

# Computer Science IV –GT

## Course Overview

This course is a fast-paced advanced level course that extends the study of the fundamental principles associated with object-oriented programming using the Java language. Topics include data structures and their implementations including an analysis of their space and time requirements. Included are two-dimensional arrays, linked lists, stacks, queues, trees, heaps, priority queues, sets, and maps. The objective of this course is to prepare students for the AB version of the Advanced Placement Computer Science examination.

The prerequisite for this course is successful completion of Computer Science III [AP CS A].

This course uses both *Java Concepts* as the primary text and the course support materials from the textbook author's website, including progressively structured labs, review questions, programming exercises, and supplementary programming projects. All end of chapter material is used to build knowledge and check understanding in preparation for chapter tests. Test questions are drawn from the test bank on the textbook author's website and the Barron's AP<sup>®</sup> Study Guide listed as a supplementary text.

Hands-on work consists of the completion of structured labs included in the text course package, assignments of several more open-ended programming projects that are found at the end of each chapter of the textbook and teacher-made projects.

A Java Style Guide is provided the first week of the course and is used throughout the course as a basis for evaluating students' programs and their adherence to acceptable style and documentation.

## Primary Text:

Horstmann, Cay; *Java Concepts, AP<sup>®</sup> Edition*; John Wiley & Sons, Inc.; 2005; 4<sup>th</sup> ed.

## Supplementary Texts:

AP<sup>®</sup> *GridWorld Case Study*, Princeton, NJ: The College Board, 2006.

Teukolsky, Roselyn. *Barron's Advanced Placement (AP<sup>®</sup>) Exam Computer Science 2007 Levels A and AB*, Hauppauge, NY: Barron's Educational Series, 2006.

## Course Plan [C2]

Unit (Weeks)	Topic	Required Readings	Major Assignments and Assessments
1 (1 – 3)	Java Review [C4][C8][C9]	<i>Java Concepts</i> Ch.1,2,3,4,	• Policies & Rules

	<ul style="list-style-type: none"> <li>• Review major hardware components, system software, types of systems, and responsible use of computer systems</li> <li>• Review variable, classes and objects, calling methods, parameters and return values, API documentation, difference between objects and object references</li> <li>• Design and implement classes and simple methods, purpose and use of constructors, accessing instance fields and local variables, documentation comments</li> <li>• Fundamental data types</li> <li>• Conditional control – if statements, comparisons, Boolean operators and variables</li> <li>• Iteration – while, for and do statements, infinite loops, nested loops</li> <li>• Interfaces and polymorphism</li> <li>• Inheritance</li> </ul>	6,7,11,13	<ul style="list-style-type: none"> <li>• Install Java/Eclipse @ home</li> <li>• Exercise P1.7 p. 31</li> <li>• Exercise P2.8 p. 61</li> <li>• Exercise P6.8, 6.9 p. 227</li> <li>• Sieve of Eratosthenes</li> <li>• Java Review Test</li> </ul>
2 (4 – 5)	<p>ArrayLists and Arrays [C4][C5]</p> <ul style="list-style-type: none"> <li>• One- and Two-dimensional arrays</li> <li>• ArrayLists</li> <li>• Generic Classes</li> <li>• Wrappers, auto-boxing, the enhanced for loop</li> <li>• Common array algorithms – traversals, insertions, deletions, iterators</li> </ul>	<p><i>Java Concepts</i> p. 279 – 311 p. 842 – 853</p>	<ul style="list-style-type: none"> <li>• Project 8.2 p. 320 – The Game of Life</li> <li>• Snakes or Treasures Game</li> <li>• Test on ArrayLists &amp; Arrays</li> </ul>

<b>Unit (Weeks)</b>	<b>Topic</b>	<b>Required Readings</b>	<b>Major Assignments and Assessments</b>
3 (6 – 8)	Recursion [C4][C5] <ul style="list-style-type: none"> <li>• The method of recursion and “thinking recursively”</li> <li>• Relationship between recursion and iteration</li> <li>• Analyze problems that are easier to solve by recursion than by iteration</li> <li>• Using recursive helper methods</li> <li>• How the use of recursion affects the efficiency of an algorithm</li> </ul>	<i>Java Concepts</i> p. 663 – 693	<ul style="list-style-type: none"> <li>• Exercises P18.1, P18.2, P18.3, P18.4, P18.5 p. 695</li> <li>• Exercise P18.8 ( Area of a Polygon) p. 696</li> <li>• Exercise P18.10 p. 696</li> <li>• Test on Recursion</li> </ul>
4 (9)	GridWorld Case Study Review [C7]	GridWorld Ch. 1 and 2	<ul style="list-style-type: none"> <li>• Exercise 4 page 13</li> </ul>
5 (10 – 13)	Sorting, Searching, and Big-O [C5] <ul style="list-style-type: none"> <li>• Selection, insertion, merge, and quicksort sorts</li> <li>• Sequential and binary searches</li> <li>• Big-Oh notation</li> <li>• Worst-case and Average-case time and space analysis</li> </ul>	<i>Java Concepts</i> p.704 – 735	<ul style="list-style-type: none"> <li>• Timing Comparisons</li> <li>• Binary Search Lab</li> <li>• Test on Sorting/Searching</li> </ul>
6 (14 – 18)	Linked Lists [C3][C4][C5][C6] <ul style="list-style-type: none"> <li>• Singly, doubly, and circularly linked lists</li> <li>• Linked lists provided by the standard library</li> <li>• Iterators to traverse linked lists</li> <li>• Distinguishing between abstract and concrete data types</li> <li>• Efficiency of fundamental operations, such as traversals, insertions, and deletions on lists and arrays</li> </ul>	<i>Java Concepts</i> p.741 – 761	<ul style="list-style-type: none"> <li>• Exercises P20.1, P20.2 p. 769</li> <li>• Create a Linked List class using ListNode class</li> <li>• Exercise P20.3 p. 768</li> <li>• Project 20.1 p. 771</li> </ul>
7 (19)	Review for Midterm Examination	<i>Java Concepts</i>	Midterm Examination

<b>Unit (Weeks)</b>	<b>Topic</b>	<b>Required Readings</b>	<b>Major Assignments and Assessments</b>
8 (20 – 23)	Stacks and Queues [C4][C5][C6][C7] <ul style="list-style-type: none"> <li>• Stacks</li> <li>• Queues</li> </ul>	<i>Java Concepts</i> p.762-766	<ul style="list-style-type: none"> <li>• Postfix Evaluation</li> <li>• Random Maze</li> <li>• Stacks and Queues implemented by a Circularly Linked List</li> <li>• Test - Linked Lists, Stacks, &amp; Queues</li> </ul>
9 (24)	GridWorld Case Study [C7]	GridWorld Ch. 3	<ul style="list-style-type: none"> <li>• Group Activity page 24</li> </ul>
10 (25 – 27)	Trees [C3][C4][C5][C6][C7] <ul style="list-style-type: none"> <li>• Binary trees</li> <li>• Tree sets and tree maps</li> </ul>	<i>Java Concepts</i> p.796 – 814	<ul style="list-style-type: none"> <li>• Exercise P21.8, P21.13, P21.15 p. 837 - 838</li> </ul>
11 (28)	GridWorld Case Study [C7]	GridWorld Ch. 4	<ul style="list-style-type: none"> <li>• Group Activity page 33</li> </ul>
12 (29 – 30)	Hashing, Sets, and Maps [C4][C5] <ul style="list-style-type: none"> <li>• Set and map data types</li> <li>• Implementation of hash tables</li> <li>• Programming hash functions</li> </ul>	<i>Java Concepts</i> p.776 – 814	<ul style="list-style-type: none"> <li>• Exercise P21.1, P21.4, P21.5, P21.6, 21.9, 21.10 on p. 836-837</li> </ul>
13 (31)	GridWorld Case Study [C7] <ul style="list-style-type: none"> <li>• Understand the BoundedGrid implementation</li> <li>• Consider alternate implementations for the BoundedGrid class</li> <li>• Understand the UnboundedGrid implementation</li> <li>• Consider and code alternate implementations for the UnboundedGrid class</li> </ul>	GridWorld Ch. 5	<ul style="list-style-type: none"> <li>• Exercises 1, 2, 3 pages 38 – 40</li> </ul>
14 (32 – 33)	Priority Queue and TreeMap [C4][C5] <ul style="list-style-type: none"> <li>• Implement the priority queue data type</li> <li>• Heaps and Heapsort</li> </ul>	<i>Java Concepts</i> p. 815	<ul style="list-style-type: none"> <li>• Emergency Room Program</li> <li>• Test on Hashing, Sets, Maps</li> </ul>
15 (34)	Review for AP Exam	n/a	<ul style="list-style-type: none"> <li>• AP Practice Exam</li> </ul>
16 (35 – 39)	Team Projects	n/a	<ul style="list-style-type: none"> <li>• Bouncing Ball Project</li> </ul>
17 (40)	Review for Final Exam	<i>Java Concepts</i>	<ul style="list-style-type: none"> <li>• Final Examination</li> </ul>

## Curriculum Designators and Requirements

C1	The teacher has read the most recent AP Computer Science Course Description, available as a free download at <a href="http://apcentral.collegeboard.com/compsci">apcentral.collegeboard.com/compsci</a> .
C2	The course includes all of the topics listed in the “Computer Science A” column of the Topic Outline in the <i>AP Computer Science Course Description</i> .
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 <i>AP Computer Science Course Description</i> .
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 <i>AP Computer Science Case Study</i> posted on AP Central.
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