



AP COMPUTER SCIENCE A SYLLABUS

Dave Wittry
Troy High School
Fullerton, California
wittry@trohigh.com
<http://www.trohigh.com/wittry/ap>

Copyright © 2003 College Entrance Examination Board. All rights reserved. College Board, Advanced Placement Program, AP, AP Vertical Teams, APCD, Pacesetter, Pre-AP, SAT, Student Search Service, and the acorn logo are registered trademarks of the College Entrance Examination Board. AP Central is a trademark owned by the College Entrance Examination Board. PSAT/NMSQT is a registered trademark jointly owned by the College Entrance Examination Board and the National Merit Scholarship Corporation. Educational Testing Service and ETS are registered trademarks of Educational Testing Service. Other products and services may be trademarks of their respective owners.

For the College Board's online home for AP professionals, visit AP Central at apcentral.collegeboard.com.



Permission to Reprint Statement

The Advanced Placement Program intends this publication for non-commercial use by AP teachers for course and exam preparation; permission for any other use must be sought from the AP Program. Teachers may reproduce this publication, in whole or in part, **in limited print quantities for non-commercial, face-to-face teaching purposes**. This permission does not apply to any third-party copyrights contained within this publication.

When educators reproduce this publication for non-commercial, face-to-face teaching purposes, the following source line must be included:

AP CS A Syllabus_Wittry. Copyright © 2003 by the College Entrance Examination Board. Reprinted with permission. All rights reserved. www.collegeboard.com.

This material may not be mass distributed, electronically or otherwise. This publication and any copies made from it may not be resold.

The AP Program defines “limited quantities for non-commercial, face-to-face teaching purposes” as follows:

Distribution of up to 50 print copies from a teacher to a class of students, with each student receiving no more than one copy.

No party may share this copyrighted material electronically — by fax, Web site, CD-ROM, disk, e-mail, electronic discussion group, or any other electronic means not stated here. In some cases— such as online courses or online workshops — the AP Program may grant permission for electronic dissemination of its copyrighted materials. All intended uses not defined within “*non-commercial, face-to-face teaching purposes*” (including distribution exceeding 50 copies) must be reviewed and approved; in these cases, a license agreement must be received and signed by the requestor and copyright owners prior to the use of copyrighted material. Depending on the nature of the request, a licensing fee may be applied. Please use the required form accessible online. The form may be found at: <http://www.collegeboard.com/inquiry/cbpermit.html>. For more information, please see AP’s Licensing Policy For AP® Questions and Materials.

School Profile

Troy High School is the home of Troy Tech, Orange County's oldest and most prestigious magnet program. Troy is located in Fullerton, California. Because of the excellence of its academic programs, Troy High School attracts students from four counties. Currently 72 percent of Troy students commute from outside of our attendance area. Students from 120 junior high schools travel to our campus to become a part of one of the most well-known magnet programs in our state. Two special programs established at Troy High School in 1985 -- the Troy Tech high technology-based magnet program for math, science, and computer science and the International Baccalaureate Honors Program -- set the high academic standards for our school. In 1999 Troy was selected once again as a California Distinguished High School. In the year 2000, Troy was further honored as a National Blue Ribbon School of Excellence and was named by the U.S. Department of Education as one of 17 New American High Schools for the year 2000-2001. Over the past five years, the Troy High School Science Olympiad team has captured four first place awards and one second place award in the National Science Olympiad Championship Competition and has won eight consecutive California State Championships.

There are seven 57-minute class periods in a day. The majority of students take all seven class periods. The computer and digital electronics labs are open several times per week at both lunch time and after school.

Grades: 9-12

Type: Comprehensive public high school with special magnet program status for science, math, and computer science

Total Enrollment: 2,200

Percentage of Minorities:

39 percent Asian
14 percent Hispanic
4 percent Filipino
2 percent Black

College Record: 98 percent continue their education at the collegiate level.
65 percent enroll directly in a four-year college/university.
33 percent enroll in two-year colleges/tech schools.

Overview of Advanced Placement Computer Science A

Computer Facilities: Our classroom is also our lab -- we find this to be very conducive to learning. We have our computers around the outside of the room, with the center set up in a traditional classroom fashion. We spend approximately 60 percent of the time in lecture, with the remaining time spent in lab. Our lab and the labs around campus are managed and maintained by a full-time employee. Not enough districts understand the necessity for such a person. He saves us countless hours and ensures that we are up and running nearly 100 percent of the time. This course is on a tight schedule; any downtime during lab is extremely detrimental to student learning. Most students have a computer at home but do work during lunch and after school as well.

The lab is equipped with 40 networked computers. The computers are Pentium 733 MHz systems with 128 MB RAM and a 20 GB HDD. We use JCreator LE as our main IDE. In addition, we use BlueJ, as it is an excellent way to both view the class/abstract class/interface structure of your project and perform unit testing of methods without creating a driver. Both environments are free – and excellent. The use of free tools for students is nice for a variety of reasons.

Student Background and Selection:

All students are required to take one semester of Digital Electronics (logic gates, Boolean algebra, DeMorgan's laws, numbering systems, Karnaugh maps, breadboard prototyping, flip-flops, counters, shift-registers, and adders/subtractors) and one semester of Fundamentals of Programming in C++ during their freshman year as part of the Troy Tech magnet program. All students in those two freshman courses must have had Algebra I and be enrolled in Geometry or beyond as freshmen. The prerequisite for AP CS A is a B+ in each of those two courses; 98 percent of the students in this course are sophomores. Over 30 percent of the students in this course are female. Troy staff members encourage female students in computer science classes beyond the required freshman courses. The entire magnet population and the program itself benefit immensely from this focus.

AP Class Size: There are approximately 160 students in four sections of AP CS A to start off each year. No, that is not a typo.

Texts: Bergin, Joseph, et al. Karel J. Robot: A Gentle Introduction to the Art of Object-Oriented Programming Using Java. Copyright Joseph Bergin.
<<http://csis.pace.edu/~bergin/KarelJava2ed/Karel++JavaEdition.html>>

Brady, Alyce. Marine Biology Simulation Case Study. The College Board, 2000.
<<http://apcentral.collegeboard.com>>

Horstmann, Cay. Computing Concepts with Java Essentials. John Wiley and Sons, 2003.

This book comes complete with a set of tests, PowerPoint slides, and all code examples and labs in electronic form. There are many support materials online. The book covers both the A and AB material. In addition, the AP Study Guide (470 pages by Trees & Horstmann) that you can get with the book is a tremendous source of focused tips, explanations, examples, and multiple choice and free response questions – complete with answers.

<<http://www.horstmann.com/ccj.html>>

Matsuoka, Cary and George Peck. Introduction to Computer Science with Java. Sunnyvale, California: Institute of Computer Technology (ICT), 2003.

ICT has produced materials for AP teachers since the days of Pascal. They have a publication of 40+ lessons that come with well-organized lesson notes for students and teachers, lab assignments, and worksheets. Their materials cover both the A and AB material. It would not, however, be advisable to use the materials as a replacement for having the students read a textbook throughout the course.

<<http://www.ict.org/>>

Syllabus at a Glance

General Topic	Week
<u>Karel J Robot</u>	0-7
<u>Java Basics</u>	8-10
<u>MBS CS Chapter 1</u>	11-12
<u>MBS CS Chapter 2</u>	13-15
<u>Arrays and ArrayList</u>	16-18
<u>Quadratic Sorts and Linear/Binary Search</u>	19-21
<u>Mergesort</u>	22
<u>MBS CS Chapters 3 and 4</u>	23-26
<u>Review</u>	27-29

Course Outline

Weeks 0-7 ([daily schedule](#)): Introduction to the principal concepts in computer science using Karel J. Robot

Objectives:

- Become familiar with the computer lab, accounts, and an IDE
- Understand object-oriented programming and top-down design/refinement of individual tasks
- Basic class structure including instance variables, local variables, parameter passing, scope, public/private visibility, use of super
- Sequence, selection, and iteration
- Recursion
- Inheritance and polymorphism, overriding methods
- `java.util.Random`, `RandomGenerator`
- Analyze, design, code, and test software
- Error categorization/correction

Comments:

I teach computer science concepts so that students have immediate visual feedback -- at least in the beginning. They will truly understand what they have done right and wrong because they can see it.

As you design your program, think about the forest, not the trees. In other words, students should not lose sight of computer science as they examine the details of the computer language. This undertaking should not be too difficult since algorithms that solve a variety of robot tasks are both plentiful and provocative, as are the topics of study associated with them. Let creativity and imagination be your guides. The suggestion may seem a bit idealistic on my part since you may initially be a bit daunted by the task of mastering Java yourself and then constructing simple lessons for your students, but not to worry. Once you get past the inevitable *What-Is-the-Basic-Structure-of-a-Java-Program-and-How-Do-I-Make-It-Run* lesson, you'll thrive. Before you know it, you and your students will be enjoying computer science at the level that it is most inspiring -- the conceptual level. Assisting you will be Karel J. Robot, a Java library (not a program), Professor Bergin et al. (who developed the free and well-written simulator and materials), and other fine folks who have made support materials (see references below).

Weeks 0-7

Comments (cont.): You will find these eight weeks tremendously productive and fun. Many of the topics in AP will be visited during these weeks. The major topics will all come around again in your course; a well-run course will spiral through the major topics, introducing more and more detail as needed about each topic each time you revisit it.

References/Readings:

- Computer Concepts with Java Essentials, chapter 2
- If Karel J. Robot has not yet been published, you'll find the online version (and many other related ideas) at the author's site. You may want a book copy of Karel++ since, as of this writing, not all exercises have been written into the online version.
<<http://csis.pace.edu/~bergin/KarelJava2ed/Karel++JavaEdition.html>>

Assignments/Labs: See the [daily schedule](#), which includes homework assignments, labs, review exercises, PowerPoint presentations, and tests.

Weeks 8-10 ([daily schedule](#)): Java basics

- Objectives:**
- Source, bytecode, compilers, interpreters, Java virtual machine, platform independence
 - Assignment statement, primitive data types
 - Arithmetic operators, ArithmeticException, precedence, casting/promotion
 - java.lang.Math (abs, pow, sqrt), static methods
 - Simple console I/O using some sort of custom library
 - Parameter passing terminology and concepts
 - String class, object references, aliasing
 - Selection in more detail
 - Object is the superclass of all superclasses, overriding toString()
 - Interfaces

Comments: Teaching Java's built-in I/O libraries is not advisable nor is it a part of the subset. Many Java books contain some sort of custom I/O library that will help to keep things simple. See references below for two such libraries.

A nice way to introduce interfaces is to provide one for your students and have them write a couple of classes that implement the interface. In this manner, you are giving their lab/class its basic structure, providing a lab specification (especially if it contains Javadoc), and giving yourself a nice way to automate testing their labs (you are guaranteeing that the students' classes all have the same method signatures, enabling you to write one client program that tests all of their methods; see the Polygon lab below).

- References/Readings:**
- Computing Concepts with Java Essentials, chapter 3
 - Appendix E of Java Methods by Litvin and Litvin contains an easy-to-use I/O library and is available for free online.
<<http://www.skylit.com/jmethods.html>>
 - A set of console and file I/O utilities by Chris Nevison called chn.util.
< <http://cs.colgate.edu/APCS/Java/JavaUtilities/JavaUtilities.htm>>

Assignments/Labs: Change lab, Car Rental lab, Polygon lab ([labs](#))

Weeks 11-12 ([daily schedule](#)): Introduction to the MBS CS and File I/O

Objectives:

- Chapter 1 of MBS CS
- Creating projects and running the MBS CS
- Reading and writing text files
- Black-box testing

Comments:

The Marine Biology Simulation Case Study (MBS CS) can be sliced into byte-sized pieces by incorporating some of the classes as early as possible in the course. In fact, initially, don't even tell the students that the Case Study exists; simply make them comfortable using libraries and objects and writing and designing object-oriented code. As they are mastering these tasks, they are also mastering important AP concepts. This subtle instruction of AP topics is relatively pain-free for the students, who will remain happily oblivious to a chore that they might have otherwise perceived as difficult. For example, they can begin to read and create text files right along with their study of chapter 1 of the Case Study. A lab assignment may prompt the user to enter environment and fish information and then create a text file that can be used as input to the Case Study. If you also require that the lab be fault-tolerant (that is, handle incorrect data entered by the user), you can give them additional practice with selection, iteration, and String and primitive comparisons and conversions (see File Creation lab below).

References/Readings:

- Marine Biology Simulation Case Study (required material for the AP Exam) <<http://www.collegeboard.com/>>
- Alyce Brady, author of the MBS CS (you will find helpful things here related to the teaching of computer science as well as the Case Study) <<http://max.cs.kzoo.edu/AP>>
- Chris Nevison (you will find many helpful things here or a link to where they can be found) <<http://cs.colgate.edu/faculty/nevison>>

Assignments/Labs:

- All analysis questions and exercise sets in chapter 1 of the MBS CS
- MBS CS File Creation lab ([labs](#))

Weeks 13-15 ([daily schedule](#)): MBS CS chapter 2

- Objectives:**
- Intercommunicating objects
 - Inheritance
 - Interfaces (Comparable, Locatable) and Abstract classes
 - Array basics

Comments: There is a lot going on in chapter 2 of the MBS CS. In order for the students (and you) to get a grasp on how the objects communicate with one another, a scripted role-playing exercise has been developed. This is an effective way to enable students to see the big picture without looking at too much code. Seeing and acting out the object responsibilities will help students internalize the complex intercommunication. Be as creative as you like and let everyone have fun with it. Professor Levine shows you how to use role plays. He also suggests how you might use role plays early in your course.

- References/Readings:**
- MBS CS chapter 2
 - David Levine maintains the role plays at <http://web.sbu.edu/cs/dlevine/RolePlay/roleplay.html>.

Assignments/Labs: All analysis questions and exercise sets in chapter 2 of the MBS CS

Weeks 16-18 ([daily schedule](#)): Arrays and ArrayList

- Objectives:**
- Declaring, constructing, initializing, and indexing arrays/ArrayList
 - Storing primitives and objects in arrays/ArrayList
 - Traversing, inserting, deleting array/ArrayList elements
 - Passing arrays/ArrayList to methods
 - Wrapper classes — Double, Integer
 - Casting, ClassCastException, ArrayIndexOutOfBoundsException

Comments: Students took a quick look at arrays in the last section while working with chapter 2 of the MBS CS. Now we go into depth. The first few labs in this section are small and focused, used for practicing simple array traversals, insertions, and deletions. It is best to keep it simple at this point and not embed array concepts within too many object-oriented concepts (see Compact, Statistics, and Mode and Histogram labs below). The Array-Records-Sort-Search lab, however, is a multipart lab with many embedded concepts. Students begin the lab in this section and build on it during both of the following two sections of this syllabus.

References/Readings: [Computing Concepts with Java Essentials](#), chapter 13

Assignments/Labs: Compact, Statistics, Mode and Histogram, Arrays-Records-Sort-Search ([labs](#))

Weeks 19-21 ([daily schedule](#)): Quadratic Sorts and Linear/Binary Searching

Objectives:

- Insertion and selection sorts
- Sequential versus binary searching
- Introduction to some friendly Big-Oh ideas
- Recursion

Comments:

While working with the traditional sorts and searches, we introduce some simple Big-Oh concepts and counting. Big-Oh is not part of the AP CS A Exam, but the counting of statements being executed is a part.

You can have the students count comparisons done while sorting and then graph the results. Explain why comparisons/operations relevant to the dataset size are used as a benchmark as opposed to execution speed. You can put the algorithms that they have studied up to now (e.g., reading data, common array algorithms) into their respective Big-Oh family. This should not be done in a formal mathematical manner (sorry, math folks) at this point; it is intimidating. The students are receiving a friendly, first look at analyzing algorithms; some of the students will appreciate this more-formalized look at efficiency (because they are competing with each other to see whose algorithm is faster!).

This is a good place to work recursion back into the course, since you can explain how the linear and binary searches can be written both iteratively and recursively. In addition, you can cover some of the traditional recursive algorithms (e.g., Fibonacci sequence, multiplication using addition, Towers of Hanoi).

References/Readings: • [Computing Concepts with Java Essentials](#), chapters 17 and 18

- Big-Oh handout ([handouts](#))

Assignments/Labs:

- Sorts Practice, Sorts and Search Practice,
- Recursion Practice (1-3), Counting Iterations, Quadratic Sorts Analysis ([worksheets](#))
- Recursion, Arrays-Records-Sort-Search lab (continued) ([labs](#))

Week 22 ([daily schedule](#)): Mergesort

- Objectives:**
- Mergesort
 - Recursion

Comments: Students will complete the Arrays-Records-Sort-Search lab that they began several sections ago and have been expanding. They will gain additional practice with arrays as they explore the nontrivial task of merging two sorted lists. In addition, students will once again see a comparison between a recursive and nonrecursive solution to an algorithm.

References/Readings: Computing Concepts with Java Essentials, chapter 18

- Assignments/Labs:**
- Mergesort Practice ([worksheets](#))
 - Arrays-Records-Sort-Search lab (continued) ([labs](#))

Weeks 23-26 ([daily schedule](#)): MBS CS chapters 3 and 4

- Objectives:**
- Inheritance and polymorphism
 - Feeling very comfortable with the Case Study

Comments: By this point in the year, students have an excellent knowledge of the Java language and object-oriented principles and can dive (pun intended) into these last chapters and have fun. Culminating the year with the Case Study is best because it will be fresh in their minds while taking the AP Exam.

Require that the students make a few modifications and then allow their creativity to guide the others. In this manner, you will ensure that they learn what you would like and, at the same time, allow them to stay motivated by choosing what they would like to do. This also enables you to get some fantastic ideas for future labs (the students are much more creative than we are). The main idea is to have them working within the many classes and to become extremely comfortable with where things are at and how they work.

References/Readings: MBS CS chapters 3 and 4

Assignments/Labs: All analysis questions and exercise sets in chapters 3 and 4 of the MBS CS

Weeks 27-29 ([daily schedule](#)): Review

- Objectives:**
- Ensure students know what is coming on the AP Exam
 - Earn a 5 on the AP Exam

Comments: These weeks are invaluable and an important factor in our students passing at a 97 percent rate, with nearly 70 percent receiving a 5. They have been answering many AP-type questions on their tests throughout the year, but now they get one after another from previous exams. Students answer one free-response question per night and several over the weekends. We discuss in detail every question during lecture and let them grade each other (we use the AP-type rubrics and teach them the basics of how the Readers score the exams). There are no labs during these weeks because we are accustoming the students to the AP Exam, which is comprised of written and multiple-choice questions only. Students cannot use a computer on the AP Exam, so we do not have them use one while practicing. They will take three full-length, multiple-choice exams during these weeks and have three class periods filled with writing free-response questions (usually one or two in the hour). It is fun for us to watch the students gain more and more confidence over these weeks; they look forward to seeing how well they did and how they did as compared to the students who originally answered the questions during a particular year. We use these exams as their final for the course -- this encourages them to study like mad in concern for their grade (the real motivation), and the positive side effect is that they come out of the AP Exam exclaiming that they did well.

- References/Readings:**
- A collection of all released prior AP Exams, complete with answers, is available through me (written and free-response). All Pascal questions have been converted into C++; others have contributed to the collection. At present, no question has been converted to a Java version.
 - [Be Prepared for the AP Computer Science Exam in Java](#) by Maria Litvin, Skylight Publishing, 2003, ISBN 0-9654853-5-8
 - Grading rubrics at <http://apcentral.collegeboard.com/>

Assignments/Labs: No lab assignments