

AP[®] Computer Science A

Syllabus 3

Course Overview

The purpose of this class is to introduce the student to the object-oriented programming paradigm using the Java language. This course teaches students to use the standard Java library classes from the AP[®] Java subset delineated in Appendices A and B of the AP Computer Science Course Description. Concepts such as classes, objects, inheritance, polymorphism, and code reusability are studied. Hands-on laboratory work helps solidify each concept. Students complete a long-term programming project that they must demonstrate in a formal presentation.

My goal for this AP Computer Science course is to challenge students to be active learners and critical thinkers. I follow the “guide on the side, not the sage on the stage” approach, which gently introduces the students to each new concept, encouraging student-driven learning via questioning and experimentation. I believe in hands-on learning. In addition to working on programs at home, students receive instant feedback during in-class lab time once or twice a week. During lab classes, I can easily look at each student’s code, informally tracking his or her progress, and assisting accordingly. I can talk to students about their programs, and they can ask specific questions about problems they are having.

Key to notations in Programming Projects column:

CSn Problems

Examples:

| | |
|---|---|
| CS1—WordList 04A1 (ArrayList, Strings) | CS4—Ballot CD 0405 (ArrayList/Iterators) |
|---|---|

CS1: Computer Science Project 1

04A1: 2004 Computer Science A Exam, Question 1 (“WordList” problem)

CD 0405: AP Computer Science Course Description (May 2007, May 2008)

Texts

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

Deitel, H.M., and P.J. Deitel. *Java: How to Program*. 5th ed. Upper Saddle River, N.J.: Prentice Hall, 2003.

Course Planner [C2]

Fall Semester

| Week | Topic | Programming Projects (Deitel Textbook) | Tests/Large-Scale Projects | |
|------|--|--|--|---|
| 1 | <ul style="list-style-type: none"> Introduction to Computer Science The Java Language and Java Program Structure/Library Classes Hardware and Software Components | <ul style="list-style-type: none"> Intro to Class and Computer Science—Java Library Classes Hardware and software components of a computer system Language and Networks System Reliability Law and Ethics Acceptable User's Policy | <p>[C8]</p> <p>[C9]</p> | <p>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.</p> <p>C9—The course teaches students to recognize the ethical and social implications of computer use.</p> |
| 2 | <ul style="list-style-type: none"> Input/Output Arithmetic Operators If, If-Else Statements Integer and floating point types | <p>p. 80, 2.27 (Odd/Even)</p> <p>p. 80, 2.28 (Multiple)</p> <p>p. 80, 2.31 (Cast Char)</p> <p>p. 81, 2.35 (Neg/Pos/Zero)</p> | [C4] [C5] [C6] | <p>C4—The course teaches students to use and implement commonly used algorithms and data structures.</p> <p>C5—The course teaches students to develop and select appropriate algorithms and data structures to solve problems.</p> |
| 3 | <ul style="list-style-type: none"> Control Statements While Loops | <p>p. 162, 4.11 (Miles Per Gal.)</p> <p>p. 162, 4.12 (Credit Limit)</p> <p>p. 163, 4.13 (Commission)</p> <p>p. 163, 4.14 (Gross Pay)</p> | <p>Test 1</p> <p>Major Topics: Basic I/O, Arithmetic Operators</p> | <p>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 <i>AP Computer Science Course Description</i>. (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> |
| 4 | <ul style="list-style-type: none"> Control Statements for Loops | <p>p. 212, 5.6 (Smallest Int)</p> <p>p. 212, 5.7 (Product Odd)</p> <p>p. 212, 5.8 (Factorials)</p> <p>p. 212, 5.9 (Interest)</p> | | |

| Week | Topic | Programming Projects (Deitel Textbook) | Tests/Large-Scale Projects |
|------|---|---|--|
| 5 | <ul style="list-style-type: none"> Methods Arrays | <p>p. 271, 6.8 (Parking)</p> <p>p. 272, 6.14 (Power)</p> <p>p. 273, 6.23 (Celsius/Fahrenheit)</p> <p>p. 274, 6.28 (GCD)</p> <p>p. 275, 6.32 (Cai)</p> <p>p. 276, 6.36 (Recursive Power)</p> <p>p. 278, 6.45 (Recursive Error)</p> | <p>Test 2</p> <p>Major Topic: Control Statements</p> |
| 6 | <ul style="list-style-type: none"> Control Statements Declarations | <p>p. 213, 5.12 [Mail Order]</p> <p>p. 213, 5.15 [Bin, Oct, Hex, Dec]</p> <p>p. 213, 5.16 [Pi]</p> <p>p. 166, 4.31 [Encrypt]</p> <p>p. 167, 4.32 [E]</p> <p>Read Chapter 11, 11.1–11.4</p> | |
| 7 | <ul style="list-style-type: none"> Methods Pre-conditions and post-conditions | Solve the Quadratic Formula!! | |
| 8 | <ul style="list-style-type: none"> ArrayLists/Iterators Solve These Problems With ArrayList/Iterators Only | <p>p. 323, 7.12 (Array Init)</p> <p>p. 323, 7.15 (Roll Die)</p> <p>p. 324, 7.18 (Airline Res.)</p> <p>p. 330, 7.29 (Tortoise Hare)</p> <p>p. 332, 7.32 (Palindrome)</p> | <p>Test 3</p> <p>Major Topic: Methods</p> |
| 9 | <ul style="list-style-type: none"> Object Creation <ul style="list-style-type: none"> Class Specifications Class Relationships Class Hierarchy Class Design | <p>p. 396, 8.2 (Complex)</p> <p>p. 396, 8.3 (Rational)</p> <p>p. 397, 8.4 (Tick)</p> <p>p. 397, 8.5 (Date)</p> <p>p. 397, 8.6 (Date/Time)</p> | [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.)

| Week | Topic | Programming Projects (Deitel Textbook) | Tests/Large-Scale Projects | |
|------|---|--|---|---|
| 10 | <ul style="list-style-type: none"> • Object Interaction <ul style="list-style-type: none"> o Isolated Tests o Debugging | Object Case Studies CS1—WordList (ArrayList, Strings) CS2—Company (ArrayList Data Structure) (Traversals, Insertions, Deletions) | [C3] [C4] [C5] [C6] Test 4 Major Topic: Object Creation | <div style="border: 1px solid black; padding: 5px;"> <p>C3—The course teaches students to design and implement computer-based solutions to problems in a variety of application areas.</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>C4—The course teaches students to use and implement commonly used algorithms and data structures.</p> </div> |
| 11 | <ul style="list-style-type: none"> • Object Interaction | Object Case Studies CS3—Plane 02A4 (ArrayList/Iterators) CS4—Ballot CD 0405 (ArrayList/Iterators) CS5—LineEditor CD 0405 (String) | | <div style="border: 1px solid black; padding: 5px;"> <p>C5—The course teaches students to develop and select appropriate algorithms and data structures to solve problems.</p> </div> |
| 12 | <ul style="list-style-type: none"> • Object Interaction | Object Case Studies CS6—College 03A1 (ArrayList/Iterators) CS7—GroceryStore 02A2 (ArrayList/Iterators) | | <div style="border: 1px solid black; padding: 5px;"> <p>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 <i>AP Computer Science Course Description</i>. (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> </div> |
| 13 | <ul style="list-style-type: none"> • Inheritance • Polymorphism • Interfaces • Abstract Classes | Inheritance Case Studies CS8—Bank Account CD 0405 (Inheritance) | Test 5 Major Topic: Object Interaction | |
| 14 | <ul style="list-style-type: none"> • Inheritance • Polymorphism • Interfaces • Abstract Classes | Inheritance Case Studies CS9—Pet 04A02 (Inheritance/Abstract Class) | | |
| 15 | <ul style="list-style-type: none"> • Inheritance • Polymorphism • Interfaces • Abstract Classes | Inheritance Case Studies CS10—Company 03A2 with Interface (Inheritance/Interfaces) | | |
| 16 | | Final Exam | Cumulative | |

Spring Semester

| Week | Topic | Programming Projects (Deitel Textbook) | Tests/Large-Scale Projects |
|-------|---|---|----------------------------|
| 1–3 | • Project Presentations | Presentations | |
| 4 | • Strings • Exception Handling | String Project 1: Your Star Wars Name String Project 2: Pig Latin | [C3] |
| 5 | • Searching/Sorting • Recursion/Big-Oh | Searching (Sequential/Binary) Sorting (Selection, Insertion, Mergesort) | [C4] [C5] [C6] |
| 6 | • Searching/Sorting • Recursion/Big-Oh | Searching/Sorting Efficiency Project | |
| 7 | • GridWorld Case Study | GridWorld Role Play Code Walk-Through | [C7] |
| 8 | • GridWorld Case Study | GridWorld Project 1—given work with bug variations based on Part 2 of GridWorld Case Study | |
| 9 | • GridWorld Case Study | GridWorld Project 2: Design your own class based on material in Part 3 of GridWorld Case Study | |
| 10 | • GridWorld Case Study | GridWorld Project 3: Design your own critters based on material in Part 4 of GridWorld Case Study | |
| 11 | • GridWorld Case Study | GridWorld Case Study Practice and Review | |
| 12–13 | • Review for AP Exam | Review for AP Exam | |
| 14 | • AP Exam • Work on Final Projects | AP Exam Work on Final Projects | |
| 15–18 | • Spring Final Project Presentations | Spring Final Project Presentations | |

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®.

Examples of Teaching Strategies

Student Presentations

Students are required to do two 55-minute PowerPoint presentations, one in February and one in June. These presentations detail the design and implementation of a large-scale program that students have proposed, designed, and written. After each presentation, time is allowed for questions from me and other students in the class. The process allows them to practice conveying technical information in a clear and understandable manner. They learn that there is a fine line between presenting too little and too much technical detail to an audience. I have found that this alternative assessment method really gauges their understanding of the proper object-oriented programming design principles and programming skills. For example, in the first project, students must make data structure design decisions, such as which classes to create and which classes should interact with other classes. In addition, students are required to justify their use of various algorithms in completing their project (before their implementation) in the proposal phase.

Lab Component

Students work independently and with a partner when appropriate in the computer lab one to two times a week. Lab work is extremely important for two reasons. First, it allows me to see students' coding in action. I casually walk from student to student, looking at their progress on assigned programs and getting a very good idea of how each person is progressing. I can check for understanding or check coding style (indentations, naming conventions for variables, methods, classes, etc.). My instant feedback enables a student to make corrections "on the spot." Second, students can ask specific questions about concepts with which they are having trouble, and I can give personalized assistance. This is especially helpful during the first month of class, when students are new to programming, as well as during the weeks leading up to their first long-term project. In all cases, students are introduced to the AP Java subset delineated in Appendices A and B of the AP Computer Science Course Description.