Oak Park
HIGH SCHOOL

AP Computer Science A Course Syllabus

This course allows all students to explore programming through problem solving. Students learn to design and implement computer-based solutions to problems in a variety of application areas. In solving these problems, students develop, select, and implement appropriate algorithms and data structures. The object-oriented paradigm is used to problem solve using the Java programming language. Students read, interpret, and instantiate standard Java library classes from the AP Java subset. Reading, understanding, and appropriately modifying large programs like the Chatbot and Numbers Case Studies that consist of several classes and interacting objects are also requirements of this course. While the *Blue Pelican Java* (Charles E. Cook) text is the main resource for students, several other resources both internet-based and text based will be utilized. Examples are CodingBat.com and *5 steps to a 5 for APCS A (*McGraw Hill Education*)*.   Topics are revisited throughout the course in a more formal way to reinforce key concepts and deepen student understanding. Discussion of social and ethical implications of programming and computer use will also be covered.

Text and Supplemental Materials

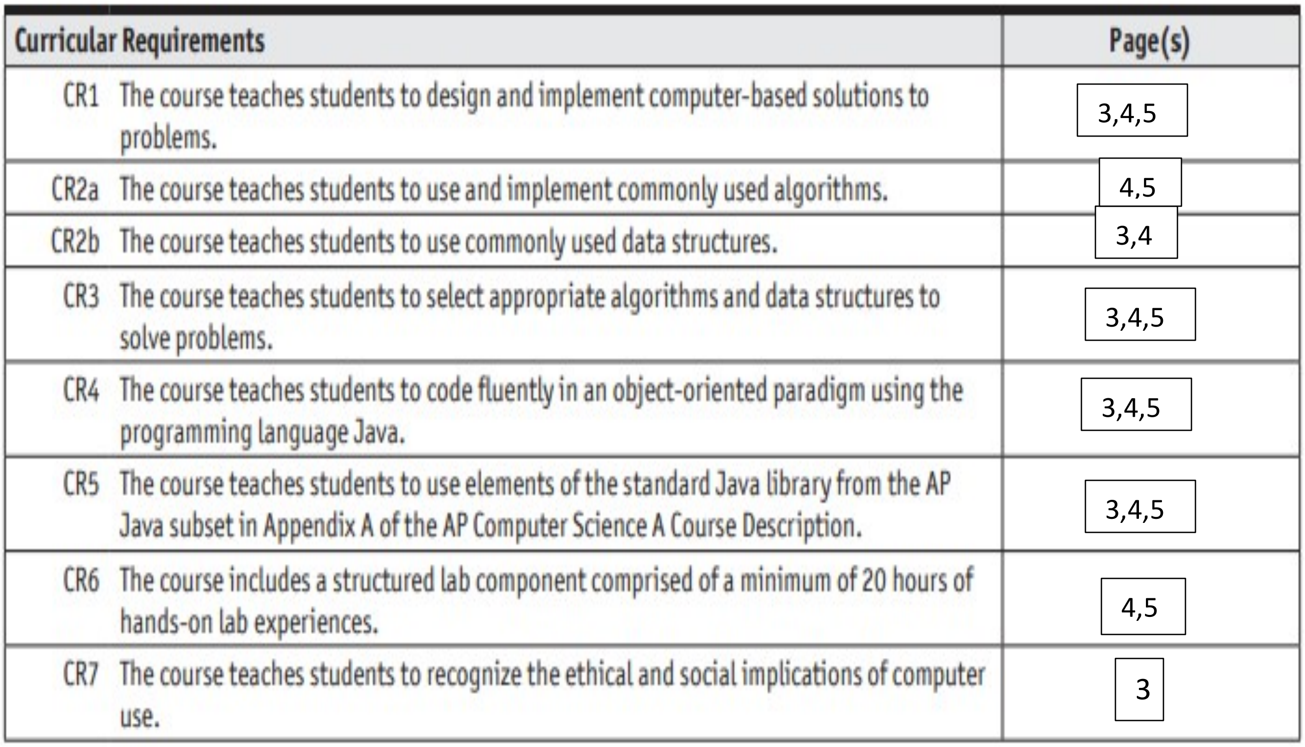
Charles E. Cook. *Blue Pelican Java*. Texas: Virtualbookworm.com Publishing., 2013.
Johnson, Paymer, Chamberlain. *5 Steps to a 5 AP Computer Science.* McGraw Hill Ed*. 2018*
Chatbot and Numbers Case Studies. The College Board website.
Code.org   Programming videos http://www.code.org
CodingBat.com. Nick Parlant   https://codingbat.com/java
The APCS A Labs, The College Board website.
Online FRQ practice exam problems:
https://apstudent.collegeboard.org/apcourse/ap-computer-science-a/exam-practice
http://practiceit.cs.washington.edu/index.jsp

Teaching Strategies

The Socratic method is used regularly to allow students to share their insights, expand their understanding, and explore new related ideas in the context of their current work. The students have one-one laptops with BlueJ and Eclipse IDEs installed. Mindfulness, thinking routines and Reflections are used daily as well as pair programming except on assessments. Weekly reading assignments from the course resources are used to provide students with initial insights that are refined through probing questioning and discussion.

Role-play is used throughout the course to allow students to visualize and act out important algorithms. Computer animations are also used from various website to reinforce student understanding. Early in the course, relatively short programming assignments are given on a weekly basis to allow students to apply their knowledge to code creation working individually and in pairs. CodingBat.com is used to allow students to improve their algorithmic problem-solving skills. As the course progresses, students are required to work on group programming projects that include design exercises and code sharing. With lab projects, students create relatively large programs that illustrate their ability to apply the concepts learned in class.

Curricular Requirements



Curricular Requirements		Page(s)
CR1	The course teaches students to design and implement computer-based solutions to problems.	3,4,5
CR2a	The course teaches students to use and implement commonly used algorithms.	4,5
CR2b	The course teaches students to use commonly used data structures.	3,4
CR3	The course teaches students to select appropriate algorithms and data structures to solve problems.	3,4,5
CR4	The course teaches students to code fluently in an object-oriented paradigm using the programming language Java.	3,4,5
CR5	The course teaches students to use elements of the standard Java library from the AP Java subset in Appendix A of the AP Computer Science A Course Description.	3,4,5
CR6	The course includes a structured lab component comprised of a minimum of 20 hours of hands-on lab experiences.	4,5
CR7	The course teaches students to recognize the ethical and social implications of computer use.	3

Couse Outline

INTRODUCTION

Students are introduced to the lab environment we will use for java coding (one-one laptops, share drives, google classroom and Piazza). They receive links to tutorials on BlueJ and the Eclipse IDE. The “Acceptable Use of School Computer Policy” is reviewed and signed. The students are provided the link to the College Board’s “Computer Science A Course Description.”

The first lesson of the course is the ethical and social implications of computer use. The following article: Brey, P. (2007). ‘Computer Ethics in (Higher) Education, is read and discussed in class. https://ethicsandtechnology.eu/wp-content/uploads/downloadable-content/Brey_2007_Higher-Education.pdf

Then, students collaboratively share what social responsibilities they have when using technology devices on the shared class blog. These responsibilities are modeled throughout the course. **[CR7]**

UNIT 1 - Fundamentals of Programming

Primitive Variables, Math Functions, relational operators, logical operators, conditional statements, looping statements, comments, and errors.

Assignments/ Projects: [CR1, CR4, CR5]
Blue Pelican Java: Lesson 3 String Operations. Project: “Name That Celebrity”
Blue Pelican Java: Lesson 4 Using Numeric variables. Project: “Cheating on Your Arithmetic Assignment”
Blue Pelican Java: Lesson 6 Math Class Methods. Project: “Compute This”
Blue Pelican Java: Lesson 8 Boolean Type, operators and the if statement. Project: “Even or Odd”
Blue Pelican Java: Lesson 9 the IF and SWITCH statement. Project: “Weight on Other Planets”
Blue Pelican Java: Lesson 11 the FOR loop. Project: “Name Reversal”

UNIT 2 - Classes and Objects and the String Class

Relationship between classes and objects, class declaration, instance variables, constructors, methods, parameters, overloaded constructors and methods, static variables and methods, data encapsulation, String variables, string objects, string concatenation, comparing strings, string methods

Assignments/ Projects: [CR1, CR2b, CR4, CR5]
Reading: 5 Steps study guide: chapter 2 pg. 118-128
Activity: Creating the Car Class (“is a” vs. “has a”)
Blue Pelican Java: Lesson 15 public, void, state variables, constructors. Project: “Overdrawn at the Bank”
Blue Pelican Java: Lesson 16 Private methods, state variables, equality of objects. Project:”Gas Mileage”
Blue Pelican Java: Lesson 17 string methods-compareTo(), indexOf(),replace(). Project: “Add em Up”

UNIT 3 - Data Structures

Arrays, 2D arrays, the ArrayList, Array and ArrayList methods

Assignments/ Projects: [CR1,CR2b,CR4,CR5,CR6]
Blue Pelican Java: Lesson 18 Declare, Init, length, passing arrays Project: “Array of Hope”
Blue Pelican Java: Lesson 19 Arrays Class, methods: arrayCopy, toCharArray Project:”Two Orders for The Price of One”
Blue Pelican Java: Lesson 19 sort, binarySearch, equals, fill, import Project:”Sorting a String Array”
LAB: Magpie Chatbot Lab **[CR6]**

UNIT 4 – Algorithms

Definition and samples, how to write an algorithm, commonly used algorithms in java coding such as swap, copy, sequential search, accumulate, and find highest.

Assignments/ Projects: [CR1,CR2a,CR3,CR4,CR5]
Activity: Algorithm vs. pseudo code vs. java code
Project: Write Swap algorithm, pseudo code and java code for an Array
Project: Write Copy algorithm for pseudo code and java code for Array or ArrayList
Project: Write Sequential algorithm, pseudo code and java code for Array
Project: Write Accumulate algorithm, pseudo code and java code for Array or ArrayList
Project: Write Find Highest algorithm, pseudo code and java code for Array

UNIT 5 – Inheritance, Polymorphism and Interfaces

Superclasses and subclasses, the Object class, how to override classes and methods, instanceof, this and super usages, abstract and final usages, Interfaces

Assignments/ Projects: [CR1, CR3, CR4, CR5, CR6]
Reading: 5 Steps Chapter 8 pg. 238-249
Video: BluePelican Lesson35-1.mp4, Lesson35-2.mp4, Lesson35-3.mp4, Lesson35-4.mp4
Blue Pelican Java: Lesson 35 superclass. Subclass. instanceOf, extends, implements Project: “Linear Functions”
LAB: Elevens Lab with Activities and Enrichments **[CR6]**

Unit 6 – Recursion
Recursion vs. looping, examples of factorial and Fibonacci recursive methods

Assignments/ Projects: [CR1,CR2a,CR3,CR4,CR5,CR6]
Reading: 5 Steps Chapter 10: Recursion pg. 279-286
Video: Java Walk--thru: Fibonacci https://www.youtube.com/watch?v=m6tOnX4VbZM
Video: Java Factorial with for Loop https://www.youtube.com/watch?v=bSnUC1lPMB0
Video: Factorial with Recursion https://www.youtube.com/watch?v=wF2CdeRmhUg&t=3s

LAB: RECURSION Created by Morgan McGuire from Williams College **[CR6]**
http://cs.williams.edu/~morgan/cs136-s08/handouts/07/recursion.pdf

UNIT 7 – Sorting Algorithms

Binary search, Insertion sort, selection sort, merge sort

Assignments/ Projects: [CR1,CR2a,CR3,CR4,CR5,CR6]
Reading: 5 steps chapter 12 pg. 313-321
Activity**:** Acting out search algorithms
Project: write “Guess my Number” using Binary search
Project: Sort a list of numbers using the selection
Project: Sort a list of numbers using the merge sort
LAB: Write a program that allows the user to enter a list of words and then displays the list of words in alphabetical order. Use Print and Scanner to get the list of words from the user. Add each word to an ArrayList. Choose to sort the ArrayList using either BubbleSort, InsertionSort, or SelectionSort. **[CR6]**

UNIT 8 – Review

Exam Practice Exams from *5 Steps to a 5 AP Computer Science.* McGraw Hill Ed*. 2018*

FRQs from https://apstudent.collegeboard.org/apcourse/ap-computer-science-a/exam-practice