

# AP<sup>®</sup> Computer Science AB

## Syllabus 4

### Course Overview

I teach AP<sup>®</sup> Computer Science over a period of three years. All ninth-grade students are required to take Pre-AP Computer Science, which covers the first portion of the AP Computer Science A curriculum. Then students can optionally take AP Computer Science A, which focuses on the remainder of the A curriculum. Some AP Computer Science AB topics, such as two-dimensional arrays and iterators, are also introduced in the AP Computer Science A course. Students who have completed the AP Computer Science A course can then opt to take AP Computer Science AB.

The content and objectives of the AP Computer Science AB course include the course objectives for AP Computer Science A and AB as discussed in the *AP Computer Science Course Description*. The course concentrates on the implementation of data structures and the Java built-in classes for data structures. It includes a major focus on the AP Computer Science GridWorld Case Study.

This course enhances students' problem-solving abilities. It builds analytical skills that are valuable in computer science, in other courses, and in life. Of course, students also increase their computer science and programming skills—skills that are needed in an ever-increasing array of college courses and workplaces.

### Personal Philosophy

I encourage very active student participation in the classroom. It's great to see students solve problems, share solutions, and make discoveries. I also enjoy coaching our Computer Science Team. We meet weekly to prepare for the many written and team-programming competitions during the year. I get to see the students having fun and being rewarded for their work while they progress both technically and personally.

#### **I enjoy seeing students:**

- gain insight into and solve the many little puzzles they encounter in this course, such as how to reverse a linked list or find the middle of one with a single loop;
- take pride in their ability to solve significant problems using powerful Java data structures such as maps;
- learn that recursion is actually useful and often allows short, elegant, easy-to-understand solutions; and
- acquire the confidence that comes from having a deep understanding of how objects and data structures really work.

*It is important to note again that the following is a summary of a three-year AP Computer Science program, to include Pre-AP Computer Science, AP Computer Science A, and AP Computer Science AB.*

## Pre-AP Computer Science

### Weeks 1–3 **Karel the Robot**

The robot world; Simple programs using classes and objects; Inheritance; Object-oriented development; Top-down development; Encapsulation & information hiding; Procedural abstraction; Debugging (categorize errors: compile-time/run-time/logic, identify and correct errors, employ techniques such as using a debugger, adding extra output statements, or hand-tracing code); Flow of control (sequential, methods); Read and understand a problem description, purpose, and goals. [C3, C4, C5, C6]

**C3**—The course teaches students to design and implement computer-based solutions to problems in a variety of application areas.

**C4**—The course teaches students to use and implement commonly used algorithms and data structures.

### Weeks 4–7 **Computer Systems**

Numerical representations and limits (representing numbers in different bases, limitations of finite representations); Hardware (primary and secondary memory, processors, peripherals); System software (language translators/compiler, virtual machines, operating systems); Types of systems (single-user systems and networks; Programming languages; Responsible use of computer systems (system reliability, privacy, legal issues and intellectual property, social and ethical ramifications of computer use). [C8, C9]

**C5**—The course teaches students to develop and select appropriate algorithms and data structures to solve problems.

### Weeks 8–12 **Objects & Primitive Data**

Primitive types vs. objects; Simple data types (**int, boolean, double, char**); Declarations (variable and constant); Console output (**System.out.print/println**); Using Classes to create objects; Java library classes (**String, Integer, Double, Math, Applet, Graphics**).

**C6**—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 Appendices 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.)

### Week 13–14 **Karel the Robot—Conditional Statements**

Using **if** and **if-else** statements with robots; Flow of control (conditional).

**C8**—The course teaches students to identify the major hardware and software components of a computer system, their relationship to one another, and the roles of these components within the system.

### Week 15–16 **Program Statements—Conditional Statements**

Boolean expressions; Using conditional expressions; Using **if, if-else**, and nested **if** statements. [C4, C5]

### Week 17 **Semester Review**

### Week 18 **Semester Finals**

### Weeks 19–20 **Karel the Robot—Instructions that Repeat**

Using **while** and **for** statements with robots; Flow of control (iteration).

**C9**—The course teaches students to recognize the ethical and social implications of computer use.

### Weeks 21–25 **Program Statements—Iterative Statements**

Using **while** and **for** statements; Analysis of algorithms (informal comparisons of running times, exact calculation of statement execution counts). [C4, C5]

**Weeks 26–31 Writing Classes**

Anatomy of classes and methods; Declarations (class, interface, method, and parameter); Method overloading; Method decomposition; Object relationships, Reasoning about programs (pre- and post-conditions, assertions); Apply data abstraction and encapsulation; Design and implement a class. [C4, C5, C6]

**C4**—The course teaches students to use and implement commonly used algorithms and data structures.

**Weeks 32–33 Boolean Logic**

Boolean expressions and laws; Truth tables.

**C5**—The course teaches students to develop and select appropriate algorithms and data structures to solve problems.

**Weeks 34–35 Semester Review**

**Week 36 Semester Finals**

**AP Computer Science A**

**Weeks 1–3** Revisit components of a computing system and their relationships; Responsible use of computer systems (system reliability, privacy, legal issues, intellectual property, and social and ethical ramifications of computer use). [C8, C9]

**Review**

Objects and primitive data; Program statements.

**Week 4 GridWorld Case Study—Experimenting**

Introduction to the case study. [C7]

**Weeks 5–6 Review—Writing Classes**

Implementing methods in classes.

**Weeks 7–9 Enhancing Classes**

References, exceptions, and class design; Understand error handling (understand runtime exceptions, throwing runtime exceptions); Java library classes (**Comparable**, **List**, **Iterator**, and **ListIterator** interfaces); Identify reusable components from existing code using classes and class libraries. [C4, C5, C6]

**C6**—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 Appendices 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.)

**Weeks 10–11 GridWorld Case Study—Tour of the Implementation**

Classes and objects used in the case study. [C7]

**Weeks 12–16 Arrays / ArrayLists / Searching & Sorting**

One- and two-dimensional arrays, **Lists** and **ArrayLists** (traversals, insertions, deletions, iterators); Searching (sequential, binary); Sorting (selection, insertion, mergesort), Testing (test classes and libraries in isolation, identifying boundary cases and generating appropriate test data, performing integration testing); Choosing appropriate data representation and algorithms. [C4, C5, C6]

**C8**—The course teaches students to identify the major hardware and software components of a computer system, their relationship to one another, and the roles of these components within the system.

**Week 17 Semester Review**

**Week 18 Semester Finals**

**Weeks 19–21 GridWorld Case Study—Part 3**

Understand and modify existing code. [C7]

**C9**—The course teaches students to recognize the ethical and social implications of computer use.

**C7**—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®.

<b>Weeks 22–24</b>	<b>Inheritance</b> Subclasses, overriding, hierarchies, using class members, polymorphism, inheritance, class hierarchy design; Java library classes ( <b>Object</b> ); Read and understand class specifications and relationships among the classes (“is-a”, “has-a”); Understand and implement a given class hierarchy; Extend a given class using inheritance; Apply functional decomposition. [C3, C4, C5, C6]
<b>Weeks 25–27</b>	<b>GridWorld Case Study—Part 4</b> Inheritance in the case study; Extend existing code using inheritance. [C7]
<b>Weeks 28–30</b>	<b>Recursion</b> Recursive thinking, programming, and sorting; Flow of control (recursion). [C4, C5]
<b>Weeks 31–34</b>	<b>AP Exam Review</b>
<b>Week 35</b>	<b>AP Exam; Post-AP Project</b>
<b>Week 36</b>	<b>Post-AP Project (cont.); Semester Finals</b>

**C3**—The course teaches students to design and implement computer-based solutions to problems in a variety of application areas.

**C4**—The course teaches students to use and implement commonly used algorithms and data structures.

**C5**—The course teaches students to develop and select appropriate algorithms and data structures to solve problems.

**C6**—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 Appendices 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.)

**C7**—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®.

**C2**—The course includes all of the topics listed in the “Computer Science AB” column of the Topic Outline in the *AP Computer Science Course Description*.

**Course Planner for AP Computer Science AB [c2]**

The following is a breakdown of the week-by-week timeline for this course. It includes reading, exercises, labs, and tests. Titles and names in parentheses refer to required texts (above) or resources listed later in this syllabus.

**Weeks 1–2**

Revisit components of a computing system and their relationships; Responsible use of computer systems (system reliability, privacy, legal issues, intellectual property, and social and ethical ramifications of computer use). Assignment: Bring in current articles on social implications of computer use from online and print media for class discussion. [C8, C9]

**Review—Writing Classes**

Writing methods in classes; Using objects; Iteration.

Lab: Environment Plotter (Brady 2003)

**Weeks 3–4**

**Review—Arrays/ArrayLists/Iterators**

One- and two-dimensional arrays; **Lists** and **ArrayLists** (traversals, insertions, deletions, iterators).

- Lab: WordList with and without iterators—AP Computer Science A Exam 2004, free-response question 1
- Lab: Iterator Lab (Brady 2003) Students implement the iterators for several algorithms that step through (traverse) a two-dimensional data structure made up of rows and columns. These algorithms are useful for many different kinds of two-dimensional data structures. [C4]

**Week 5**

**Classes, Hierarchies, Abstract Classes, and Interfaces**

Inheritance; Polymorphism using classes and interfaces; Specify the purpose and goals for a problem; Decompose a problem into classes; Define relationships and responsibilities of those classes; Design and implement a set of interacting classes; Design an interface.

- Reading: *Java Methods*, Chapter 11
- Exercises: *Java Methods*, exercises 11.1–11.5, 11.10
- Lab: Welcome (*Java Methods*, exercise 11.8)
- Tests: take-home free-response and in-class multiple-choice [C3, C4, C5, C6]

**C8**—The course teaches students to identify the major hardware and software components of a computer system, their relationship to one another, and the roles of these components within the system.

**C9**—The course teaches students to recognize the ethical and social implications of computer use.

**C4**—The course teaches students to use and implement commonly used algorithms and data structures.

**C3**—The course teaches students to design and implement computer-based solutions to problems in a variety of application areas.

**C5**—The course teaches students to develop and select appropriate algorithms and data structures to solve problems.

**C6**—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 Appendices 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.)

Week 6

**GridWorld Case Study—Overview**

Review of the classes and objects used in the case study.

- Reading: Case Study Narrative, Parts 1 and 2
- Lab: DropGame—AP Computer Science A Exam 2006, free-response question 4
- Test: in-class fill-in-the-blank and multiple-choice [C7]

**C7**—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®.

Weeks 8–9

**“Big-O” Analysis of Algorithms**

Worst-case and average-case time and space analysis of algorithms using Big-O notation; Review of sorts (selection, insertion, mergesort, and quicksort).

- Reading: *Java Methods*, Chapter 18
- Exercises: *Java Methods*, exercises 18.1–18.4ab, 18.7–18.10
- Tests: take-home free-response and in-class multiple-choice [C5]

**C5**—The course teaches students to develop and select appropriate algorithms and data structures to solve problems.

Weeks 10–12

**Java Collections Framework (JCF)**

Java Collections Framework interfaces and classes (**Collection**, **List**, **ArrayList**, **LinkedList**, **Set**, **TreeSet**, **HashSet**, **Map**, **TreeMap**, **HashMap**, **Stack**, **Queue**, **PriorityQueue**, **Iterator**, **ListIterator**); Choose appropriate advanced data structures and algorithms.

- Reading: *Java Methods*, Chapter 19
- Exercises: *Java Methods*, exercises 19.1–19.4, 19.7–19.9, 19.13, 19.18, 19.21
- Lab: Email Addresses—AP Computer Science AB Exam 2005, free-response question 4
- Lab: Thesaurus—AP Computer Science AB Exam 2006, free-response question 1
- Tests: take-home free-response and in-class multiple-choice [C4, C5, C6]

**C4**—The course teaches students to use and implement commonly used algorithms and data structures.

Weeks 13–14

**Lists and Iterators**

Implementation (using **ListNode**) and use of linked lists (singly, doubly, and circular); Traversals, insertions, deletions, iterators.

- Reading: *Java Methods*, Chapter 20
- Exercises: *Java Methods*, exercises 20.1–20.10
- Lab: Teletext (*Java Methods*, section 20.7)
- Tests: take-home free-response and in-class multiple-choice [C6]

**C6**—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 Appendices 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.)

Weeks 15–16

## Stacks and Queues

Implementation and use of stacks and queues (traversals, insertions, deletions, iterators); Review of **JCF Stack** and **Queue**.

- Reading: *Java Methods*, Chapter 21
- Exercises: *Java Methods*, exercises 21.1–21.3, 21.5abc, 21.6–21.8
- Lab: Browsing (*Java Methods*, section 21.3)
- Tests: take-home free-response and in-class multiple-choice [C4, C5]

**C4**—The course teaches students to use and implement commonly used algorithms and data structures.

**C5**—The course teaches students to develop and select appropriate algorithms and data structures to solve problems.

Week 17

## Semester Review

Week 18

## Semester Finals

Weeks 19–20

## Recursion

Understanding and implementing structural and procedural recursion.

- Reading: *Java Methods*, Chapter 22
- Exercises: *Java Methods*, exercises 22.1–4, 22.6–22.9, 22.11–22.12, 22.14–22.15
- Lab: The Game of Hex (*Java Methods*, section 22.6)
- Tests: take-home free-response and in-class multiple-choice [C4, C5]

Weeks 21–22

## Binary Search Trees

- Implementation (using **TreeNode**) and use of Binary Search Trees (traversals, insertions, deletions, iterators); Review of **JCF Set**, **Map**, **TreeSet**, **TreeMap**.
- Reading: *Java Methods*, Chapter 23
- Exercises: *Java Methods*, exercises 23.1–23.3, 23.6–23.10, 23.12, 23.15abcdef, 23.16–23.19, 23.21
- Lab: Morse Code (*Java Methods*, section 23.6)
- Tests: take-home free-response and in-class multiple-choice [C4]

Weeks 23–24

### Look-up Tables and Hashing

Implementation and use of look-up tables and hash tables (traversals, insertions, deletions, iterators); **hashCode** method; Review of JCF **Set**, **Map**, **HashSet**, **HashMap**. [C4]

- Reading: *Java Methods*, Chapter 24
- Exercises: *Java Methods*, exercises 24.1–24.3, 24.5–24.7a, 24.9
- Lab: Search Engine (*Java Methods*, section 24.6)
- Tests: take-home free-response and in-class multiple-choice

Weeks 25–26

### Heaps and Priority Queues

Implementation and use of heaps and priority queues (traversals, insertions, deletions, iterators); Review of JCF **Queue** and **PriorityQueue**. [C4]

- Reading: *Java Methods*, Chapter 25
- Exercises: *Java Methods*, exercises 25.1–25.3ab, 25.4–25.5, 25.7ab
- Lab: Heapsort (*Java Methods*, section 25.5)
- Tests: take-home free-response and in-class multiple-choice

Week 27

### Big-O Revisited (*Java Methods*, Chapter 18)

Review and analysis of sorts.

Lab: SortDetective (Levine 2002) [C4, C5, C6]

Weeks 28–30

### GridWorld Case Study

- Running the GridWorld Project
- Learn how to use the GridWorld GUI
- Review of the GridWorld classes, interfaces, and objects and their interactions
- Experimenting to determine Actor Behavior
- Analyzing and extending the Bug Class
- Use and understand the Critter class within a GridWorld, understand the behaviors of the Critter class when invoking methods, evaluate the effect of changes made when developing subclasses of the Critter class [C7]

Weeks 31–33

### AP Exam Review

Review of AB course material.

- Reading and Exercises: *Barron's*

**C4**—The course teaches students to use and implement commonly used algorithms and data structures.

**C5**—The course teaches students to develop and select appropriate algorithms and data structures to solve problems.

**C6**—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 Appendices 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.)

**C7**—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®.

<b>Week 34</b>	<b>AP Exam</b>
	<b>After the Exam Projects</b>
	Various projects, depending on the interest and ability of the students. The content is not critical to the course.
<b>Week 35</b>	<b>Continue Work on Projects</b>
<b>Week 36</b>	<b>Semester Finals</b>

## Teaching Strategies

Class time is fully utilized for discussion, labs, and multiple-choice tests, so outside classwork (reading and homework) is critical to students' understanding of the material. I reserve the last few weeks prior to the AP Exam for review. Students work on multiple-choice and free-response questions at home and in class. I rely heavily on the practice questions in Teukolsky's AP Exam preparation book (see Texts, above).

## Lab Component

Writing computer programs is critical to understanding the material. I usually assign one lab per chapter. Labs involve the definition of classes with all their constants, variables, and methods as well as interactions among those classes, their implementation, and their use.

These assignments are typically done on an individual basis; I encourage collaboration but do not allow copying. There is ample class time for most students to complete the lab assignments. I also provide lab time and tutoring before and after school. My classroom contains desks and computer tables with desktop machines. These computers have the *Sun Java SDK* and the *JCreator* Interactive Development Environment, tailored for our use. All of the Java-specific software we use in the classroom is available at no cost. I make CDs of this software available for students to install on their home computers.

## Teacher Resources

Brady, Alyce. AP Computer Science Teaching Resources. Kalamazoo College, 2003. <http://max.cs.kzoo.edu/AP/>

The College Board. *AP GridWorld Case Study*. New York: College Entrance Examination Board, 2006.

The College Board. *Quick Reference: AP Computer Science A*. New York: College Entrance Examination Board, 2007.

[http://apcentral.collegeboard.com/apc/public/repository/compsci\\_a\\_exam\\_appendix.pdf](http://apcentral.collegeboard.com/apc/public/repository/compsci_a_exam_appendix.pdf)

Litvin, Maria, and Gary Litvin. *Java Methods A & AB: Object Oriented Programming and Data Structures*. Andover, Mass.: Skylight Publishing, 2006.

Litvin, Maria, and Gary Litvin. *Java Methods AB: Data Structures*. Andover, Mass.: Skylight Publishing, 2003.

Teukolsky, Rosalyn. *Barron's AP Computer Science Levels A and AB 2007*, 3rd ed. Hauppauge, N.Y.: Barron's Educational Series, 2006.