AP® Computer Science AB
Syllabus 2

Course Design

This course extends the concepts of AP® Computer Science A with an emphasis on object-oriented programming (OOP) and design. Students are expected to gain high proficiency in creating and implementing classes including creating new classes using inheritance and implementing interfaces. Topics studied in this course include one and two dimensional arrays, advanced data structures (including trees, linked lists, sets, and maps), algorithms, algorithmic analysis, and object oriented design.

All AP Computer Science classes are taught in a computer lab. Each student has a computer workstation. Lectures are interactive—many students take notes electronically and program sample classes with the instructor. Most class periods are about one third lecture and two thirds lab or written work.

Computer Facilities

The Computer Science Department has two labs of 30 PCs. Each lab also includes scanners, CD-RW drives, a data projector, and laser printers. Students have accounts on a networked server and store all of their class files in their accounts. Lab facilities are open before and after school and some lunch periods for student use.

Textbooks and Resources

- Horstmann, Cay. *Big Java*. New York: Wiley, 2002.
- Weiss, Mark Allen. *Data Structures and Problem Solving Using Java*, 2nd ed. New York: Addison Wesley, 2002.
- The College Board's GridWorld Case Study
- AP Central®: Computer Science AB Quick Reference Guide

Course Outline [C2]

Note: Designing, implementing, and using classes are ongoing topics taught throughout the course.

First Six Weeks	
Topic: Reintroduce Programming (2 weeks) [C8] [C9] • Understand the hardware/software components of computers, their relationships, and their roles within a system • User responsibility of computer systems and legality of district use of the network • Students are required to research one of the following topics and make a presentation. Suggested topics include: Software piracy, copyright and intellectual property, network security, malware, spamming	**Objectives:** [C3] [C6] • Make connections to previous knowledge and program techniques • Develop and implement a set of interacting classes **Reading:** *Big Java* Chapters 2, 3, 5, 6, 7, 9, 11, 13 **Programs:** • Lottery: Write a set of classes that models a lottery machine similar to the lottery drawing seen on TV. Design of these classes includes encapsulation, polymorphism, and composition among classes • Sieve of Eratosthenes: Write a class that will produce an array of prime numbers in a given range of 1 to n • Word list: Implement a set of classes that stores a searchable list of words
Topic: Two Dimensional Arrays (2 weeks)	**Objectives:** [C4] • Declare and instantiate 2-D array objects • Use nested loops to manipulate 2-D arrays • Write methods using 2-D arrays **Readings:** *Big Java*, pp. 544–548 **Program:** Snakes and Treasure Game: Write a set of classes that will simulate the playing of a Snakes and Treasure game
Topic: Stacks (2 weeks)	**Objectives:** [C4] [C5] • Understand the behavior of a stack and identify situations where a stack is the appropriate data structure to use. • Operations on Stacks: traversals, insertions, deletions **Reading:** *Big Java*, pp. 757–760 **Reference:** AB Quick Reference: Stack Class **Programs:** Postfix evaluation: Write a postfix evaluation class that will evaluate a given postfix expression

C2—The course includes all of the topics listed in the "Computer Science AB" column of the topic outline in the *AP Computer Science Course Description*.

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 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.)

C8—Evidence of Curricular Requirement: 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—Evidence of Curricular Requirement: The course teaches students to recognize the ethical and social implications of computer use.

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.

Second Six Weeks	
Topic: Queue Interface (2 weeks)	**Objectives:** [C4] [C5] [C6] [C7] • 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 **Reading:** *Big Java*, pp. 757–760 **Reference:** AB Quick Reference: Queue interface **Programs:** • Selection Sort using Queues: Write a static class that sorts an array using the Selection Sort. The Selection class will use queues instead of an array to perform the sort. • Radix Sort using Queues: Note: The Radix Sort algorithm sorts the numbers based on a different digit, working from the least to the most significant digit. It uses an intermediate data structure, ques, an array of ten queues. • *Big Java*, p. 765: P19.9 (Circular Array Queue)
Topic: GridWorld Case Study Parts 1–4 (2 weeks)	**Objectives:** • Review Testing • Class modification • Inheritance **Reference/Reading:** GridWorld Case Study, Parts 1–4
Topic: Recursion (2 weeks)	**Objectives:** • Design and implement recursive solutions to problems • Recursion: Simple recursion, binary recursion • Implement a Recursive Merge Sort • Implement a Binary Search using a recursive method **Reading:** *Big Java*, pp. 667–698 **Programs:** • *Big Java*, p. 699: P17.8 • *Big Java*, p. 700: P17.10

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®.

Third Six Weeks	
Topic: Linked Lists using ListNode class (4 weeks)	**Objectives:** [C3] [C4] [C5] [C6] Create a LList class using ListNode class Build a linked list like a stack, queue, or inorder using a recursive helper method Implement Operations on LList object to perform traversals, insertions, and deletions **Readings:** *Big Java*, pp. 737–753 **Reference:** AB Quick Reference: ListNode class **Programs:** • Doubly Linked List and Circularly Linked List Maintenance: Reimplement the ListNode class to include another private instance variable. • Josephus Problem: Weiss, pp. 429–431
Topic: Review (2 weeks)	**Objectives:** Review topics to this point: • OOP • Stacks • Queues • Linked Lists • arrays • ArrayList • Recursion **Assignments:** Sample AP free-response and multiple-choice questions found in *Big Java Student Guide*
Fourth Six Weeks	
Topic: Trees (4 weeks)	**Objectives:** [C3] [C4] [C5] [C6] [C7] • Learn tree terminology • Distinguish between general trees, binary trees, binary search trees, and heaps • Operations on a BST: traversals, insertions, deletions • Create a BST class using TreeNode class • Build a BST using a recursive helper method **Reading:** *Big Java*, pp. 790–798 **Reference:** AB Quick Reference: TreeNode class **Program:** Recursive Tree methods from released AP Exams

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®.

Topic: GridWorld Case Study (2 weeks)	**Objectives:** [C7] • Understand the BoundedGrid implementation • Consider alternate implementations for the BoundedGrid class • Understand the UnboundedGrid implementation • Consider and code alternate implementations for the UnboundedGrid class **Reference/Reading:** Part 5 of the GridWorld Case Study **Program:** Selected exercises from Part 5 of the GridWorld Case Study
Fifth Six Weeks	
Topic: Java LinkedList class and Iterators (1 week)	**Objectives:** [C3] [C4] [C5] [C6] • Understand the List interface and the LinkedList implementation • Perform basic operations on a LinkedList object: traversals, insertions, deletions, iterators • Understand the difference between an iterator and a list iterator **Reading:** *Big Java*, pp. 737–753 **Program:** *Big Java*, pp. 764–765: P19.1, P19.2, P19.3, P19.7
Topic: Hashing, Sets, and Maps (2 weeks)	**Objectives:** [C4] [C5] • Understand the features of a set and the appropriate use of a set • 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 implementations of the Set interface • Use HashMap and TreeMap 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 **Readings:** *Big Java*, pp. 772–801 **Program:** Word list revised using a TreeMap: Implement a set of classes that stores a searchable list of words *Big Java*, p. 807: P20.1, P20.4, P20.5, P20.6, P20.8, P20.9

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®.

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.)

Topic: PriorityQueue (1 Week)	**Objectives:** [C4] [C5] • Understand the difference between a PriorityQueue and a Queue • Study different implementations of a PriorityQueue and discuss their performance tradeoffs • Perform operations on PriorityQueues: traversals, insertions, deletions **Reading:** *Big Java*, pp. 798–801 **Reference:** AB Quick Reference: PriorityQueue interface **Program:** • Emergency Room Program: Given the PriorityQueue class, implement a set of classes that simulate processing patients in a hospital emergency room. Design of these classes involves OO design principles including inheritance.

Topic:
Sorting, Searching, and Big-Oh (2 weeks)
	Objectives:
	• Perform complexity analysis of algorithms using Big-Oh notation
	• Recognize typical Big-Oh functions and be able to order them in order of increasing growth rate:
	  ◦ c
	  ◦ log N
	  ◦ N
	  ◦ N log N
	  ◦ N^2
	  ◦ N^3
	  ◦ 2^N
	• Understand the following sorting algorithms: Merge, Selection, Heap, 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 and average-case time and space analysis
	• Efficiency discussions/comparisons/contrasts between all data structures learned
	Readings:
	Big Java, pp.704–735
	Program:
	• Run sorting routines for different-sized data sets
	• Collect data for timing comparisons

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.

Sixth Six Weeks	
Topic: Preparing for the AP Exam (4 weeks)	**Objectives:** [C6] [C7] • Revisit GridWorld Case Study • Appropriate free-response • Appropriate multiple-choice questions **Reference:** • Free-response questions (appropriately rewritten in Java) from previous AP Exams • *Big Java Student Guide*
Topic: Team Projects (2 weeks)	Students should plan and implement a game or other useful applications

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®.

C3—The course teaches students to design and implement computer-based solutions to problems in a variety of application areas.

An Example of Student Program Assignment

RandomGrid Generator class (JMBS) [C3] [C6] [C7]

Background

This lab will provide you with another opportunity to work with the classes found in the AP Computer Science Case Study.

What You'll Need

- The GridWorld files

Assignment

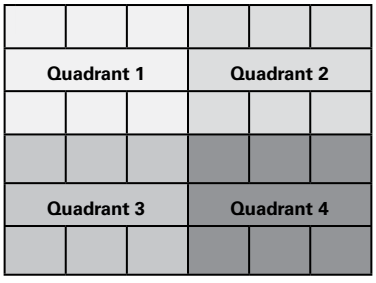
Write a class RandomGrid generator class.

Specifications:

- No constructor for this class; this is a static class—all methods are static.
- One public method—the generate method will accept a Grid object and populate the object with a random number of Bug objects. The number of Bug objects should be no more than 50 percent of the available space in the environment.
- It is possible that the Grid object already contains Bugs.

Extensions:

- Modify the class to add a method, generateQuadrant, that will populate a given quadrant of the environment. Number the quadrants as shown below:



- generateQuadrant should accept an int to identify the quadrant to populate and should randomly populate the quadrant. When the process is completed, the quadrant should have no more than 50 percent of the available spots filled. Remember, there may be Bug objects located in this quadrant before the process begins.