



HIGH SCHOOL COURSE OUTLINE

Department	Mathematics/Technology		Course Title	Computer Science A (AP)			
Course Code	3067	Grade Level	11-12	Course Length	2 semesters	Credits/Semester	5
Required for Graduation		Meets H.S. Grad Requirement			Elective Credit		X
Prerequisites	Completion of Geometry with a C or better and concurrent enrollment in Intermediate Algebra or higher						
Meets UC "a-g" Requirement		Yes (g)	Meets NCAA Requirement		No		

COURSE DESCRIPTION:

The content of Computer Science A emphasizes object oriented programming methodology with an emphasis on problem solving and algorithm development. It is the equivalent of a college level semester course in Computer Science. It also covers the study of data structures, design and abstraction. The course emphasizes the design issues that make programs understandable, adaptable and reusable.

GOALS:

The goals of an AP course in computer science are comparable to those in the introductory sequence of courses for computer science majors offered in college and university computer science departments. It is not expected, however, that all students in an AP Computer Science course will major in computer science at the university level. An AP Computer Science course is intended to serve both as an introductory course for computer science majors and as a course for people who will major in other disciplines that require significant involvement with technology. It is not a substitute for the usual college-preparatory mathematics courses. The following goals apply to both of the AP Computer Science courses when interpreted within the context of the specific course. Students should be able to:

- design and implement computer-based solutions to problems in a variety of application areas
- use and implement commonly-used algorithms and data structures
- develop and select appropriate algorithms and data structures to solve problems
- code fluently in an object-oriented paradigm using the programming language Java. Students are expected to be familiar with and be able to use standard Java library classes from the AP Java subset
- read and understand a large program consisting of several classes and interacting objects. Students should be able to read and understand a description of the design and development process leading to such a program. (An example of such a program is the AP Computer Science Case Study.)
- 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
- recognize the ethical and social implications of computer use

CONTENT STANDARDS:

The content standards for this class are based upon the latest version of the AP Computer Science A available at apcentral.collegeboard.com. The two key documents are the course outline and Java AP subset. In the following outline of content and time allotment, these documents are referenced to the most recent AP course outline, May 2007.

PERFORMANCE STANDARDS:

The content standards for this class are based upon the latest version of the AP Computer Science A available at apcentral.collegeboard.com. The two key documents are the course outline and Java AP subset. In the following outline of content and time allotment, these documents are referenced to the most recent AP course outline, May 2007.

OUTLINE OF CONTENT AND SUGGESTED TIME ALLOTMENT:

First Semester-

Week	General Topic	Specific Standards	Instructional Materials	Vocabulary	Assesments
1	INTRODUCTION TO COMPUTER SCIENCE THE JAVA LANGUAGE & JAVA PROGRAM STRUCTURE	VI.B.1-2, IV A, II B 3, II C, additionally Program Spacing and Arithmetic Operators	Java SDK Download, Textbook	Compilation, spacing, indentation, input, output, operators, data types	Algorithmic Programming Assignment-
2	INPUT / OUTPUT ARITHMETIC OPERATORS IF, IF-ELSE STATEMENTS	II B 4 3, II B 4 4,	Textbook- Java Programming Environment	if, if-else, while, compound assignment, increment, decrement	Algorithmic Programming Assignment-
3	CONTROL STATEMENTS WHILE LOOPS	II B 4 4	Textbook- Java Programming Environment	for, do while, logical operators,AND, OR, XOR, NOT, short circuit evaluation, boolean	Algorithmic Programming Assignment-
4	MORE CONTROL STATEMENTS FOR LOOPS INTRO TO METHODS	II B 4 4, II B 2 5, II B 4 1 (Also For each statement)	Textbook- Java Programming Environment	methods, parameters, overloading, passing references,	Multiple Choice and Free Response Test; Algorithmic Programming Assignment-
5	METHODS ARRAYS	II B 2 5, II B 4 1, IV C	Textbook- Java Programming Environment	array, array of objects, arrays of primitives,index	Algorithmic Programming Assignment-
6	METHODS ARRAYS	II B 2 5, II B 4 1, IV C	Textbook- Java Programming Environment	array, array of objects, arrays of primitives,index	Algorithmic Programming Assignment-
7	ARRAYLIST-Iterators	IV- Arraylist data structure	Textbook- Java Programming Environment	Arraylist, iterator	Algorithmic Programming Assignment-
8	ARRAYLIST-Iterators	IV- Arraylist data structure	Textbook- Java Programming Environment	Arraylist, iterator	Multiple Choice and Free Response Test; Algorithmic Programming Assignment-

Week	General Topic	Specific Standards	Instructional Materials	Vocabulary	Assesments
9	OBJECT CREATION	I: 3-A, 5-A: II: 1-A, 2-B,	TEXTBOOK, PROGRAMMING ENVIRONMENT	Class Diagram Collaboration Diagram Has-a Relationship Is-a Relationship Knows-a Relationship Unified Modeling Language (UML)	Design Programming Assignment
10	OBJECT INTERACTION	I: 3-A, 5-A, 1-B, 2-B, 3-B	TEXTBOOK, PROGRAMMING ENVIRONMENT		Design Programming Assignment
11	OBJECT INTERACTION	I: 3-A, 5-A, 1-B, 2-B, 3-B	TEXTBOOK, PROGRAMMING ENVIRONMENT		Design Programming Assignment
12	INHERITANCE POLYMORPHISM INTERFACES ABSTRACT CLASSES	I:3-B: II:A-1	TEXTBOOK, PROGRAMMING ENVIRONMENT	INHERITANCE, SUPER CLASS, SUBCLASS, BASE CLASS, CLASS HIERARCHY, EXTENDS, OBJECT CLASS, INTERFACES	Design Programming Assignment
13	INHERITANCE POLYMORPHISM INTERFACES ABSTRACT CLASSES	I:3-B: II:A-1	TEXTBOOK, PROGRAMMING ENVIRONMENT	POLYMORPHISM, ABSTRACT CLASSES, INTERFACE DECLARATION, INSTANCE OF OPERATOR, STATIUC BINDING, ITERATOR CLASS	Design Programming Assignment
14	INHERITANCEPOLYM ORPHISMINTERFACE SABSTRACT CLASSES	I:3-B: II:A-1	TEXTBOOK, PROGRAMMING ENVIRONMENT	ABSTRCTBUTTON CLASS, ADAPTER CLASS, BOX CLASS,EVENT, EVENT DRIVEN, EVENT HANDLER, FOCUSJLIST CLASS, NESTED CLASS, ROLLOVER ICON,SWING GUI COMPONENTS,	Design Programming Assignment
15	INHERITANCE POLYMORPHISM INTERFACES ABSTRACT CLASSES	I:3-B: II:A-1	TEXTBOOK, PROGRAMMING ENVIRONMENT	CREATEGRAPHIC S METHOD, DRAW METHOD, GRAPHICS CLASS, COORDINATES	Multiple Choice and Free response test; Design Programming Assignment

Second Semester

General Topic	standard	content vocabulary	assessments
STRINGS EXCEPTION HANDLING	III.E.1, III.E.2, IV.A	characters, throw, exception, finally, catch, try	Design Programming Project
SEARCHING/ SORTING RECURSION / BIG (O)	II.B.4.5, III.G.1, III.G.2, V.B.1, V.B.2	permutations, recursion, linear search, binary	Algorithmic Programming Project

General Topic	standard	content vocabulary	assessments
		search, big (O)	
SEARCHING/SORTING RECURSION / BIG (O)	II.B.4.5, III.G.1, III.G.2, V.C.1, V.C.2, V.C.3	algorithm analysis, selection sort, insertion sort, merge sort, quick sort	Algorithmic Programming Project
MARINE BIOLOGY SIMULATION			Design Programming Project
MARINE BIOLOGY SIMULATION			Design Programming Project
MARINE BIOLOGY SIMULATION			Design Programming Project
MARINE BIOLOGY SIMULATION			Design Programming Project

APPLICATION OF COURSE CONTENT:

Career connection-

Students will research a variety of career options related to computer science. Students will research educational opportunities both directly and indirectly related to computer science. Guest speakers from industry will present to classes about a careers in computer science. Job Shadowing and internships will be made available to students.

Related Career Titles-

Service Learning- As an option towards satisfying LBUSD requirements for high school graduation, students will share their knowledge with high school freshman and middle school students.

METHODS: A variety of instructional strategies will be utilized to accommodate all learning styles:

1. Teacher Directed Methods
 - Direct instruction (lecture) should provide all components of EEEI lesson design:
 - Direct demonstration of programming code.
2. Peer Collaborative Methods
 - Peer to peer problem solving: students will work with partners to solve introductory programming problems.
 - Programming project will be developed in groups of 2 or 3.
3. Literacy Strategies
 - Cued response reading- after completing a chapter reading, students will pair up with a partner and chart each other's understanding of the reading.
 - Groups of students will present summaries to each other of chapter sections and case study materials.
 - Role play- students will physically model abstract concepts (arrays, case study, object creation)
4. Student Directed Methods
 - Active Participation- students will modify an existing code segment or program during class time.
 - Directed research opportunities using as a possible resource, libraries, internet access, and computer manuals.
 - Student selected programming projects.
5. Student Differentiation
 - Use informal assessment to gauge student's understanding
 - Provide additional assistance and instruction for students who require it.

- Provide options for students to select the programming project goals and specifications

MATERIALS USED IN TEACHING THE COURSE

Recommended Textbooks: Java Software Solutions for AP Computer Science A, 2nd Edition; Lewis, Loftus and Cocking, 2007; Pearson Education (Addison- Wesley)
ISBN 0-13-222251-5

Required Software:

- Java 5.0- J2EE- Free download from Sun
- Java IDE- Several free versions available, BlueJ, JCreator

Supplemental Materials:

- Alice 2.0- 3D animation environment to teach object oriented programming fundamentals.
- Karel J. Robot - Java version of Karel the Robot
- Lego Mindstorms Kit and the leJOS programming system

EVALUATION: Student achievement in this course will be measured using multiple assessment tools including but not limited to: (a grading scale and/or rubric should be included)

There are three types of programming assessments: Algorithmic, Design, Project.

1. Algorithmic Programming
 - a. The student needs to solve an algorithmic problem and translate it into a program.
 - b. The student needs to design a set of test cases that will test the algorithm that they designed
2. Design Programming
 - a. The student needs to understand the relationship between various software components
 - b. The student needs to apply changes to the correct software component.
 - c. The student needs to design a new software component which will interact with the existing software architecture
 - d. The student writes a design document which has the following sections: purpose, design, specification, and testing. It will be two pages in length.
3. Project Programming
 - a. The student selects a game to program.
 - b. The student specifies the rules of the game
 - c. The student designs a set of software components that will implement the rules of the game
 - d. The student implements a working computer game that will be demonstrated during class.
 - e. The student writes a report that has the following sections: description of the game, specification of the game rules, design of the software and testing of the software. The report will be five to ten pages in length.

There will be three tests each semester. Each test will include a multiple-choice and free response section in order to emulate the actual College Board AP test.

Weighted Percentage Scale

Category	Weighted Percentage
Algorithmic Programming	20%
Design Programming	20%
Project Programming	30%
Tests	30%

Standard Grading Scale

A = 90% - 100%

B = 80% - 89%

C = 70% - 79%

D = 60% - 69%

F = Below 60%

Submitted by: Vanitha Chandrasekhar

School/Office: OCIPD/Technology

Original Date: _____

Revised Date: _____ 9/25/07

Hscourse/business/technology/ComputerScienceAAP