Lake Zurich High School
Mathematics Department

AP – Computer Science

Course Description

Prerequisites:	CP or H-Geometry AND A GRADE OF B OR HIGHER IN HONORS COMPUTER PROGRAMMING OR PERMISSION FROM THE DEPARTMENT CHAIR.
Open To:	10, 11, 12
Credit:	1 unit
Level:	Advanced Placement

The Advanced Placement Computer Science course is a one-year course in computer science that is designed to prepare the student to take the Advanced Placement Examination (offered each May) in an attempt to receive advanced placement and/or credit in the freshman year of college. Because the development of computer programs to solve problems is a skill fundamental to the study of computer science, a large part of the course is built around the development of computer programs or parts of programs that correctly solve a given problem. The course also emphasizes the design issues that make programs understandable, adaptable, and, when appropriate, reusable. At the same time, the development of useful computer programs and program modules is used as a context for introducing other important concepts in computer science, including the development and analysis of algorithms, the development and use of fundamental data structures, and the study of standard algorithms and typical applications. In addition, an understanding of the basic hardware and software components of computer systems and the responsible use of these systems are integral parts of the course.

Textbook

Title:	Big Java
Publisher:	John Wiley & Sons, Inc
Author(s)	Cay Horstmann
Copyright date:	2006
ISBN number:	0-471-69703-6

Course Syllabus & Objectives

Unit/ Time	Topics	Objectives/(Resource)	Activities	Assessment
Unit 1 2 weeks	Reintroduce Programming Develop Interacting Classes	- Data types (2.1, 2.5, 4.1, 4.2 & Handouts) - Operations (4.3, 4.4) - Decision Making (6.1- 6.4.3) - Repetition (7.1 7.2) - Arrays (8.1) - Introduce Object Oriented Design (objects, methods, attributes) (2.3, 2.4)	- Greeting program - Sum Reciprocals Program - Data types & operations examples - Decision making examples - Write Program - Loops - Interacting classes examples - Write Program – Sort an array	- Interacting classes assignment - Quiz: Results of calculations - Quiz: Results of loops and decisions - Quiz: Identifying object terminology within program
Unit 2 1 weeks	Develop Constructor, Accessor, and Mutator Methods	- Constructing Objects (2.6 – 2.10, 3.1 – 3.8) - Write Methods developing proficiency using Object Oriented Programming (increasing understanding of objects, object references, instantiation, immutability)	- Program Constructing Multiple Instances – staged development: use interacting classes, rectangle object, multiple instances, constructor, accessor, and mutator instance methods, object references. - Object References Exercise	- Program as described - Quiz: Objects & methods
Unit 3 1 weeks	Two – Dimensional Arrays	- Declare and Instantiate 2-D array objects (8.6, 8.7)` - Nested Loops - Write Methods	- Analyze TicTacToe Program - Change TicTacToe Program using 2-D arrays, nested loops, and methods (Battleship)	- Program as described (Battleship) - Quiz from publisher's test bank

Unit/ Time	Topics	Objectives/(Resource)	Activities	Assessment
Unit 4 2 weeks	Stack Interface and ArrayList Implementation	- Develop ArrayList implementation of the Stack interface (20.4) - Understand behavior of a stack and identify situations where a stack is appropriate - Stack operations: traversals, insertions, deletions	- Discuss arraylist example - Discuss strategy to implement traversals, insertions, deletions in ArrayList - Compare & contrast above with that of an array - LIFO (last in, first out) examples - Walk through Stack logic to solve computations - Review exercises	- Program (Postfix Evaluation) - Program to simulate shuffling a deck of cards - Quiz from publisher's test bank
Unit 5 2 weeks	Queue Interface and ArrayList implementation	- Develop ArrayList implementation of the Queue interface (20.4) - Understand the behavior of a queue and Identify situations where a queue is the appropriate data structure to use - Operations on Queues: traversals, insertions, deletions	- FIFO (first in, first out) examples - Discuss strategy to implement traversals, insertions & deletions - Review selection sort logic	- Program to implement selection sort using queues - Program to implement radix sort using queues - Quiz from publisher's test bank
Unit 6 2 weeks	Java Marine Biology Simulation	- Review Testing/Debugging - Review class modification - Inheritance	- Walk through interacting classes and inheritance - Modify JMBS to add a class	- Predator Fish Class
Unit 7 2 weeks	Recursion	- Design and implement recursive solutions to problems (18.1 – 18.5) - Recursion: Simple and Binary - Implement a Recursive Merge Sort - Implement a Binary Search using a recursive method	- Recursion With Web Animations - Walk through code examples in text - Exercise P18.8 – application of mathematical recursion and point comparisons - Exercise P18.10 – application of string recursion	- Programs from exercises P18.8 & P18.10 - Quiz from publisher's test bank
Unit/	Topics	Objectives/(Resource)	Activities	Assessment

Time				
Unit 8 4 weeks	Linked Lists using ListNode class	- Create a LList class using ListNode class (20) - Build a linked list like a stack, queue, or in order using a recursive helper method - Implement Operations on LList object to perform traversals, insertions, and deletions - Re-implement Stack, Queue, and PriorityQueue interfaces using LList class	- Demonstrate Linked List with Barrel of Monkeys - Walk through code examples of linked list of alphabetical names - Program – Doubly Linked List and Circularly Linked List Maintenance - Program – Re-implement Stack and Queue interfaces with Circularly Linked List	- Programs as described - Quiz from publisher's test bank
Unit 9 2 weeks	AP Review	- Object Oriented Programming - Stacks - Queues - Arrays - ArrayList - Recursion - Linked Lists	- Exercises from supplemental guide and AP College Board Website	- Evaluation of exercises from supplemental guide and AP College Board Website
Unit 10 4 weeks	Trees	- Learn tree terminology (21.5-21.6) - Distinguish between general trees, binary trees, binary search trees (BST), and heaps - Operations on a BST: traversals, insertions, deletions - Create a BST class using TreeNode class - Build a BST using a recursive helper method	- Binary Tree Activity - Binary Tree Animation - Binary Tree Traversals - Heap Animation - Recursive Tree Methods From Old AP exams	- Program – Recursive Tree Method

Unit/ Time	Topics	Objectives/(Resource)	Activities	Assessment
Unit 11 2 weeks	Java Marine Biology Simulation	- Understand the BoundedEnv implementation (Ch 5 MBS) - Consider alternate implementations to the BoundedEnv class - Understand the UnboundedEnv implementation - Consider and code alternate implementations for the UnboundedEnv class	- Chapter 5 of the Marine Biology Simulation narrative - Selected Exercises from Chapter 5 of the JMBS narrative	- Evaluation of analysis exercises from Chapter 5 of JMBS
Unit 12 1 week	Java Linked List class and Iterators	- Understand the List interface and the LinkedList implementation (Ch 20) - Perform basic operations on a linked List object: traversals, insertions, deletions, iterators - Understand the difference between an iterator and a list iterator	- Exercises P20.1, P20.2, P20.3	- Unit 12 Program Linked List Employees

Unit/ Time	Topics	Objectives/(Resource)	Activities	Assessment
Unit 13 2 weeks	Hashing, Sets, and Maps	- Understand the features of a set and the appropriate use of a set (Ch 21) - Understand the features of a map and the appropriate use of a map - Introduce and use the Set and Map interfaces - Understand the purpose of a key - Understand the difference between a key and a value - Use HashSet and TreeSet (21.7) implementations of the Set interface - Use HashMap and TreeMap (21.7)implementations of the Map interface - Understand the expected running time of the HashSet and TreeSet implementations of the Set interface - Understand the expected running time of the HashMap and TreeMap implementations of the Map interface	- Walk through heap as a priority queue - Discuss additions, deletions - Walk through heap sort algorithm - Exercises P21.1, P21.4, P21.5, P21.6, P21.9, P21.10	- Evaluate program exercises

Unit/ Time	Topics	Objectives/(Resource)	Activities	Assessment
Unit 14 1 week	PriorityQueue interface and implementation with TreeMap or TreeSet	- Understand the difference between a Priority Queue and a Queue (Ch 21 p. 1107) - Study different implementations of a Priority Queue and discuss their performance tradeoffs - Develop TreeMap and TreeSet implementation of the PriorityQueue interface - Perform operations on PriorityQueues: traversals, insertions, deletions	- Priority Queue Animation - Emergency Room Program - Wordlist revised using a TreeMap	- Emergency Room Program - Wordlist Program
Unit 15 2 weeks	Sorting, Searching, and Big-Oh	- Perform complexity analysis of algorithms using Big-Oh notations - Recognize typical Big-Oh functions and be able to order them in order of increasing growth rate - Understand the following Sorting Algorithms: Merge, Selection, Heap (21.9 – 21.10), Quick, and Insertion - Understand the following Searching Algorithms: Sequential and Binary - Understand the Big-Oh analysis for the above sorting and searching algorithms, worst-case , average-case time and space analysis	- Binary Search: Dictionary Demo - Run sorting routines for different sized datasets - Collect timings and compare (timing sheets) - Discuss different growth rates - Racing Sorts Page (web animation)	- Quiz on growth rates and typical Big-Oh functions

Unit/ Time	Topics	Objectives/(Resource)	Activities	Assessment
Unit 16 3 weeks	Review for AP Exam	- Review MBS using exercises in MBS - Appropriate Free Response - Appropriate Multiple Choice Questions	- Additional exercises and programs from textbook & supplemental materials	- Evaluation of exercises and programs
Unit 17 2 weeks	Team Projects	- Website programming - Game programming	- Interactive game or website developed in a team environment with all members developing interactive classes	- Game or website developed by team

Unit Objectives

Unit 1 - Reintroduce Programming & Develop Interacting Classes

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
I.B.1 Design and implement a set of interacting classes
II.B.1 Primitive types vs. objects
II.B.2a Constant declarations
II.B.2b Variable declarations
II.B.2c Class declarations
II.B.2e Method declarations
II.B.3 Console output
II.B.4a Methods
II.B.4b Sequential
II.B.4c Conditional
II.B.4d Iteration
II.B.4e Recursion
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
III.F.1 Pre- and post-conditions
III.H.1 Representations of numbers in different bases
IV.A Simple data types (integer, boolean, double)
IV.B Classes
IV.C One-dimensional arrays
V.B.1 Sequential

Unit 2 - Develop Constructor, Accessor, and Mutator Methods

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
I.A.3 Decompose a problem into classes: define relationships and responsibilities of those classes
I.A.4 Understand and implement a given class hierarchy
I.A.5 Identify reusable components from existing code using classes and class libraries
I.B.2 Design an interface
II.A.1a Object-Oriented development
II.A.1b Top-down development
II.B.2d Interface declarations
II.B.2e Method declarations
II.B.2f Parameter declarations
II.B.3. Console output
II.B.4a Methods
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
IV.B Classes

Unit 3 - Two – Dimensional Arrays

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
I.B.4 Apply functional decomposition
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
IV.D Two-dimensional arrays
V.B.1 Sequential

Unit 4 - Stack Interface and ArrayList Implementation

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
I.B.3 Choose appropriate advanced data structures and algorithms
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
III.E.2 Throw runtime exceptions
IV.C One-dimensional arrays
IV.F Stacks
V.A.1 Traversals
V.A.2 Insertions
V.A.3 Deletions
V.A.4 Iterators
V.B.1 Sequential

Unit 5 - Queue Interface and ArrayList implementation

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
I.B.3 Choose appropriate advanced data structures and algorithms
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
III.E.2 Throw runtime exceptions
IV.G Queues
V.A.1 Traversals
V.A.2 Insertions
V.A.3 Deletions
V.A.4 Iterators
V.B.1 Sequential

Unit 6 - Java Marine Biology Simulation

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
I.A.2 Apply data abstraction and encapsulation
I.A.3 Decompose a problem into classes: define relationships and responsibilities of those Classes
I.A.4 Understand and implement a given class hierarchy
I.B.5 Extend a given class using inheritance
II.A.1c Encapsulation and information hiding
II.A.1d Procedural Abstraction
III.A.1 Test classes and libraries in isolation
III.A.2 Identify boundary cases and generate appropriate test data
III.A.3 Perform integration testing
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.D Extend existing code using inheritance
III.E.1 Understand runtime exceptions
III.F.2 Assertions

Unit 7 - Recursion

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
I.A.5 Identify reusable components from existing code using classes and class libraries
II.B.4e Recursion
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
IV.H Trees
V.B.2 Binary Search

Unit 8 - Linked Lists using ListNode class

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
I.B.3 Choose appropriate advanced data structures and algorithms
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
III.E.2 Throw runtime exceptions
IV.E Linked lists (singly, doubly, circular)
V.A.1 Traversals
V.A.2 Insertions
V.A.3 Deletions
V.A.4 Iterators

Unit 9 - AP Review

Topics Covered by the AP Computer Science Exam
Topics from previous units as needed

Unit 10 - Trees

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
I.B.3 Choose appropriate advanced data structures and algorithms
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
V.B.2 Binary Search

Unit 11 - Java Marine Biology Simulation

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
II.C Java library classes (AP Java Subset)
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions

Unit 12- Java Linked List class and Iterators

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
I.B.3 Choose appropriate advanced data structures and algorithms
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
V.A.1 Traversals
V.A.2 Insertions
V.A.3 Deletions
V.A.4 Iterators

Unit 13 - Hashing, Sets, and Maps

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
IV.K Sets
IV.L Maps
V.B.3 Hashing

Unit 14 - PriorityQueue interface and implementation with TreeMap or TreeSet

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
IV.J Priority Queues
V.A.1 Traversals
V.A.2 Insertions
V.A.3 Deletions
V.A.4 Iterators

Unit 15 - Sorting, Searching, and Big-Oh

Topics Covered by the AP Computer Science Exam
I.A.1 Specify the purpose and goals for a problem
III.B.1 Categorize errors: compile-time, run-time logic
III.B.2 Identify and correct errors
III.B.3 Techniques: use a debugger, add extra output statements, hand-trace code
III.E.1 Understand runtime exceptions
III.G.1 Informal comparisons of running times
III.G.2 Exact calculation of statement execution counts
III.G.3 Big-Oh notation
III.G.4 Worst-case and average-case time and space analysis
III.H.2 Limitations of finite representations
IV.I Heaps
V.C.1 Selection
V.C.2 Insertion
V.C.3 Mergesort
V.C.4 Quicksort
V.C.5 Heapsort

Unit 16 - Review for AP Exam

Topics Covered by the AP Computer Science Exam
Topics from previous units as needed

Unit 17 – Team Projects

Topics Covered by the AP Computer Science Exam
I.B.3 Choose appropriate advanced data structures and algorithms
All topics from previous units

CrossReference to Topics Covered by the AP Computer Science Exam

Unit	Topics Covered by the AP Computer Science Exam
	I. Object-Oriented Program Design
	A. Program Design
1 - 17	1. Specify the purpose and goals for a problem.
6	2. Apply data abstraction and encapsulation
2, 6	3. Decompose a problem into classes: define relationships and responsibilities of those classes
2, 6	4. Understand and implement a given class hierarchy
2,7	5. Identify reusable components from existing code using classes and class libraries
	B. Class design
1	1. Design and implement a set of interacting classes
2	2. Design an interface
4, 5, 8, 10, 12	3. Choose appropriate advanced data structures and algorithms
3	4. Apply functional decomposition
6	5. Extend a given class using inheritance
	II. Program Implementation
	A. Implementation techniques
	1. Implementation techniques
1, 2	a. Object-Oriented development
1, 2	b. Top-down development
6	c. Encapsulation and information hiding
6	d. Procedural abstraction
	B. Programming constructs
1	1. Primitive types vs. objects
1	2. Declaration
1	a. Constant declarations
1	b. Variable declarations
1	c. Class declarations
2	d. Interface declarations
1, 2	e. Method declarations
2	f. Parameter declarations
1, 2	3. Console output
2	4. Control
1, 2	a. Methods
1	b. Sequential
1	c. Conditional
1	d. Iteration
7	e. Recursion
1 - 17	C. Java library classes (AP Java Subset)

Unit	Topics Covered by the AP Computer Science Exam
	III. Program Analysis
	A. Testing
6	1. Test classes and libraries in isolation
6	2. Identify boundary cases and generate appropriate test data
6	3. Perform integration testing
6	B. Debugging
1 - 12	1. Categorize errors: compile-time, run-time logic
1 – 12	2. Identify and correct errors
1 – 12	3. Techniques: use a debugger, add extra output statements, hand-trace code
1 – 12	C. Understand and modify existing code
6	D. Extend existing code using inheritance
	E. Understand error handling
1 – 17	1. Understand runtime exceptions
4, 5, 8	2. Throw runtime exceptions
	F. Reason about programs
1	1. Pre- and post-conditions
	2. Assertions
	G. Analysis of algorithms
15	1. Informal comparisons of running times
15	2. Exact calculation of statement execution counts
15	3. Big-Oh notation
15	4. Worst-case and average-case time and space analysis
	H. Numerical representations and limits
1	1. Representations of numbers in different bases
15	2. Limitations of finite representations
	IV. Standard Data Structures
1	A. Simple data types (integer, boolean, double)
1, 2	B. Classes
1, 4	C. One-dimensional arrays
3	D. Two-dimensional arrays
8	E. Linked lists (singly, doubly, circular)
4	F. Stacks
5	G. Queues
7	H. Trees
15	I. Heaps
14	J. Priority Queues
13	K. Sets
13	L. Maps
	V. Standard Algorithms
	A. Operations on AB-level data structures previously listed
4, 5, 8, 12, 14	1. Traversals
4, 5, 8, 12, 14	2. Insertions
4, 5, 8, 12, 14	3. Deletions
4, 5, 8, 12, 14	4. Iterators

Unit	Topics Covered by the AP Computer Science Exam
	B. Searching
1, 3, 4, 5	1. Sequential
7, 10	2. Binary
13	3. Hashing
	C. Sorting
15	1. Selection
15	2. Insertion
15	3. Mergesort
15	4. Quicksort
15	5. Heapsort