AP* Computer Science Topics	Sections Where Taught
I. OBJECT-ORIENTED PROGRAM DESIGN	
The overall goal for designing a piece of software (a computer program) is to correctly solve the given problem. At the same time, this goal should encompass specifying and designing a program that is understandable, can be adapted to changing circumstances, and has the potential to be reused in whole or in part. The design process needs to be based on a thorough understanding of the problem to be solved.	
A. Program design	
1. Read and understand a problem description, purpose, and goals.	Sections 1.3, 3.0, 3.7
2. Apply data abstraction and encapsulation.	Sections 2.1, 4.1, 7.7
3. Read and understand class specifications and relationships among the classes ("is-a," "has-a" relationships).	Sections 4.5, 7.0
4. Understand and implement a given class hierarchy.	Sections 7.0, 7.2
5. Identify reusable components from existing code using classes and class libraries.	Sections 2.0, 7.4
B. Class design	
1. Design and implement a class.	Sections 4.0–4.4, 5.0, 5.1, 5.4
2. Choose appropriate data representation and algorithms.	Sections 3.7, 5.4
3. Apply functional decomposition.	Sections 4.2, 4.4
4. Extend a given class using inheritance.	Sections 7.0, 7.1, 7.3, 7.4
II. PROGRAM IMPLEMENTATION	
The overall goals for program implementation parallel those of program design. Classes that fill common needs should be built so that they can be reused easily in other programs. Object-oriented design is an important part of program implementation.	
A. Implementing techniques	
1. Methodology	
a. Object-oriented development	Section 2.0
b. Top-down development	Section 3.0
c. Encapsulation and information hiding	Section 4.1
d. Procedural abstraction	Section 4.2
B. Programming constructs	
1. Primitive types vs. objects	Sections 2.0, 2.1, 2.4
2. Declaration	
a. Constant declarations	Section 2.3
b. Variable declarations	Section 2.3
c. Class declarations	Section 4.1
d. Interface declarations	Section 5.3
e. Method declarations	Section 4.2
f. Parameter declarations	Section 4.2
3. Console output (`System.out.print/println`)	Section 2.1
4. Control	
a. Methods	Section 4.2
b. Sequential	Section 3.1
c. Conditional	Section 3.2
d. Iteration	Sections 3.5, 3.6
e. Understand and evaluate recursive methods	Sections 8.0–8.2

AP* Computer Science Topics	Sections Where Taught
C. Java library classes (included in the A-level AP* Java Subset)	
Object	Section 7.2
Comparable	Section 5.3
Integer	Section 2.7
Double	Section 2.7
String	Section 2.7
Math	Section 2.8
Random	Section 2.8
ArrayList	Section 6.7
III. PROGRAM ANALYSIS	
The analysis of programs includes examining and testing programs to determine whether they correctly meet their specifications. It also includes the analysis of programs or algorithms in order to understand their time and space requirements when applied to different data sets.	
A. Testing	
1. Test classes and libraries in isolation	Section 5.4
2. Identify boundary cases and generate appropriate test data	Section 5.4
3. Perform integration testing	Section 5.4
B. Debugging	
1. Categorize errors: compile-time, run-time, logic	Section 1.4
2. Identify and correct errors	Sections 1.4, 5.4
3. Techniques: use a debugger, add extra output statements, hand-trace code	Sections 1.4, 5.4
C. Understand and modify existing code	**Learned throughout the text**
D. Extend existing code using inheritance	**Sections 7.0, 7.1, 7.3, 7.4**
E. Understand error handling	
1. Understand runtime exceptions	Section 5.2
F. Reason about programs	
1. Pre- and post-conditions	Section 4.2
2. Assertions	Section 4.2
G. Analysis of algorithms	
1. Informal comparisons of running times	Sections 6.2, 6.4
2. Exact calculation of statement execution counts	Sections 6.2, 6.4
H. Numerical representations and limits	
1. Representations of numbers in different bases	Section 1.0, Appendix E
2. Limitations of finite representations (e.g., integer bounds, imprecision of floating-point representations, and round-off error)	Sections 2.4, 3.3
IV. STANDARD DATA STRUCTURES	
Data structures are used to represent information within a program. Abstraction is an important theme in the development and application of data structures.	
A. Simple data types (int, boolean, double)	**Section 2.4**
B. Classes	**Sections 2.0, 4.1**

AP* Computer Science Topics	Sections Where Taught
C. Lists	**Sections 6.7, 9.0–9.1**
D. Arrays	**Sections 6.0, 6.1**
V. STANDARD ALGORITHMS	
Standard algorithms serve as examples of good solutions to standard problems. Many are intertwined with standard data structures. These algorithms provide examples for analysis of program efficiency.	
A. Operations on data structures previously listed	
1. Traversals	Sections 6.0, 6.1, 6.7
2. Insertions	Sections 6.0, 6.1, 6.7
3. Deletions	Sections 6.0, 6.1, 6.7
B. Searching	
1. Sequential	Section 6.2
2. Binary	Section 6.2
C. Sorting	
1. Selection	Section 6.3
2. Insertion	Section 6.3
3. Mergesort	Section 8.3
VI. COMPUTING IN CONTEXT	
An awareness of the ethical and social implications of computing systems is necessary for the study of computer science. These topics need not be covered in detail but should be considered throughout the course.	
A. System reliability	**Appendix F**
B. Privacy	**Appendix F**
C. Legal issues and intellectual property	**Appendix F**
D. Social and ethical ramifications of computer use	**Appendix F**

Acknowledgments

We are most grateful to the AP* teachers and AP* students from around the world who have provided their feedback on the previous edition of this book. We are pleased to see the depth of the faculty's concern for their AP* students and the AP* students' thirst for knowledge. Your comments and questions are always welcome.

Michael Hirsch and Stephanie Sellinger, our editors at Addison-Wesley, went above and beyond the call of duty to ensure that the book met the highest quality standards. Their support and enthusiasm are greatly appreciated. We are also grateful to Andrea Sheehan and Courtney Marsh with Pearson School for making sure that AP* teachers understand the pedagogical advantages of this text. The devotion that the Addison-Wesley and Pearson folks show to their books is evident in the high-quality results.