key Term(s)	Evaluation Guideline(s)	Important Consideration(s)	Reference
	Select appropriate: determining which among several alternative algorithms or data structures is more appropriate for specific applications.	The syllabus must provide evidence of instruction of <i>comparison</i> of algorithms and data structures to solve problems.	An informal comparison of running time is sufficient to meet this component (e.g., the course could present a side-by-side comparison of algorithms for sorting or choose to introduce “Big-Oh” analysis).	For additional information, see "F. Reason About Programs" and "G. Analysis of Algorithms" and “III. Program Analysis” in the AP Computer Science Course Description.
	Samples of Evidence			
	Sample 1	Sample 2	Sample 3	
	The syllabus includes <ul style="list-style-type: none"> • A class discussion of when and when not to use recursion. • A discussion of when an array is more appropriate than an ArrayList. 	The syllabus includes a comparison of the time required <ul style="list-style-type: none"> • For the execution of several sorting algorithms • For inserting and deleting from arrays and ArrayLists 	The syllabus includes the following sub-topics under various units <ul style="list-style-type: none"> • Informal comparison of algorithm running times • Exact calculation of execution counts • Recursion • Inefficient recursion 	

<p>Curricular Requirement: The course teaches students to code fluently in an object-oriented paradigm using the programming language Java. The course teaches students to use standard Java library classes from the AP Java subset delineated in Appendixes A and B of the AP Computer Science Course Description. (Note: Students who study a language other than Java in AP Computer Science must also be taught to use Java, as specified in the AP Java subset.)</p>	<p>Scoring Component 5: The course teaches students to code fluently in an object-oriented paradigm using the programming language Java.</p>			
	<p>Key Term(s)</p>	<p>Evaluation Guideline(s)</p>	<p>Important Consideration(s)</p>	<p>Reference</p>
	<p>Object-oriented paradigm: applying concepts of encapsulation, inheritance, and polymorphism in the solutions of problems; learning to model components of a problem and its solution with objects; learning how to use given, existing classes together with new classes.</p>	<p>The syllabus must include an assignment/project that involves implementing a solution using a set of classes that are related by inheritance.</p> <p>If the syllabus provides evidence of instruction for the GridWorld Case Study, then this scoring component is met.</p> <p>Students who study a language other than Java during an AP Computer Science course must also be taught to use Java, as specified in the AP Java subset.</p>	<p>Scoring Component is clear and explicit. No Important Considerations are needed.</p>	<p>No references to external documents are needed for this Scoring Component.</p>
	<p>Samples of Evidence</p>			
	<p>Sample 1</p>	<p>Sample 2</p>	<p>Sample 3</p>	
	<p>The syllabus includes the following exercise: Extend a given BankAccount class by writing a SavingsAccount class. The BankAccount class provides methods for inquiring about the balance in the account, making deposits, and making withdrawals. The SavingsAccount class also allows inquires, deposits, and withdrawals but, unlike the general kind of account, updates the balance during each operation by computing accumulated interest.</p>	<p>The syllabus includes exercises from the GridWorld Case Study extending the Bug and Critter classes.</p>	<p>In addition to the GridWorld Case Study, the syllabus includes units on encapsulation, Inheritance, and Interfaces. Each unit includes the following sections: Topics, Student Objectives, Guided Practice, Labs, and Assessments.</p>	

<p>Curricular Requirement: The course teaches students to code fluently in an object-oriented paradigm using the programming language Java. The course teaches students to use standard Java library classes from the AP Java subset delineated in Appendixes A and B of the AP Computer Science Course Description. (Note: Students who study a language other than Java in AP Computer Science must also be taught to use Java, as specified in the AP Java subset.)</p>	<p>Scoring Component 6: The course teaches students to use standard Java library classes from the AP Java subset delineated in Appendixes A and B of the AP Computer Science Course Description.</p>			
	<p>Key Term(s)</p>	<p>Evaluation Guideline(s)</p>	<p>Important Consideration(s)</p>	<p>Reference</p>
	<p>All terminology in the Scoring Component is clear. No clarification is needed.</p>	<p>The syllabus must provide evidence of instruction of standard Java library classes, methods, and interfaces from the AP Java subset.</p> <p>If the syllabus mentions the use of the AP Computer Science Quick Reference Guide in the course then this scoring component is met.</p> <p>If the syllabus mentions the use of the Appendix B of the AP Computer Science Course Description in the course then this scoring component is met.</p>	<p>Evidence may be inferred that the course introduces students to the standard Java library classes, methods, and interfaces from the AP Java subset through examples, e.g., lectures, study topics, labs, assignments and exercises in the syllabus (see examples below).</p>	<p>For additional information see Appendixes A and B in the AP Computer Science Course Description.</p>
	<p>Samples of Evidence</p>			
	<p>Sample 1</p>	<p>Sample 2</p>	<p>Sample 3</p>	
<p>The syllabus includes an exercise that asks students to override the definition of the toString() method. This use of a method that is inherited from the Object class is evidence that the course introduces students to the Object class, as well as the String class.</p>	<p>The syllabus includes an exercise that asks students to write a program that computes the square root of a given number through an application of Newton's method. This implicitly calls upon students to learn about the Math.abs(), Math.sqrt(), and Math.pow() methods.</p>	<p>The syllabus lists an exercise that asks students to define a class named "Person." Instances of the class contain a name, address, telephone number, and date of birth. The exercise asks students to sort a list of Persons by age. This is evidence that the course introduces students to the Comparable interface.</p>		

<p>Curricular Requirement: The course teaches students to read and understand a large program consisting of several classes and interacting objects, and enables students to read and understand the current AP Computer Science Case Study posted on AP Central.</p>	<p>Scoring Component 7: The course teaches students to read and understand a large program consisting of several classes and interacting objects. In particular, the course enables students to read and understand the current AP Computer Science Case Study posted on AP Central.</p>			
	Key Term(s)	Evaluation Guideline(s)	Important Consideration(s)	Reference
	<p>Current AP Computer Science Case Study: GridWorld Case Study</p>	<p>The syllabus must include Parts 1 – 4 of the GridWorld Case Study.</p>	<p>The GridWorld Case Study can be integrated throughout the course either as part of instruction on data structures or by including assignments from the case study at various stages of the course. It is not necessary that an entire unit be spent on the GridWorld Case Study.</p>	<p>For additional information see “Case Studies” in the AP Computer Science Course Description. In addition, the Gridworld Case Study can be downloaded from AP central at http://apcentral.collegeboard.com/apc/public/courses/teachers_corner/4483.html</p>
	<p>Samples of Evidence</p>			
	Sample 1	Sample 2	Sample 3	
	<p>Students read and solve problems posed in Parts 1 through 4 of the GridWorld Case Study.</p>	<p>The syllabus includes lessons labeled as follows:</p> <ul style="list-style-type: none"> • Running the GridWorld simulation. • Writing programs that create and populate a grid. • Defining new kinds of critters and new behaviors for critters. • Designing critters that can interact with one another. 	<ul style="list-style-type: none"> • The syllabus indicates that students become familiar with the code supplied as part of the GridWorld Case Study. • The syllabus indicates that students write programs that modify and extend the supplied code. 	

Scoring Component 8: The course teaches students to recognize the ethical and social implications of computer use.				
Key Term(s)	Evaluation Guideline(s)	Important Consideration(s)	Reference	
<p>Curricular Requirement: The course teaches students to recognize the ethical and social implications of computer use.</p>	<p>The syllabus must include explicit evidence of how ethical or social implications of computer use are addressed within the course. If the syllabus does not identify an activity (paper, presentation, etc.), then it must at least identify topics of discussion in the course schedule (protection of privacy, intellectual property, public safety, etc.).</p> <p>If the syllabus states that this topic was addressed in another course, then it must also include content and materials taught in that course (e.g., a textbook citation and chapter titles or a descriptive narrative).</p> <p>Optional or extra credit assignments on ethical and social implications of computer use are not sufficient evidence.</p> <p>Reference to an “acceptable use of school computer” policy alone is not sufficient evidence.</p>	<p>Evidence that the course teaches students to recognize the ethical and social implications of computer use can be illustrated in a variety of ways, including but not limited to scheduled lessons or discussion of “ethical use of technology,” “responsible use of technology,” or discussions of one or more of topics like the following:</p> <ul style="list-style-type: none"> • Respect for/protection of privacy • Respect for/protection of intellectual property • Defense against vandalism, fraud, and other kinds of crime on the Internet • Potential of new technologies to affect public safety • Potential of new technologies to affect employment • Liability of software engineers for defects in their products • Inclusion, free speech, Section 508 compliance (US law that requires equitable access) 	<p>For additional information see “VI. Computing in Context” in the AP Computer Science Course Description.</p>	
	Samples of Evidence			
	Sample 1	Sample 2	Sample 3	
<p>The syllabus includes discussions of copyright law, software piracy, intellectual property, privacy, and network reliability in the schedule of lessons.</p>	<p>The syllabus includes a day on which students view and discuss a video (such as Triumphs of the Nerds) together in the schedule of lessons.</p>	<p>The syllabus calls for students to address a significant question related to ethical use of technology in a paper.</p>		