AP® Computer Science AB
Syllabus 3

Overview

This course provides students with an opportunity to further develop and refine their programming skills. In particular, the emphasis of this course is on the organization of information; the implementation of common data structures such as lists, stacks, queues, trees, and graphs; and techniques of data abstraction, including encapsulation and inheritance. Students also explore recursion and the close relationship between data structures and algorithms. Hands-on programming is a central component of this course. The course includes numerous "minilabs" during normal class time. Outside programming assignments continue the focus on the design, implementation, and testing of object-oriented programs.

At the conclusion of this course, students will understand common data structures and algorithms and be able to apply that understanding to implementing new data abstractions and using existing library components. Students will also be stronger programmers and feel comfortable programming in Java

Textbooks and Resources

- Collins, William. *Data Structures and the Java Collections Framework*, McGraw-Hill, 2002.
- Vermeulen, et al. *The Elements of Java Style*, Cambridge University Press, 2000.
- The AP® Computer Science GridWorld Case Study
- AP Computer Science Quick Reference Guide

Note on Collins's book: The reading assignments for this course focus on the use of the standard classes from a client code perspective; the implementation of simpler classes, especially for linked lists and trees; and the data structures and algorithms concepts that span implementations.

Assignments

Students are expected to do three types of assignments outside of class time. The first are *reading assignments* from the textbook and the GridWorld Case Study. For *discussion questions*, students are expected to come to class prepared to present their solutions or participate in discussions based on the discussion questions. *Programming projects* are usually complete programs that students are expected to design, implement, and document on their own or with a partner. Students are often provided "black-box code," such as classes that implement a graphical user interface that they can use without understanding how they are implemented. In this course there is a close association between the lab exercises students work on in class and the programming projects they work on outside of class.

Course Planner [C2]

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*.

Fall Semester:

The fall semester begins with an overview of important hardware and software components that are available in our labs, including primary and secondary memory peripherals, networks, and system versus application software. If they haven't already, students will become extremely familiar with the labs' network and operating systems throughout the course. With an emphasis on *discussion questions,* students will also understand and be able to clearly discuss social and ethical ramifications of computer use through daily class discussions. Discussion topics include software piracy, copyright and fair use, and the gender gap in the computing field. Students are also divided into groups to present and discuss a current news article on ethical issues in computing. **[C8, C9]**

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.

Unit 1 [C4, C5, C6, C7]

- What Are Data Structures? What Are Algorithms?
- Review of Java and OOP
    1. Language Features
    2. Classes, Encapsulation, Abstract Data Types
        a. Decompose into classes
    3. Inheritance, Polymorphism, and Dynamic Binding
    4. Interfaces
    5. JavaDocs, Specifications, Pre/Post Conditions
- Arrays and ArrayLists Review of Linear Traversal Patterns
- Chapters 1–4 of the GridWorld Case Study

Reading Assignments: from Chapters 1 and 2 of Collins

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.

Unit 2

- Collection Classes, List Interface, Set Interface
    1. Interacting Classes, advanced data structure, and algorithms
- ArrayList
- Iterators
- Sequential and Binary Search

Reading Assignments: from Chapters 1, 2, 3, 5, and 13 of Collins and Chapter 5 of the GridWorld Case Study

Programming Project: VLBoundedGrid

Lab and Programming Project: Iterator Lab

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®.

Unit 3 [C4, C5, C6, C7]

- Linear Lists (Client View)
- Linked Lists, Doubly Linked Lists (Implementation)

Reading Assignments: from Chapters 2 and 6 of Collins

Labs: Linked List

- Array of movie actors and an iterator to traverse the array list

Unit 4

- More on Linked Lists, Circular Lists
- Introduction to Queues and Stacks

Reading Assignments: from Chapters 6 and 7 of Collins

Lab and Programming Project: Linked List Labs

Unit 5

- Queues and Stacks (continued)
    1. Heaps, Sets
- Two-Dimensional Arrays

Reading Assignments: from Chapters 4, 7, and 12 of Collins

Labs: SMBoundedGrid and Queues Lab

Spring Semester

The Spring Semester can begin with revisiting topics on Program Analysis before continuing with the Collins textbook for Units 6–10:

- Number Representation—Round-off Error
- Pre/Post Conditions
- Debugging
- Runtime Exceptions (Throw)/Error Handling
- Big-Oh Notation
- Analysis: Worst/Average Case

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®.

Unit 6 [C4, C5, C6]

- Declarations
    1. Constant, Variable, Parameter
- Methods
    1. Conditional, Sequential, Iterations
- Recursion Sorting Algorithms

Reading Assignments: from Chapters 6 and 7 of Collins

Programming Project: Car Wash Simulation

Lab: N Queens Lab

Units 7 and 8

- Sorts and Search
    1. Mergesort, Quicksort, Heapsort, Insertion, Selection
- Trees
- Tree Traversal Algorithms
- Binary Search Trees

Reading Assignments: from Chapters 8, 10, and 13 of Collins

Labs: Binary Tree Lab, Binary Search Tree Lab, and Comparison of Sorts Lab

- Linear and binary search

Units 9 and 10

- Maps/Dictionaries
- Hash Tables, Hash Functions, HashMap, Chaining Sets, HashSet, Open Address Hashing
- HashIterator
- Priority Queues
- Selection of algorithms and data structures based on time and space requirements

Reading Assignments: from Chapter 13 of Collins

Lab and Programming Project: HMUnboundedGrid

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.)

Description of Programming Assignments

The hands-on experience provided by in-class lab exercises and follow-up programming projects is an important component of the course. A number of the assignments come from the AP Computer Science GridWorld Case Study. Several others apply the GridWorld interface to other contexts, such as using a BoundedGrid object to represent the N x N "chessboard" in the N Queens problem. Some focus on implementing common data structures, and others focus on using them to build client applications.

- **GridWorld Case Study:** Students write several alternative implementations for the Grid interface using lists, a sparse matrix, a binary search tree, and a hash map.
- **Iterator Lab:** Students implement iterators for a number of different traversals and partial-traversals of a two-dimensional BoundedEnv data structure.
  (http://max.cs.kzoo.edu/AP/MBS/RelatedAssignments/Iterators/IteratorLab.shtml)
- **Linked List Lab Series:** This series gives students a chance to write a program that uses the standard Java LinkedList class and then to write two simpler classes that implement a linked list and a doubly linked list.
- **Queues Lab** and **Car Wash Simulation Project:** In the Queues Lab, students write a class that implements the Queue interface defined for the AP Program using a linked list. They then use that queue in the Car Wash Simulation Project. This project also provides a case study of the object-oriented design process.
- **N Queens Lab:** Students implement a solution to the classic N Queens problem, using a BoundedGrid as the N x N board.
- **Binary Tree Lab** and **BST Lab:** Students implement a number of recursive, depth-first traversal algorithms for a binary tree and a binary search tree. A breadth-first insertion method is provided for the binary tree, but students implement insertion and deletion algorithms for the binary search tree. Students also write several classes that implement a NodeVisitor interface written for this lab. The combination of a traversal algorithm with a visitor interface provides a general mechanism for handling many tasks that require tree traversal.

Comparison of Sorts Lab:

Students implement various sorts on different data sets and compare their efficiency. Students graph their findings to contrast between n^2 sorts and nlogn sorting algorithms. **[C3, C6, C7]**

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

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®.