Syllabus Development Guide: AP® Computer Science A

The guide contains the following sections and information:

Curricular Requirements	The curricular requirements are the core elements of the course. The syllabus must provide clear evidence that each requirement is fully addressed in the course.
Scoring Components	Some curricular requirements consist of complex, multipart 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.
Evaluation Guideline(s)	These are the evaluation criteria that describe the level and type of evidence required to satisfy each scoring component.
Key Term(s)	These ensure that certain terms or expressions, within the curricular requirement or scoring component that may have multiple meanings, are clearly defined.
Examples For each scoring component, three separate samples of evidence are provided. These statements provide clear descriptions of what acceptable evidence should look like.	

Syllabus Development Guide: AP® Computer Science A

Curricular Requirement 1	The course teaches students to design and implement computer-based solutions to problems.
Evaluation Guideline(s)	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 percent of the time. Representative examples of assignment titles and/or short descriptions must be included in the syllabus. 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.
Key Term(s)	**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. **Implement:** creating those classes by writing the Java code and running and testing the solution designed.

Samples of Evidence

1. 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.

2. The syllabus includes a variety of projects involving interacting classes including a project where students design and implement an application for a registrar's office. This application has classes `Student`, `Course`, `CourseSection`, and `Instructor`.

3. The syllabus includes a "Lab Assignment" section under each unit of the course that includes the name and a short description of each lab, such as:

   - Inches to Miles, or Milliseconds to Hours: Write a program that converts a number of inches to miles, yards, feet, and inches. Alternatively, write a program that converts a number of milliseconds to hours, minutes, seconds, and milliseconds.
   - Implement a `CashRegister` class that includes the methods `getPayment` and `giveChange`. Write a client program to test your implementation.

Curricular Requirement 2	The course teaches students to use and implement commonly used algorithms and data structures.
Scoring Component 2a	The course teaches students to use and implement commonly used algorithms.
Evaluation Guideline(s)	The syllabus must include evidence of instruction for • Operations on one-dimensional arrays • Sequential and binary search • Insertion, selection, and merge sorts
Key Term(s)	**Use:** knowing how to integrate classes and methods written by someone else (the Java implementers, the teacher, or the textbook author, for example). **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.

Samples of Evidence

1. The syllabus
   - Includes lessons on insertion, selection, and merge 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`, and traversals of arrays and `ArrayLists`.
2. The syllabus
   - Includes a lesson on insertion, selection, and merge 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`.
3. The syllabus includes compare and contrast activities for
   - Arrays and `ArrayLists.`
   - Sequential and binary search.
   - Insertion, selection, and merge sorts.

Curricular Requirement 2	The course teaches students to use and implement commonly used algorithms and data structures.
Scoring Component 2b	The course teaches students to use commonly used data structures.
Evaluation Guideline(s)	The syllabus must contain study of one- and two-dimensional arrays and `ArrayLists`.
Key Term(s)	None at this time.

Samples of Evidence

1. The syllabus lists exercises labeled as follows:
   - Practice with arrays.
   - Instantiate `ArrayLists`.
   - Storing numerical data in `ArrayLists`.
   - Use loops to traverse two-dimensional arrays.
2. The syllabus lists exercises labeled as follows:
   - Storing objects in arrays and `ArrayLists`.
   - Storing numbers in arrays.
   - Two-dimensional arrays.
3. The syllabus includes the following subtopics under various units:
   - Discussion of traversals, insertions, and deletions in arrays.
   - Comparison of arrays and `ArrayLists`.
   - Practice in lab with exercises titled Pascal's Triangle, Magic Squares, Word Patterns, Student Roster, Anagrams, and Picture.

Curricular Requirement 3	The course teaches students to select appropriate algorithms and data structures to solve problems.
Evaluation Guideline(s)	The syllabus must provide evidence of comparison of algorithms to solve problems.
Key Term(s)	**Select appropriate:** determining which among several alternative algorithms or data structures is more appropriate for specific applications.

Samples of Evidence

1. The syllabus includes
   - A class discussion of when and when not to use recursion.
   - A discussion of when an array is more appropriate than an `ArrayList`.
2. The syllabus includes a comparison of the times required
   - For the execution of several sorting algorithms.
   - For inserting and deleting from arrays and `ArrayLists`.
3. The syllabus includes the following subtopics under various units:
   - Informal comparison of algorithm running times.
   - Exact calculation of execution counts.
   - Recursion versus iteration.
   - Compare and contrast arrays and `ArrayLists`.

Syllabus Development Guide: AP® Computer Science A

Curricular Requirement 4	The course teaches students to code fluently in an object-oriented paradigm using the programming language Java.
Evaluation Guideline(s)	The syllabus must describe an assignment or activity that involves implementing a solution using inherited methods or overriding inherited methods.
Key Term(s)	**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.

Samples of Evidence

1. 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.

2. The syllabus includes an activity where students are given a class hierarchy and will write complete class declarations.

3. In addition to the three AP Computer Science A Labs, the syllabus includes abstraction, encapsulation, inheritance, and polymorphism.

Curricular Requirement 5	The course teaches students to use elements of the standard Java library from the AP Java subset in Appendix A of the AP Computer Science A Course Description.
Evaluation Guideline(s)	The syllabus must provide evidence of instruction in at least two classes from the AP Java subset.
Key Term(s)	None at this time.

Samples of Evidence

1. Students define a class that overrides the definition of the `toString()` method that the class inherits from the `Object` class.
2. Students write a program that manipulates text and a program that computes a numerical result using mathematical functions (for example, square roots, logarithms, or sines).
3. Students write a method that returns a randomly selected object from a list. This implicitly uses the `Math` class and the `List` interface.

Curricular Requirement 6	The course includes a structured lab component comprised of a minimum of 20 hours of hands-on lab experiences.
Evaluation Guideline(s)	The syllabus must include an explicit statement that at least 20 hours of instructional time is spent in hands-on laboratory experiences.
Key Term(s)	None at this time.

Samples of Evidence

1. The syllabus includes the following statement: "Students are engaged in hands-on laboratory experiences, integrated throughout the course, which account for 20 hours of course time."

2. The syllabus states that the total hands-on lab time will be a minimum of 20 hours of total instructional time, with labs distributed throughout the course.

3. The syllabus includes a statement to indicate that students spend a minimum of 20 hours of the course engaged in hands-on labs.

Curricular Requirement 7	The course teaches students to recognize the ethical and social implications of computer use.
Evaluation Guideline(s)	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.). Optional or extra credit assignments on ethical and social implications of computer use are not sufficient evidence. Course goals and/or objectives alone are not sufficient evidence. Reference to an "acceptable use of school computer" policy alone is not sufficient evidence.
Key Term(s)	**Recognize the ethical and social implications**: any activity that engages students in discussions of principles for the responsible use of computers.

Samples of Evidence

1. The syllabus includes the following topics in the course schedule under Unit 1: copyright law, software piracy, intellectual property, privacy, and network reliability.

2. The syllabus calls for students to read stories in the news about social changes and ethical challenges that advances in computing are producing, and respond to what they read in journals.

3. The syllabus includes a day on which students view and discuss a relevant video (such as *Triumph of the Nerds*) together in the schedule of lessons.