

AP[®] Computer Science A

Syllabus [C2]

C2 - 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*.

Course Overview

Advanced Placement Computer Science is a rigorous, college-level course designed to prepare students for the AP[®] Computer Science A examination. This course uses the Java programming language and emphasizes object-oriented programming, problem solving, and algorithm development using Java. It also includes an introductory study of data structures, design and abstraction. Many topics in this course are mathematically intensive and require a firm understanding of algebraic concepts.

The course meets daily for the entire school year. It includes both classroom and computer lab instruction. Students will need to complete significant assignments and preparations in addition to class time.

All students are expected to prepare for and take the College Board's Advanced Placement Computer Science A exam.

Course Prerequisites

The course is open to all students who have passed both Algebra II and Programming - Design and Logic. Honors Java Programming is also strongly recommended.

Textbooks

Fundamentals of Java: AP Computer Science, Second Edition*

Lambert and Osborne, 2003, Course Technology (Referred to as Lambert Text)

AP Computer Science, Levels A & AB, 4th Edition

Barron's Educational Series, 2007 (Referred to as Barrons Text)

Supplemental Resources

AP[®] GridWorld Case Study (Referred to as Case Study)

Fundamentals of Java: AP* Computer Science, Third Edition

Lambert and Osborne, 2006, Course Technology (Referred to as *Third Edition*)

Fundamentals of Java: AP Computer Science, Second Edition Instructor Resources*

Lambert and Osborne, 2003, Course Technology (Referred to as IR). Includes test bank and instructor manual (Referred to as IM)

Fundamentals of Java: AP Computer Science Review Companion Workbook, Third Edition*

Adelman & Nagin, 2006 Course Technology (Referred to as Workbook)

Java Programming, Comprehensive Concepts and Techniques

Shelly, Cashman, Starks and Mick, 2004, Course Technology (Referred to as Shelly Cashman)

Introduction to Computer Science using Java <http://chortle.ccsu.edu/java5/cs151java.html> , Bradley Kjell, 2006. Central Connecticut State University (Referred to as Kjell)

Enhance looping in Java 5.0 with for/in (30 Nov 2004) from IBM Developer Works Web (Referred to as Enhanced For Loop) site: <http://www-128.ibm.com/developerworks/java/library/j-forin.html>

OO Design Using Pong Learning Module (2004). Computer Science Teachers Association - Recommended Resources. from csta|Computer Science Teachers Association Web

site: <http://csta.acm.org/Resources/sub/RecommendedResources.html> (Referred to as Pong)

Conquer Medieval Kingdoms with CodeRuler (29 Jun 2004) from IBM Developer Works Web

site:<http://www-128.ibm.com/developerworks/java/library/j-coderuler/> (Referred to as CodeRuler)

Lab Setup/Lab Usage

AP Computer Science A is taught in a computer lab. Every student has their own computer to independently use during class time. A projector is mounted in the lab to enable computer images to be displayed to the entire class. Classroom management software also enables the teacher to take control of student computers to:

- Display their image on all student computers
- Display the image from a student's computer to all student computers
- Blank student monitors to focus student attention
- The computer lab also includes access to a network printer, a scanner, and digital camera.

Students typically use the computers daily to complete assignments. On average, students spend 3 hours of class time per week independently using the computer. Students also have access to computers before and after school in their computer science lab and in the media center. In addition, students will normally need to spend additional time on their computers at home developing their projects.

Network

All computers are networked. Each student has a unique login and password. Students primarily save their assignments and projects in their individual, secured folder called the H drive. Students can also submit completed projects to be graded to a special Inbox folder on the network.

Software

In school, students use workstations with a Windows operating system. Outside of school, students may use other operating systems. The following software is installed on all Computer Science lab workstations. Each version is currently available to all students via the Internet. As needed, students are provided with the software on CD or other means so they can complete assignments outside of class.

[Java 2 SE version 5.0](#)

Sun® Microsystems

[Eclipse SDK 3.2.1.](#)

Eclipse.org

[OO Design Using Pong Learning Module](#)

Computer Science Teachers of America

[CodeRuler](#)

IBM®

Strategies

This course uses a variety of teaching and assessment strategies. Students are instructed through lectures, labs, homework, readings, recitation and discussion. Students are assessed through project evaluations, written exams, quizzes and homework. Students learn to work with relatively large programs such as the GridWorld case study. The case study is used as a resource throughout the year both to teach its content and to develop student skills in working with larger projects.

Concepts are frequently presented to students using teacher-guided instruction. Concepts and examples are presented using sample code displayed on the LCD and student monitors. Material distributed through our classroom management software can be viewed by students on their computers, but cannot be manipulated. This technique focuses attention on the concepts being taught or reviewed without allowing students to “experiment”. Students can also retrieve files from a sharable location on the network. Students can manipulate and experiment with code during teacher-guided instruction. Material presented originates from a variety of sources such as teacher-created, textbook, and Internet resources.

Concepts are also introduced and reviewed using our textbooks and study guide. Homework includes reading assignments, completion of review questions and programming assignments.

Hands-on programming assignments are assigned to solidify each concept. Assignments range from quick five to ten minute code modifications to multiple-day projects encompassing skills from several units. Students who complete assignments early are encouraged to try enrichment activities which broaden their understanding of computer science concepts.

Students are given feedback and assistance during class time as they develop code. Student progress is constantly monitored to determine if an individual, group, or entire class needs additional instruction. If several students are observed making the same mistake, additional instruction can occur using the classroom management software. The code from one student can be displayed on selected or all student monitors. Discussion can then occur as to why the code is in error. Students are always encouraged to ask questions and receive prompt feedback on assignments. Rubrics are distributed with each assignment and comments are

included with graded feedback as appropriate. Students work on assignments individually and collaboratively as the particular assignment dictates.

The course uses a variety of resources in addition to its principle texts. Significant use is made of the third edition of the Lambert text to address such recent topics as generics, the "For – Each" loop and the Scanner class (each of these are new to the version of Java used in the course). Additionally, the third edition is used to address the merge sort algorithm and algorithm complexity analysis. These uses are indicated in the topic outline.

The Lambert Text provides a case study in each of its lessons. Case studies are used in each lesson to illustrate topics taught. The case studies provide students with experience in analyzing problems and developing larger solutions. Instructors can provide students with source code for functioning versions of each of the case studies. Students learn by manipulating the objects and algorithms contained in the lesson in addition to formal instruction and written work.

Additional resources provide for a variety of instructional techniques and learning styles. The Kjell resource is used frequently to provide visual presentation of instruction and to support programmed instruction. It also provides a source of instruction for students who are ill or otherwise out of class. The Pong resource allows for early use of objects and classes. Students get an informal introduction to these topics while developing the game. The GridWorld case study is both a topic and a resource. It is used instructionally to teach program development using an object-oriented approach. Use of these resources and others are also indicated in the topic outline.

Syllabus at a Glance:

Unit	Description	Weeks	Assessments	Lambert Projects
1	Getting Started with Java Introduction to Java, first programs and control structures	1 - 7	Exams 1 – 4 Unit Projects (1 – 4)	1(1) 2(1, 3, 4 & 5) 3(1 – 4) 4(1 – 8)
2	The Next Step With Java Introduction to Objects Oriented Programming, Boolean expressions and nested control structures	8 - 14	Exams 5 – 7 GridWorld (ex 1 and 2) Unit Projects (1 – 4)	5(1 – 6) 6(1, 3, 5 and 7) 7(3, 5 and 6)
3	Arrays, Recursion and Complexity Linear and Binary searches Selection, Insertion and Merge sorts	15- 23	Exams 8 – 12 GridWorld (Group Activity I) Unit Projects (1 – 5)	8(1, 3, 6, 8 and 9) 9(1, 2, 3 and 4) 10(1, 2, 8 and 9) 11(2, 3, 4, 6 and 7)
4	Exam Preparations GridWorld CaseStudy	24 - 29	GridWorld (Group Activity II) Barron's tests: 1 – 7 , 12 and 13	Sample Exam Problems from 2005, 2006 and 2007
5	Introduction to Larger Projects – Code Ruler[C7]	31 - 34	Move, Capture and Strategy	

Topic Outline:

Objective: This lesson discusses computing in general, hardware and software, the representation of information in binary and general concepts of object-oriented programming.

Unit	Topics	Assessments	Resources
Unit 1 - Lesson 1 (week 1)	Background [C8] [C9] <ol style="list-style-type: none"> History of computers (p 3) How hardware and software make up computer architecture (p 5) Understanding the binary representation of data and programs in computers (p 9) The evolution of programming languages (p 15) Fundamental concepts of object-oriented programming (p 19) Computer Ethics: Copyright, Intellectual Property and Digital Information (p 22) 	Project 1-1 (p. 25) Test 1 IR - Lesson 1 test bank IR - Lesson 1 IM	Lambert Text pages 3 - 25 (Unit 1 Lesson 1) <u>Supplemental Teaching Strategies:</u> Use KJell (chapters 1 – 4) with topic a.

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 Appendixes A and B of the *AP Computer Science Course Description*.

Objective: Students will learn how to write their first simple Java programs, compile and run them in the Eclipse 3.2 environment. Basic concepts of variables, input and output statements, and sending messages to objects will be introduced.

Unit	Topic	Assessments	Resources
Unit 1 Lesson 2 (week 2)	First Java Programs [C3] [C6][C9] <ol style="list-style-type: none"> Why is Java an important programming language? (p 27) The Java Virtual Machine (JVM) and byte code (p 28) User interface style (p 29) Structure of a simple Java program (p 30) Write a simple Java Program (HelloWorld.java) (p 30) Edit, compile, and run a Java program (p 32) Understand compile-time errors (p 35) Code readability (p 36) Computer Ethics: Intrusive Hacking (p 37) 	Project 2-1 (p. 48) Project 2-3 (p. 49) Project 2-4 (p. 49) Project 2-5 (p. 49) Test 2 IR Lesson 2 test bank IR Lesson 2 IM	Lambert Text pages 27 - 49 (Unit 1 Lesson 2) <u>Supplemental Teaching Strategies:</u> Use Kjell (chapter 5) with topic b.

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

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.

Objective: Students learn basic elements of the Java language and how to detect and correct syntax and logic errors.

Unit	Topic	Assessments	Resources
Unit 1 Lesson 3 (weeks 3 - 4)	Syntax, Errors and Debugging [C3] [C4] [C5] [C6] [C9] a. Language elements (p 51) b. Basic syntax and semantics (p 53) c. Data types (p 53) i. Primitive ii. Objects d. Construct and use numeric and String literals (p 55) e. Name and use variables and constants (p 56) f. Terminal Input and Output g. Understand the precedence of arithmetic operators (p 59) h. Casting to int and double (p 62) i. Concatenate two Strings or a number and a String (p 63) j. Use of comments in a program (p 70) k. The difference between syntax, run-time and logic errors (p 75) l. Debugging and Testing program (p 80) m. Computer Ethics: Computer Viruses (p 74) n. Case study	Case Study 1: Income Tax Calculator (p 72) Case Study 2: Count the Angels (p 82) Project 3-1 (p. 88) Project 3-2 (p. 88) Project 3-3 (p. 88) Project 3-4 (p. 88) Test 3 IR Lesson 3 test bank IR Lesson 3 IM	Lambert Text pages 51 - 88 (Unit 1 Lesson 3) <u>Supplemental Teaching Strategies:</u> Use Third Edition section 3.3 with topic f (Scanner class). Use Kjell chapters 8 and 9 with topic c. Use Shelly Cashman pages 1.19 – 1.21 with topic l.

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 Appendixes A and B of the *AP Computer Science Course Description*.

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

Objective: All programs to this point have consisted of short, sequential instructions. This lesson introduces control statements that allow repetition and selecting within the Java programming language.

Unit	Topic	Assessments	Resources
Unit 1 Lesson 4 (weeks 5 - 7)	<p>Introduction to Control Structures [C3] [C4] [C5] [C6]</p> <p>a. Extended assignment operators (p 91)</p> <p>b. Increment and decrement operators (p 92)</p> <p>c. The Math class and methods (p 93)</p> <p>d. If and if else statements (p 97)</p> <p>e. The While statement (p 101)</p> <p>f. The for statement (p 105) Note: Enhanced For loop topic covered in unit on arrays</p> <p>g. Counter-controlled input (p 108)</p> <p>h. Nested control statements and the break statement (p 110)</p> <p>i. Sentinel-controlled input (p 111)</p> <p>j. Errors in loops and loop verification (p 116)</p> <p>k. Case study</p>	<p>Case Study: The Folly of Gambling (p 112)</p> <p>Project 4-1 (p. 123)</p> <p>Project 4-2 (p. 123)</p> <p>Project 4-3 (p. 123)</p> <p>Project 4-4 (p. 123)</p> <p>Project 4-5 (p. 123)</p> <p>Project 4-6 (p. 124)</p> <p>Project 4-8 (p. 124)</p> <p>IR Lesson 4 test bank</p> <p>IR Lesson 4 IM</p> <p>Test 4</p> <p>Unit Projects 1 – 4</p>	<p>Lambert Text pages 91 - 124 (Unit 1 Lesson 4)</p> <p><u>Supplemental Teaching Strategies:</u></p> <p>Use Kjell chapters 14 – 18 with topic e.</p> <p>Use Third Edition section 6.7 with topic f.</p>

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 Appendixes A and B of the *AP Computer Science Course Description*.

Objective: In this lesson, students will explore the internal structure of classes and objects, how to read and modify classes, and how to create classes of their own.

Unit	Topic	Assessments	Resources
Unit 2 Lesson 5 (weeks 8 - 9)	Introduction to Defining Classes [C3] [C4] [C5] [C6] [C7] <ol style="list-style-type: none"> The internal structure of Classes and Objects (p 131) Pong project Primitive types, reference types and the null value (p 137) Constructors (p 142) The structure and behavior of methods. Pre and post conditions (p 149) Scope and lifetime of variables and assertions (p 153) Case study 	Case Study: Student Test Scores (p 146) Project 5-1 (p. 167) Project 5-1 (p. 167) Project 5-4 (p. 168) Project 5-5 (p. 168) Project 5-6 (p. 168) IR Lesson 5 test bank IR Lesson 5 IM Test 5	Lambert Text pages 131 – 168 (Unit 2 Lesson 5) <u>Supplemental Teaching Strategies:</u> Use Pong with topics a and b. Use Kjell chapters 25 – 33 with topics a – e.

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 Appendixes A and B of the *AP Computer Science Course Description*.

Objective: This lesson introduces the GridWorld case study

Unit	Topic	Assessments	Resources
Unit 2 Lesson 5.1 (week 10)	Introduction to <i>GridWorld</i> case study [C7] <ol style="list-style-type: none"> Chapter 1 (part 1) of the <i>GridWorld</i> case study 	“Did You Know”: 1 and 2 Exercises set 1	<u>Supplemental Teaching Strategies:</u> <i>GridWorld</i> case study

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.

Objective: In this lesson, students will explore more advanced aspects of control statements, such as logical operators, nested if statements, and nested loops. Logic and syntax debugging techniques are also examined.

Unit	Topic	Assessments	Resources
Unit 2 Lesson 6 (weeks 11 - 12)	Control Statements Continued [C3] [C4] [C5] [C6] a. Logical Operators (p 169) b. Testing if statements (p 182) c. Nested if statements (p 185) d. Logical errors in nested ifs (p 188) e. Nested loops (p 192) f. Testing loops (p 196) g. Technology Careers: Artificial Intelligence, Robots and Softbots (p 184) h. Case study	Case Study: Compute Weekly Pay (p 176) Case Study: Fibonacci Numbers (p 195) Project 6-1 (p. 200) Project 6-3 (p. 201) Project 6-5 (p. 201) Project 6-7 (p. 202) IR Lesson 6 test bank IR Lesson 6 IM Test 6	Lambert Text pages 169 - 202 (Unit 2 Lesson 6) <u>Supplemental Teaching Strategies:</u> Use Kjell chapters 14 – 18 with topics a – f. Use GridWorld case study with topics b – d.

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.

Objective: This lesson explores several ways to enhance a program's interface. Standard techniques for enhancing a terminal-based interface are presented, as well as an introduction to event-driving GUI applications.

Unit	Topic	Assessments	Resources
Unit 2 Lesson 7 (weeks 13 - 14)	Improving the User Interface [C3] [C6] a. Repeating sets of inputs and type char (p 204) b. Menu-driven program (p 206) c. GUI-based program (p 208) d. Formatted output (p 218) e. GUIs and Applets (p 225 and) f. HTML and Applets g. Case study	Case Study: A Sales Table (p. 220) Project 7-3 (p. 226) Project 7-5 (p. 227) Project 7-6 (p. 227) IR Lesson 7 test bank IR Lesson 7 IM Test 7 Unit Projects 1 - 4	Lambert Text pages 203 – 227 (Unit 2 Lesson 7) <u>Supplemental Teaching Strategies:</u> Use Shelly Cashman pages 2.51 – 2.60. Use Kjell chapters 36 – 38

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 Appendixes A and B of the *AP Computer Science Course Description*.

Objective: This lesson explains the mechanics of declaring arrays and several basic algorithms for manipulating them. The enhanced for loop with Java 5.0 is also introduced.

Unit	Topic	Assessments	Resources
Unit 3 Lesson 8 (weeks 15 - 16)	Introduction to Arrays [C3] [C4] [C5] [C6] a. Conceptual overview (p 235) b. Simple array manipulations (p 237) c. Looping through arrays (p 238) d. Declaring arrays (p 240) e. Working with arrays that are not full (p 243) f. Parallel arrays (p 245) g. Two-dimensional arrays (p 247) h. Using the enhanced for loop (p 321 of supplemental text) i. Arrays and methods (p 249) j. Arrays of objects (p 252) k. For Each loops l. Case study	Case Study: Student Test Scores Again (p. 253) Project 8-1 (p. 265) Project 8-3 (p. 266) Project 8-6 (p. 266) Project 8-8 (p. 267) Project 8-9 (p. 267) IR Lesson 8 test bank IR Lesson 8 IM Test 8	Lambert Text pages 235 – 268 (Unit 3 Lesson 8) <u>Supplemental Teaching Strategies:</u> Use Third Edition pages 321-322 with topic h. Use Kjell chapters 46 – 48 with topic b – f. Use Enhanced For Loop with topic c and h.

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 Appendixes A and B of the *AP Computer Science Course Description*.

Objective: This lesson explores the real power of object-oriented programming with the capacity to reduce code and distribute responsibilities for such things as error handling in a software system.

Unit	Topic	Assessments	Resources
Unit 3 Lesson 9 (weeks 17 - 18)	Classes Continued [C3] [C4] [C5] [C6] a. Class (static) variables and methods (p 270) b. Java interfaces – the client perspective (p 272) c. Java interfaced – the implementation perspective (p 277) d. Code reuse through inheritance (p 282) e. Inheritance and abstract classes (p 286) f. Interfaces and inheritance (p 292) g. Acceptable classes for parameters and return values (p 301) h. Error handling with classes (p 304) i. Exceptions (p 305) j. Reference types, equality and object identity (p 308) k. Case study	Case Study: Compute Weekly Pay Revisited (p 295) Project 9-1 (p. 312) Project 9-2 (p. 313) Project 9-3 (p. 313) Project 9-4 (p. 314) IR Lesson 9 test bank IR Lesson 9 IM Test 9	Lambert Text pages 269 – 313 (Unit 3 Lesson 9) <u>Supplemental Teaching Strategies:</u> Use GridWorld Case Study with b – e. Use Kjell chapters 50 - 53 with topic b – d.

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 Appendixes A and B of the *AP Computer Science Course Description*.

Objective: The lesson builds on the introduction to arrays covered in Unit 8. More complex operations on arrays, such as searching, sorting, insertions, and removals are covered. Additionally, arrays of objects and the ARRAYLIST class are covered topics.

Unit	Topic	Assessments	Resources
Unit 3 Lesson 10 (weeks 19 - 20)	Arrays Continued [C3] [C4] [C5] [C6] a. Advanced operations on Strings (p 315) b. Searching (p 319) c. Sorting (p 325) d. Insertions and removals (p 330) e. Working with Arrays of Objects (p 334) f. The Class java.util.ArrayList (p 352) g. Case study	Case Study: Student Test Scores Yet Again (p 345) Project 10-1 (p. 358) Project 10-2 (p. 358) Project 10-8 (p. 359) Project 10-9 (p. 359) IR Lesson 10 test bank IR Lesson 10 IM Test 10	Lambert Text pages 315 – 359 <u>Supplemental Teaching Strategies:</u> Use Third Edition pages 269 – 297 for parts b and C.

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.

Objective: This lesson continues the GridWorld case study

Unit	Topic	Assessments	Resources
Unit 3 Lesson 10.1 (week 21)	GridWorld case study [C7] a. Chapter 2 (part 2) of the GridWorld case study	"Did You Know": 3 – 6 Group Activity I	GridWorld case study

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 Appendixes A and B of the *AP Computer Science Course Description*.

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.

Objective: This lesson continues the topic of sorting and searching by introducing recursion and complexity. A recursive algorithm is one that refers to itself by name in a manner that appears to be circular (rather than iterative). Complexity analysis is concerned with determining an algorithm's efficiency.

Unit	Topic	Assessments	Resources
Unit 3 Lesson 11 (weeks 22 - 23)	Recursion, Complexity, and Searching and Sorting [C3] [C4] [C5] [C6] a. Recursion (p 362) b. Complexity analysis (p 373) c. Binary search (p 381) d. Selection Sort e. Insertion Sort f. Quicksort (p 385) g. Mergesort h. Design, testing and debugging hints i. Technology Careers: Programming Skills: Recursion need not be expensive (p 374) j. Case study	Case Study: Comparing Sort Algorithms (p. 389) Project 11-2 Project 11-3 Project 11-4 Project 11-6 Project 11-7 Test 11 Test 12 3 rd edition IR Lesson 11 test bank IR Lesson 11 IM Unit Projects 1 - 4	Lambert Text pages 361 – 396 Use Third Edition sections 11.3 and 12. 5 with topics d, e and g. Use Kjell chapters 70 – 74 with topic a.

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 Appendixes A and B of the *AP Computer Science Course Description*.

Objective: This completes the study of the GridWorld case.

Unit	Topic	Assessments	Resources
Unit 4 (weeks 24 - 25)	a. GridWorld case study [C7] b. Chapters 3 & 4 (parts 3 & 4) of the GridWorld case study c. Enhanced For loop	"Did You Know": 7 – 9 Exercise set 3 Group Activity II	GridWorld case study

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.

Objective: AP Exam Preparation			
Unit	Topic	Assessments	Resources
Unit 4 (weeks 26 - 29)	Extensive Subject Review [C3] [C4] [C5] [C6] a. Language Features b. Objects c. Inheritance d. Standard Classes e. Program analysis f. Array and ArrayLists g. Recursion h. Standard Algorithms i. GridWorld Case Study	Barron's tests: 1 – 7 , 12 and 13 (Chapter and test numbers may be revised with new version of Barron's text.) Practice Exams: Previous Exams (Free response sections)	Barrons Text Chapters 1 – 7 , 12 and 13 Previous Exams (Free response sections) GameWorld Case Study

Objective: This lesson introduces the IBM Code Ruler Project.			
Unit	Topic	Assessments	Resources
Unit 4 (weeks 31 - 34)	Introduction to Code Ruler[C7] a. Introduction b. Tactics c. Strategies	Movement, Captures and Strategy Design and Implementation	<i>Conquer Medieval Kingdoms with CodeRuler</i>

APCS A Topic Outline Correlation [c2]

I. Object-Oriented Program Design

A. Program design

1. Read and understand a problem description, purpose, and goals.	Unit 2 Lesson 5.g
2. Apply data abstraction and encapsulation.	Unit 2 Lesson 5.a
3. Read and understand class specifications and relationships among the classes ("is-a", "has-a" relationships).	Unit 1 Lesson 1.e
4. Understand and implement a given class hierarchy.	Unit 2 Lesson 5.b
5. Identify reusable components from existing code using classes and class libraries.	Unit 3 Lesson 9.d

B. Class design

1. Design and implement a class.	Unit 2 Lesson 5.a
3. Choose appropriate data representation and algorithms.	Unit 3 Lesson 10.c
4. Apply functional decomposition.	Unit 3 Lesson 9.d
5. Extend a given class using inheritance.	Unit 3 Lesson 9.e

II. Program Implementation

A. Implementation techniques

1. Methodology	
a. Object-oriented development	Unit 1 Lesson 1.e; Unit 2 Lesson 5.c
b. Top-down development	Unit 2 Lesson 1.e
c. Encapsulation and information hiding	Unit 2 Lesson 5.a
d. Procedural abstraction	Unit 2 Lesson 5.e

APCS A Topic Outline Correlation [c2]

B. Programming constructs	
1. Primitive types vs. objects	Unit 1 Lesson 3.c
2. Declaration	
a. Constant declarations	Unit 1 Lesson 3.a and 3.b
b. Variable declarations	Unit 1 Lesson 3.e
c. Class declarations	Unit 2 Lesson 5.a
d. Interface declarations	Unit 3 Lesson 9.b
e. Method declarations	Unit 2 Lesson 5.e
f. Parameter declarations	Unit 2 Lesson 5.e
3. Console Output (System.out.print/println)	Unit 1 Lesson 2.e
4. Control	
a. Methods	Unit 2 Lesson 5.e
b. Sequential	Unit 1 Lesson 2.d
c. Conditional	Unit 1 Lesson 4.d
d. Iteration	Unit 1 Lesson 4.e
e. Recursion	Unit 3 Lesson 11.a
C. Java library classes	
java.lang.Math	Unit 1 Lesson 4.c
java.lang.String	Unit 3 Lesson 10.a
java.lang.Integer	Unit 3 Lesson 10.e
java.lang.Double	Unit 3 Lesson 10.e
java.lang.Comparable (interface)	Unit 3 Lesson 10.a
java.lang.Object	Unit 3 Lesson 10.a

APCS A Topic Outline Correlation [c2]

java.util.ArrayList	Unit 3 Lesson 10.f
III. Program Analysis	
A. Testing	
1. Test classes and libraries in isolation.	Unit 2 Lesson 6.b
2. Identify boundary cases and generate appropriate test data.	Unit 2 Lesson 6.b
3. Perform integration testing.	Unit 1 Lesson 3.l
B. Debugging	
1. Categorize errors: compile-time, run-time, logic.	Unit 1 Lesson 3.k
2. Identify and correct errors.	Unit 2 Lesson 6.b
3. Employ techniques such as using a debugger, adding extra output statements, or hand-tracing code.	Unit 1 Lesson 3.l
C. Understand and modify existing code	Unit 1 Lesson 3.n
D. Extend existing code using inheritance	Unit 3 Lesson 9.k
E. Understand error handling	
1. Understand runtime exceptions	Unit 3 Lesson 9.i
F. Reason about programs	
1. Pre- and post-conditions	Unit 2 Lesson 5.d
2. Assertions	Unit 2 Lesson 5.e
G. Analysis of algorithms	
1. Informal comparisons of running times	Unit 3 Lesson 11.b
2. Exact calculation of statement execution counts	Unit 3 Lesson 11.b
H. Numerical representations and limits	
1. Representations of numbers in different bases	Unit 1 Lesson 1.c
2. Limitations of finite representations (e.g., integer bounds, imprecision of floating-point representations, and round-off error)	Unit 1 Lesson 4.j

APCS A Topic Outline Correlation [c2]

IV. Standard Data Structures

A. Simple data types (int, boolean, double)	Unit 1 Lesson 3.c
B. Classes	Unit 1 Lesson 1.e
	Unit 2 Lesson 2.a – 2.g
C. One-dimensional arrays	Unit 3 Lesson 8.a – 8.j

V. Standard Algorithms

A. Operations on A-level data structures previously listed	
1. Traversals	Unit 3 Lesson 8.c
2. Insertions	Unit 3 Lesson 10.d
3. Deletions	Unit 3 Lesson 10.d
B. Searching	
1. Sequential	Unit 3 Lesson 10.b
2. Binary	Unit 3 Lesson 10.b
C. Sorting	
1. Selection	Unit 3 Lesson 10.c
2. Insertion	Unit 3 Lesson 10.c
3. Mergesort	Unit 3 Lesson 10.g

VI. Computing in Context

A. Major hardware components	
1. Primary and secondary memory	Unit 1 Lesson 1.a
2. Processors	Unit 1 Lesson 1.a
3. Peripherals	Unit 1 Lesson 1.a

APCS A Topic Outline Correlation [c2]

B. System software	
1. Language translators/compilers	Unit 1 Lesson 1.d
2. Virtual machines	Unit 1 Lesson 2.b
3. Operating systems	Unit 1 Lesson 1.b
C. Types of systems	
1. Single-user systems	Unit 1 Lesson 1.a
2. Networks	Unit 1 Lesson 1.a
D. Responsible use of computer systems	
1. System reliability	Unit 1 Lesson 3.m
2. Privacy	Unit 1 Lesson 2.i
3. Legal issues and intellectual property	Unit 1 Lesson 1.f
4. Social and ethical ramifications of computer use	Unit 1 Lesson 2.i
5. Extend a given class using inheritance.	Unit 3 Lesson 9.e