

AP COMPUTER SCIENCE **AB** SYLLABUS

Overview of AP® Computer Science AB:

It is the goal of this course to prepare incoming freshman at the college level for beginning programming classes by teaching and following all of the topics listed in the "Computer Science AB" column of the Topic Outline in the AP Computer Science Course Description. Also by teaching students to design and implement computer-based solutions to problems in a variety of application areas and teaching students to use and implement commonly used algorithms and data structures. The labs that are assigned are meant to teach students to develop and select appropriate algorithms and data structures to solve problems. 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 Appendixes A and C of the AP Computer Science Course Description. The course teaches students to read and understand a large program consisting of several classes (workspace) and interacting objects, and enables students to read and understand the current AP Computer Science Case Study, GridWorld, posted on AP Central. 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.

Major Texts:

Deitel, H. M., and P. J. Deitel. Java: How to Program. 5th ed. Upper Saddle River, N.J.: Prentice Hall, 2003.

College Board. AP GridWorld Case Study. New York: College Entrance Examination Board, 2006.

Horstmann, Cay. Big Java. Hoboken, N.J.: Wiley, 2002.

Lambert, Ken, and Martin Osborne. Fundamentals of Java, Comprehensive Course. 2nd ed. Boston: Course Technology, 2002.

Schram, Leon. Exposure Java, John L. Schram, Dallas, TX , 2007.

Steps in Learning Computer Science:

- Closely listen to the lecture
- Pay close attention to program examples
- Play with the computer. If you don't understand a program example, type it in and try it.
- Review the material again after the lesson. Come back to class with questions ready.
- Read the lab assignments before working on them. Prepare an algorithm in advance.
- Work hard on the labs. Come before school, after school, or during lunch if you need extra time on the computer.
- Do not wait until the last minute. Clarify areas of confusion before you take the test.
- Ask questions.

Grading:

Tests: 40%
Lab Assignments/daily/homework/quizzes: 60%

Pre-Requisite:

Computer Science A AP

Environment:

All students are supplied with a desktop computer with JCreator LE and SDK

1.5 installed. There is an online version of the book linked to my website, here you will also find PowerPoint presentations, keys to exercises and quizzes, and all files needed to do the lab assignment. This is done so that anything that can be done at school can also be done at home.

Lab Assignments:

Every Unit has labs that are assigned and designed to help the student practice what is being taught. There are usually 2 labs per unit where the first is practice where they receive instruction and the second is done for a grade and is graded in class from the computer. All coding can be viewed at all times.

where they receive instruction and the second is done for a grade and is graded in class from the computer. All coding can be viewed at all times.

Once every 9 weeks the students are given 7-9 short programming assignments to be completed and graded in one 50 minute class period. They range in difficulty from correcting programs with bugs to writing an entirely new class. This is a test grade.

AP Computer Science AB First Semester

Unit Main Topics, Sub Topics and Lab Assignments Objective Cycle/Week
XXIII

Review of Java Syntax

Start of school house keeping items

Primitive data types

Decision control structures

Loop control structures

Program input/output

Lab Assignment

Lab 23, "The Perfect Number Program"

Administer Chapter 23 multiple choice test

Go over Chapter 23 test

XXIV

Focus on OOP, Encapsulation

OOP Terminology

Class declaration

Instantiation and construction

Get methods

Private and public class members

Overloaded constructors

Accessing multiple files

Set methods

Copy constructors

Objects are references
Scope of an object
Using the "this" reference
Static attributes and static methods

C9
C8
C1, W1
2 days
C1, W1
2 days
C1, W1-2
1-1 days
C3
C4
C5
C7 C1, W2
2 days

Review of AWT graphics Review of AWT graphics
Lab Assignments

1 day
Lab24a, "The Polygon Program" C1, W2-3
1-1 days
Lab24b, "The Circle Class Program" C1, W3
3 days
Administer Chapter 24 multiple choice test C1, W3-4
Go over Chapter 24 test 1-1 days
Unit Main Topics, Sub Topics and Lab Assignments Objective Cycle/Week
C3
XXV Focus on OOP, Inheritance & Composition C4
C5
Inheritance syntax C7 C1, W4
Passing information to a super class
Overriding super class methods
Public, private and protected class members
Composition
Multiple inheritance
2 days
Lab Assignment C1, W4
Lab25, "The Inheritance Program" 2 days
Administer Chapter 25 multiple choice test C1, W5
Go over Chapter 25 test 2 days
C3
XXVI Static Arrays and ArrayList C4
C5

Data structures C7 C1, W5

Array definition

Static Java arrays

Static one-dimensional arrays

Constructing Java arrays

The <length> field

Static two-dimensional arrays

Using <length> with two-dimensional arrays

Ragged arrays

The <List> case study

Review array algorithms

3 days

Dynamic Storage with <ArrayList> C1, W6

Methods <add> and <get>

Method <set>

Method <remove>

Methods <add (overloaded)>, <clear> and <isEmpty>

Storing primitive data in <ArrayList> objects

Methods <add> and <get>

Method <set>

Method <remove>

Methods <add (overloaded)>, <clear> and <isEmpty>

Storing primitive data in <ArrayList> objects

Administer Programming Lab Test #1 C1, W6

1 day

Unit Main Topics, Sub Topics and Lab Assignments Objective Cycle/Week

Catch-up days C1, W6

2 days

XXVI Static Arrays and ArrayList Continued

Lab Assignments

Lab26a, "The dynamic matrix program"

Lab26b, "The knight's tour program"

Administer Chapter 26 multiple choice test

Go over Chapter 26 test

C3

C4

C5

C7 C2, W1

3 days

C2, W1-2

2-2 days

C2, W2

2 days

XXVII Focus on OOP, Redefining Existing Methods

Redefining the <toString> method

The <TimeTest> class
 Redefining the <equals> method
 Implementing the <compareTo> method
 A close look at parameter passing
 Lab Assignment
 Lab27, "The Geometry Program"
 Administer Chapter 27 multiple choice test
 Go over Chapter 27 test
 C3
 C4
 C5
 C7 C2, W2-3
 1-1 days
 C2, W3
 2 days
 C2, W3
 2 days
 XXVIII Input/Output with Sequential Files
 C3
 C4
 C5 C5
 C7 C2, W4
 Using the <File> class
 Files of character strings
 Files of numbers
 Keyboard input revisited
 2 days
 Lab Assignment C2, W4
 Lab28, "The Student Records Program" 2 days
 Administer Chapter 28 multiple choice test C2, W4-5
 Go over Chapter 28 test 1-1 days
 Unit Main Topics, Sub Topics and Lab Assignments Objective Cycle/Week
 C3
 XXIX The Stack Abstract Data Structure C4
 C5
 Data vocabulary C7 C2, W5
 Information hiding
 Stacks at the abstract level
 Using the <MyStack> class
 Processing data with a stack class
 Using the <MyStack> class with different data
 Implementing the <MyStack> class
 2 days
 Lab Assignments C2, W5-6
 Lab29a, "The Random Maze Program" 2-2 days
 Administer Programming Lab Test #2

C2, W6

1 day

Catch-up days

C2, W6

2 days

C3

XXIX The Stack Abstract Data Structure Continued C4

C5

Lab Assignments C7 C3, W1

Lab29b, "The Stack Implementation Program" 2 days

Administer Chapter 29 multiple choice test C3, W1

Go over Chapter 29 test 2 days

C3

XXX The Queue Abstract Data Structure C4

C5

Queues at the abstract level C7 C3, W1-2

Using the <MyQueue> class

Processing data with a queue class

1-1 days

Using <MyQueue> with different data

Implementing a queue class

Lab Assignments

Lab30a, "The Screen Saver Program"

Lab30b, "The Queue Implementation Program"

Administer Chapter 30 multiple choice test

Go over Chapter 30 test

Using <MyQueue> with different data

Implementing a queue class

Lab Assignments

Lab30a, "The Screen Saver Program"

Lab30b, "The Queue Implementation Program"

Administer Chapter 30 multiple choice test

Go over Chapter 30 test

Unit Main Topics, Sub Topics and Lab Assignments Objective Cycle/Week

XXXI Collections, Interfaces, Abstract Classes and Polymorphism

What is an interface?

The <Collection> hierarchy

Java collection classes

Linked lists, hashing and trees

Interfaces and implementations

Abstract classes and implementations

Iterators

Polymorphism

Lab Assignment

Lab31, "The Polymorphic List"

Administer Chapter 31 multiple choice test

Go over Chapter 31 test

C3

C4

C5

C7 C3, W3-4

2-1 days

C3, W4

3 days

C3, W4-5

1-1 days

Administer Programming Lab Test #3 C3, W5

1 day

Semester Exam Review and Catch-up Days C3, W5

3 days

Semester I Examination C3, W6

Start Second Semester

Unit Main Topics, Sub Topics and Lab Assignments Objective Cycle/Week

XXXII Recursion II

Pre-recursion assignment

Recursion fundamentals

Recursive methods

Fibonacci, a recursive nono

Evaluating recursive methods

Manipulating parameters of recursive methods

The tower of hanoi

Why recursion

The recursive bubble sort

The Merge Sort casestudy

Mutual recursion

Lab Assignments

Lab32a, "The Disappearing Squares Program"

Lab32b, "The Square Fractal Program"

Administer Chapter 32 multiple choice test

Go over Chapter 32 test

C3

C4

C5

C7

C4, W1

4 days

C4, W1-2

1-1 days

C4, W2

4 days

C4, W3

2 days

XXXIII Linked Lists I

Review of the <LinkedList> class

Multiple queue implementations

Pre-OOP <ListNode> class

C3

C4

C5

C4, W3-4

3-1 days

OOP <ListNode> class
Loops and linked lists
Considering cases
An ordered linked list
Linked lists and memory
The <GfxNode> class
Lab Assignments
Lab33a, "The Graphical Linked List Program"
Lab33b, "The Polynomial Program"
Administer Chapter 33 multiple choice test
Go over Chapter 33 test
OOP <ListNode> class
Loops and linked lists
Considering cases
An ordered linked list
Linked lists and memory
The <GfxNode> class
Lab Assignments
Lab33a, "The Graphical Linked List Program"
Lab33b, "The Polynomial Program"
Administer Chapter 33 multiple choice test
Go over Chapter 33 test

Unit Main Topics, Sub Topics and Lab Assignments Objective Cycle/Week

XXXIV Linked Lists II

Dereferencing <null>

The <MyStack> class

Queues and circular linked lists

Doubly linked lists

Linked list of linked lists

Practice exercises

C3

C4

C5

C7 C4, W5

3 days

C4, W3-4

1-3 days

C4, W4

2 days

Administer Programming Lab Test #4 C4, W6

1 day

Lab Assignment

Lab 34, "The Sparse Matrix Program"

Catch-up day

Administer Chapter 34 multiple choice test

Go over Chapter 34 test

C3

C4

C5

C7

C4, W6

3 days

C4, W6

1 day

C5, W1

2 days

XXXV Binary Trees

Binary tree vocabulary Binary tree vocabulary

The binary search tree 3 days

Declaring a binary tree node

Traversal methods

Creating a binary search tree

Deleting nodes from a binary tree

Different types of binary trees

Using an auxiliary methods

Binary tree methods

Traversing a binary tree by levels

The tree statistics program

Unit Main Topics, Sub Topics and Lab Assignments Objective Cycle/Week

Lab Assignment C5, W2

Lab 35 "The Student Tree Program I" 3 days

Administer Chapter 35 multiple choice test C5, W2

Go over Chapter 35 test 2 days

C3

XXXVI Algorithmic Analysis C4

C5

Measuring execution efficiency C7 C5, W3

Measuring simple algorithms

Linear search and binary search

Adding a node to a linked list

Measuring nested loop algorithms

Measuring the merge sort

Algorithms with drastic behavior

Measuring algorithmic consistency

Algorithmic analysis methods

What is Big-O?

Computing Big-O methods

Best, Worst and Average Case scenarios

Lab Assignment

3 days

There is no lab assignment for Chapter 36 C5, W3

Administer Chapter 36 multiple choice test

Go over Chapter 36 test

2 days

C7

XXXVII XXXVII

C5, W4

3 days

C5, W4

2 days

XXXVIII Sets and Maps

Set implementations <HashSet> and <TreeSet>

<Set> methods <add>, <contains>, <remove> and <size>

<Iterator> methods <next>, <hasNext> and <remove>

User-defined methods <intersection>, <union> and <difference>

<Map> implementations <HashMap> and <TreeMap>

<Map> methods <put>, <get>, <remove>, <containsKey>, <size> and <keySet>

Lab Assignment

Lab 38, "The Dictionary Program"

Administer Chapter 38 multiple choice test

Go over Chapter 38 test

C3

C4

C5

C7 C5, W5

3 days

C5, W5

2 days

C5, W6

2 days

Administer Programming Lab Test #5 C5, W6

1 day

Catch-up Days C5, W6

2 days

GridWorld Case Study and AP Examination Review

C7

C6, W1-3

15 days

AP Computer Science Examination

XXII XXII C6, W4-6

15 days

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.

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 C 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® .

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