

AP Computer Science A Syllabus

Course Overview

This course is for college bound students interested in taking a college level computer science course. The course covers the development and analysis of standard algorithms and the development of data structures. Students learn to design code using Object Oriented techniques, implement with an eye toward reusability, document code using javadoc and test code using the JUnit testing frameworks. In addition to using traditional text and reference materials the students are encouraged to use online resources such as the Java API specifications and the sun java tutorials. The class is run as if the students are going to a job, at a small congenial start-up, where everyone works hard. Here the students work collaboratively for the betterment of the group and oneself.

Text and Reference Materials

Horstmann, Cay. Big Java. 3rd ed. Hoboken, NJ: Wiley, 2008

**This is the main textbook*

Trees, Francis. AP CS Study Guide Peoples Publishing Group. Saddle Brook, NJ 07663

Litvin, Maria and Litvin, Gary. Java Methods: An Introduction to Object-Oriented Programming. Andover, MA.: Skylight Publishing, 2001

Litvin, Maria and Litvin, Gary. Java Methods AB: Data Structures. Andover, MA.: Skylight Publishing, 2003

Litvin, Maria. Be Prepared for the AP Computer Science Exam in Java. Andover, MA.: Skylight Publishing, 2003

Institute for Mathematics and Computer Science Online Courses (eIMACS):
<http://www.eimacs.com> (AP approved – This is used as a self-paced tutorial throughout the term and as a reference)

GridWorld Case Study.

http://apcentral.collegeboard.com/apc/members/courses/teachers_corner/151155.html

AP Java Class Assignment Web site

<http://www.vickigold.com/CppAssignments.html>

AP Java Class LessonPlans and Assessment Web site

<http://www.vickigold.com/GradeCppRubric.htm>

Sun Java 5.0 API

<http://java.sun.com/j2se/1.5.0/docs/api/>

Sun Java Big Index

<http://java.sun.com/docs/books/tutorial/reallybigindex.html>

Assignments/Labs

All assignments and labs are posted on the class Web site at

<http://vickigold.com/JavaAssignments.html> . The site includes homework assignments, labs, links to tutorials and other important AP Java links. A companion Web site at <http://vickigold.com/GradeJavaRubric.htm> contains rubrics and daily lesson plans.

Syllabus

All references to text and assessment materials refer to Big Java, Third Edition, Cay Horstmann

Unit 1. Fundamentals about Hardware and Software

This unit briefly introduces the students to computer architecture, networking, system software, and the process of compiling and executing code.

Text and Assessment

Chapter 1 – *Introduction*, Review Exercises and Programming Project 1.1, pg. 31
(compile and run project in Eclipse)

Unit 2. An Introduction to Objects

Students learn how to use objects of existing classes, and develop the skills necessary for implementing classes whose design has been given to them. Homework problems emphasize the use of library classes and the implementation of given designs, by providing students with class outlines to complete or with tester classes that constrain the API.

Text and Assessment

Chapter 2 – *Using Objects*, Review Exercises and Programming Exercises P2.7, pg 76.

The Random class implements a random number generator, which produces sequences of numbers that appear to be random. To generate random integers, you construct an object of the Random class, and then apply the nextInt method... Write a program DieSimulator

that uses the Random class to simulate the cast of die, printing a random number between 1 and 6 every time the program is run.

Programming Exercise P2.8 pg 76 – Write a LotteryPrinter Program.

Chapter 3 – *Implementing Classes*, Review Exercises and Programming Exercise P3.9, pg 127 – *Writing a Bug moving class*

Programming Exercise P3.11 pg 128 *Implement a class RoachPopulation that simulates the growth of a roach population.*

GridWorld Case Study Part 1

Unit 3. Programming Fundamentals

This unit covers fundamental programming concepts: numbers and strings, decisions, and loops. These concepts are presented in the context of classes and methods.

Text and Assessment

Chapter 4 - *Fundamental Data Types*, Review Exercises, pg 171

Chapter 5 – *Decisions*, Review Exercises, p 217

Chapter 6 – *Iterations*, Review Exercises and P6 276

GridWorld Case Study is used to visualize loops

Unit 4. Arrays and an introduction to algorithms

This unit covers arrays, array lists, and array algorithms. The emphasis is on array algorithms and an informal introduction of running time analysis through the discussion of examples with linear and quadratic running time.

Text and Assessment

Chapter 7 – *Arrays and Array Lists*, Review Exercises and Programming Exercises P7.14 and Programming Projects P7.2, pg 332 the Game of Life.

GridWorld case study is used to visualize array algorithms.

Unit 5. An Introduction to Class Design

This unit covers the fundamentals of OO design: identifying classes and methods, coupling, cohesion, programming by contract.

Text and Assessment

Chapter 8 – *Designing Classes*, Review Exercises and Programming Exercises P8.12 and Programming Project P8.2

Chapter 9 – *Interfaces and Polymorphism*, (Omit section 9.) - Review Exercises and Programming Exercises P9.9 and Programming Projects 9.2, The Game of Nim pg 435.

Unit 6. Testing and Debugging

This unit covers program testing in a systematic way. Note that the AP exam will not test sections 5 - 7, but these are important in practice and should not be omitted.

Text and Assessment

Chapter 8 – *Designing Classes*, Section 8.10 Unit Test Frameworks.- Programming Exercises P8.15, Writing a JUnit test case

Unit 7. Interfaces and Inheritance

This unit covers interfaces and inheritance. Interfaces are covered first, thus allowing students to see polymorphism without the syntactical complications of superclass construction and superclass method invocation.

Text and Assessment

Chapter 9 – *Interfaces and Polymorphism*

Chapter 10 – *Inheritance*, Review Exercises and P10.4 and P10.2, Implement the toString, equals and clone methods for all subclasses of a BankAccount class and write unit tests to verify your methods work correctly, pg 495

Chapter 12 – *Object-Oriented Design*, Review Exercises

GridWorld parts 2 - 4

Unit 8. Algorithms

This unit covers recursion, sorting, and searching.

Text and Assessment

Chapter 13 – *Recursion*, Review Exercises and Programming Exercise P13.5 and Programming Project P13.1 with recursion pg 624

Chapter 14 – Sorting and Searching, Review Exercises and P14.7 and Programming Project P14.1, Write a program that keeps an appointment book. (details pg 663)

Unit 9. Social and Ethical Issues in Computer Science

The AP course requires activities related to “Computing in Context”. Random Topic notes are supplemented with recent news stories about electronic voting machine problems, software and music piracy, privacy breaches, viruses and phishing, or spectacular computer system failures. Students produce reports and presentations that show evidence of reflective contemplation. In addition students are responsible on a rotating basis to report on a new technology issues in the news. This is called “Technology of the Day”.

Text and Assessment

Selected "Random Topic" readings, supplemented by readings of current interest.

News articles on all aspects of technologies impact on life today and tomorrow.

Unit 10. IMACS/Grid World Exercises

This unit will extend students’ knowledge of the AP Grid World case study by having students extend existing Grid World classes to do a variety of fun activities. Students will also perform exercises from the Grid World section of the <http://www.eimacs.com> tutorial.

Text and Assessment

GridWorld case study Exercises and programming activities

Unit 11. IMACS

In this unit students’ receive further hands-on practice for the AP exam by completing the eimacs tutorial on Algorithms which covers algorithms (searching and sorting), traversals, replacements, insertions and deletions.

Unit 12. Final Project

Students’ final projects will be a summative project representing both the year’s work and their interests. The project will be an agreed upon between myself and the student.

AP Computer Science Course Description

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.

<i>Computer Science A</i>	Resources – The Book referred to is Big Java by Cay Horstmann
A. Program design	
1. Read and understand a problem description, purpose, and goals.	Chapters 3, 9, and exercises throughout the course
2. Apply data abstraction and encapsulation.	Chapters 2, 3, 9
3. Read and understand class specifications and relationships among the classes (“is-a,” “has-a” relationships).	Chapters 2, 3, 9
4. Understand and implement a given class hierarchy.	Chapters 10, 12
5. Identify reusable components from existing code using classes and class libraries.	Chapters 2, 8, 15, and exercises throughout the course
B. Class design	
1. Design and implement a class.	Chapters 3, 8
3. Choose appropriate data representation and algorithms.	Chapters 8, 15
4. Apply functional decomposition.	Chapters 3, 8, 9
5. Extend a given class using inheritance.	Chapter 10

II. Program Implementation

The overall goals of 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.

<i>Computer Science A</i>	Resources
---------------------------	------------------

A. Implementation techniques	
1. Methodology	
a. Object-oriented development	Chapters 2, 3, 9,12
b. Top-down development	
c. Encapsulation and information hiding	Chapters 2, 3, 9
d. Procedural abstraction	Chapters 3, 9,10,12
B. Programming constructs	
1. Primitive types vs. objects	Chapter 4
2. Declaration	
a. Constant declarations	Chapter 4
b. Variable declarations	Chapter 2
c. Class declarations	Chapter 3
d. Interface declarations	Chapter 9
e. Method declarations	Chapter 3
f. Parameter declarations	Chapter 3
3. Console output (System.out.print/println)	Chapter 2
4. Control	
a. Methods	Chapters 2, 3
b. Sequential	Chapter 6
c. Conditional	Chapter 6
d. Iteration	Chapter 7
e. Recursion	Chapter 13
C. Java library classes (included in the A-level AP Java Subset)	Chapters 2, 4, 8

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.

<i>Computer Science A</i>	Resources
A. Testing	
1. Test classes and libraries in isolation.	Chapters 2, 3, 8
2. Identify boundary cases and generate appropriate test data.	Chapter 5
3. Perform integration testing.	Chapter 8

B. Debugging	
1. Categorize errors: compile-time, run-time, logic.	Chapter 1
2. Identify and correct errors.	Chapter 1 and numerous "Common Error" notes
3. Employ techniques such as using a debugger, adding extra output statements, or hand-tracing code.	Chapter 6
C. Understand and modify existing code	Exercises, GridWorld Case Study
D. Extend existing code using inheritance	Chapter 10, GridWorld Case Study
E. Understand error handling	
1. Understand runtime exceptions.	Chapters 1, 11
F. Reason about programs	
1. Pre- and post-conditions	Chapter 8
2. Assertions	Chapter 8
G. Analysis of algorithms	http://www.eimacs.com
1. Informal comparisons of running times	http://www.eimacs.com
2. Exact calculation of statement execution counts	http://www.eimacs.com
H. Numerical representations and limits	
1. Representations of numbers in different bases	Chapter 4
2. Limitations of finite representations (e.g., integer bounds, imprecision of floating-point representations, and round-off error)	Chapter 4

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.

<i>Computer Science A</i>	Resources
A. Simple data types (int, boolean, double)	Chapter 4
B. Classes	Chapters 2, 3, 9
C. One-dimensional arrays	Chapter 7

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.

<i>Computer Science A</i>	Resources
A. Operations on A-level data structures previously listed	
1. Traversals	Chapter 7
2. Insertions	Chapter 7
3. Deletions	Chapter 7
B. Searching	
1. Sequential	Chapters 7, 14
2. Binary	Chapter 14
C. Sorting	
1. Selection	Chapter 14
2. Insertion	Chapter 14
3. Mergesort	Chapter 14

VI. Computing in Context

A working knowledge of the major hardware and software components of computer systems is necessary for the study of computer science, as is the awareness of the ethical and social implications of computing systems. These topics need not be covered in detail but should be considered throughout the course.

<i>Computer Science A</i>	Resources
A. Major hardware components	
1. Primary and secondary memory	Chapter 1
2. Processors	Chapter 1
3. Peripherals	Chapter 1
B. System software	
1. Language translators/compilers	Chapter 1, "Random Fact" in Chapter 11.1,

	13.1
2. Virtual machines	Chapter 1
3. Operating systems	Chapter 1, "Random Fact" in Chapter 9
C. Types of systems	
1. Single-user systems	Chapter 1, "Random Fact" in Chapter 8
2. Networks	Chapter 1
D. Responsible use of computer systems	
1. System reliability	"Random Fact" in Chapters 4, 8, 11
2. Privacy	A research project based on the Responsible use of computers
3. Legal issues and intellectual property	
4. Social and ethical ramifications of computer use	