2014-2015 Syllabus for AP Computer Science

October 12, 2014

1 Introduction

I love Computer Science and think that it's valuable to the students (whether they know it or not) for several reasons:

- students will increase their ability to think abstractly and solve problems;
- students will have exposure to an ever-growing field of work;
- students will learn skills and concepts in this class that are transferable to other programming languages; and
- students will learn the basics about how to automate tasks for computers.

Because most of the students in this class to go on to work in a job that involves computers at least peripherally, I expect this class to be one of the most useful many of them will see.

For me, I often look at the class as a series of puzzles and I think the students tend to think the same way. Each one is a challenge and they generally feel pretty proud of themselves when they finish their programs.

In this class I expect to show the students basic examples of Python and the language used in "Context Free Art" (http://www.contextfreeart.org/) later on in the year, but we will be focusing on object oriented programming in Java over 90% of the year. In particular, we will focus on the subset of Java laid out in the AP Computer Science Course Description and Appendix A thereof.

The students will engage in well-over 20 hours worth of labs throughout the class including tasks implementing each new topic we learn about. Indeed, most of the class will be spent with students taking an active part in their learning by designing, coding, testing, and revising programs that they will use to learn and implement each of the topics in the class. Each of the students in the class has a laptop to use during class time and they will be used almost every day. In addition to the numerous small labs particular to this class, we will use the Magpie, PictureLab, and Elevens labs provided from the College Board. We will use the BlueJ development environment for this class,[1] but may switch to JCreator later in the year.

I expect all students in the class to be prepared for and take the AP Computer Science Exam. We have a unit at the end of the class focused exclusively on studying for the test.

This syllabus[2] provides an outline the course.

2 Semester One Course Outline

Introductory Material (Weeks 0-1)

Topics:

- What is computer Science?
- Algorithm definition and basic examples (recipes, etc)

[1] with a short interlude to `<https://koding.com/>` during our command line phase

[2] Based largely on example syllabus 2: ID# 1172786v1 from the College Board website

- What is a Programming Language?
- Environment Setup
- Using a compiler
- Hello World! / Basic console output (`System.out.println` and `System.out.print`)
- Number Bases

Labs / Assignments:

- Task: Write an algorithm for your morning routine
- Lab: Hello World! Program & Setting up the BlueJ development Environment
- Worksheet: Converting between Binary, Decimal, and Hexadecimal

Primitives and Objects (Weeks 2-3)

Topics:

- Variables as storage
- Primitives (`int`, `double`, `char`, `boolean`)
- Assignment vs expressions
- Operators (`+, -, *, /, %, +=, -=, *=, /=, %=, ++, --`)
- Declaring Variables
- Differences between primitives and objects
- Java's `Math` class (`pow, sqrt, abs, ceil, random`)
- Creating Random Numbers
- Java's `Scanner` class for user input (`nextInt, nextLine`)
- `String` basics: quotations, unicode, escape sequences (`\n, \t`, etc)
- String methods: (length, substring, indexOf, charAt)
- Integer, Double, Boolean, Character wrapper classes

Labs / Assignments:

- Lab: Base conversion - converts a four digit decimal number to binary
- Lab: Calculating the distance and slope of a segment given two endpoints
- Lab: Printing out a table of values
- Lab: Reading back a survey to the user
- Lab: Find the distribution of letters in a string.
- Worksheet: Mathematical Expressions
- Worksheet: Variable Assignment
- Worksheet: Creating random numbers

Conditionals (Weeks 4-5)

Topics:

- `If` statements
- `If / else` statements
- Nested `If` Statements
- Boolean expressions
- More Operators (`==, !=, >, <, >=, <=, ||, &&`)
- `a == b` vs `a.equals(b)`

Labs / Assignments:

- Worksheet: Evaluating boolean expressions
- Lab: Choose your own adventure - Students must write a program with nested if statements that allows the user to go on their own adventure. They will draw a corresponding flow chart of the possibilities.
- Lab: Rock / Paper / Scissors Game
- Lab: Calculator - Basic calculator in which the user inputs two numbers and an operation and the computer outputs the result to the console.

Iteration (Weeks 6-8)

Topics:

- `while` Loops
- `for` Loops
- Introduction to basic running time
- Infinite Loops

Labs / Assignments:

- Worksheet: For each sequence of numbers, the students must write a loop that prints out the sequence using only one print statement.
- Worksheet: For each set of given loops (possibly nested), students must count the number of executions of a particular statement. For an example, the code snippet

  ```
  int total = 0;
  for (int i = 0 ; i < 10 ; i++) {
      for (int j = i ; j < 10 ; j++) {
          total++;
      }
  }
  System.out.println(total);
  ```

  will print 55.
- Lab: Testing for primality of integers
- Lab: Greatest common factor

- Lab: Summing the digits of a number
- Lab: Printing patterns of asterisk
- Lab: Printing out a multiplication table
- Lab: Guess my number game - Students have to write a program that has the user guess a random between 1 and 100. At each step, the computer tells the user whether the number is higher or lower.

Classes Part (Weeks 9-14)

Topics:

- Classes conceptually
- Writing methods
- Method signatures & parameters
- Functional decomposition
- Creating new objects using classes
- Instance variables
- Constructors
- Overloading
- Public vs private
- Getters and setters
- Static vs not static
- Overriding `.toString()`
- Abstract classes & interfaces

Labs / Assignments:

- Lab: Methods lab. Write fifteen methods to accomplish tasks like "sum all the numbers from two parameters a and b", "print 'Hello' n times", "return the larger of two numbers", "determine whether a number is a multiple of another", "print all the powers of two up to 2 to the nth power", etc.
- Lab: Revisiting the guessing game: students are to take their number guessing game and split it up into functions (e.g. "getGuessFromUser", "printComputerResponse")
- Lab: Revisit Rock / Paper / Scissors lab with functional decomposition and add a feature to play again as well as a scoreboard feature.
- Lab: Pet Interface w/ Implementations according to preference
- Lab: Dog Class - Students must create a dog class with methods enabling the dog to bark, gain weight, etc. This class must also have a `.toString()` method.
- Lab: Elevens Introduction & Part 1 - design a card class

Miscellaneous Topics, Computer Hardware, & Architecture (Week 15)

Topics:

- Computer History
- Computer Components (Motherboard, processor, storage, peripherals, etc)
- Windows vs Apple vs Unix vs Linux
- Source code vs Byte Code vs Assembly vs Machine Code
- Command Line Basics

Labs / Assignments:

- Task: Build a virtual desktop computer online.
- Assignment: Report on one historical computer topic or figure. (E.g. Charles Babbage, Ada, Alan Turing, Claude Shannon, Grace Hopper, Bill Gates, Steve Jobs, Linus Torvalds, Richard Stallman, etc.)
- Lab: Write a script to back up all the files in a given directory
- Worksheet: Basic linux / unix command line commands (`ls, less, cd, cp, mv, rm, mkdir, rmdir`)

Review & Finals Weeks (Weeks 16-17)

Topics:

- Summary of topics
- Magpie Lab (I didn't have it at the appropriate time earlier in the semester)
- Final Exam

3 Semester Two Course Outline

Arrays & Searching (Weeks 18-21)

Topics:

- Informal running time analysis via counting iterations
- Declaring arrays, indices, using arrays
- Two-dimensional arrays
- Searching in a sorted array: binary vs sequential (and corresponding run times)

Labs / Assignments:

- Worksheet: Declaring and using arrays
- Lab: Sieve of Eratosthenes
- Lab: Write a binary search algorithm
- Lab: Picture Lab Activities 1-9
- Lab: Histogram creation given a set of values, and an interval. It will print out in text to the console.
- Lab: Picture Lab 1-9
- Exercises: Java array exercises at `<http://codingbat.com>`

Lists, `ArrayLists`, Selection & Insertion Sort (Weeks 22-24)

Topics:

- Lists and `ArrayLists`
- Arrays vs `ArrayLists` (advantages, disadvantages)
- Declaration, adding elements, insertion, deletion,
- Insertion Sort and Selection Sort
- Choosing data structures and algorithm for problem solving

Labs / Assignments:

- Worksheet: Lists / ArrayLists / Arrays worksheet
- Lab: Implement insertion sort and selection sort.
- Lab: Elevens Activities 2-4 (create a Deck class)

Inheritance (Weeks 25-27)

Topics:

- Subclasses (`extend`), superclasses, and class heirarchy
- Is-A vs Has-A classifications
- Overriding methods from superclasses
- Using class members
- Polymorphism

Labs / Assignments:

- Lab: Create three breeds of dogs inheriting from our earlier `Dog` class. This will include overriding some methods. For an example, whereas a standard generic dog might have a bark method that prints "Bark!" to the console, a French poodle might have have a method that prints "Le Woof!" to the console. This lab will also have examples of polymorphism and assigning e.g. poodles to variables with type `Dog`.
- Lab: Elevens Activities 6-9 (`Board` and `AbstractBoard` classes)

Recursion, Merge Sort (Weeks 28-30)

Topics:

- Recursive programming, base cases
- Comparison to sequences in math
- Towers of Hanoi Problem
- Factorial Function
- Merge Sort & Running time analysis
- Relation to Fractals

Labs / Assignments:

- Activity: Describe how to solve the Towers of Hanoi puzzle with words using a recursive algorithm. Also, calculate the number of moves required to solve the puzzle.
- Lab: Implement Factorial Algorithm recursively and iteratively
- Lab: Implement Fibonnaci Algorithm recursively and with arrays (also, analyze run times).
- Lab: Implement MergeSort Algorithm
- Lab: Use Context Free Art to create a fractal

Review (Weeks 31-33)

Topics:

- Overview & Review of Computer Science A

Labs / Assignments:

- Daily batch of multiple choice questions with discussion
- Practice of short-answer problems
- Practice Test

Ethics (Week 34)

Topics:

- Responsible Use of Computers
- Privacy Concerns
- Legality Concerns
- Intellectual Property / Software Piracy
- The role of social media in politics

Labs / Assignments:

- Report: Students are to pick one chapter from *Blown to Bits* and write a summary to share with the class about it.
- Report: Students are to pick one current event either from their lives or from the news and write a three-page blurb summarizing the event and describing what it means to them.

Post-AP Unwinding / Exposure to Miscellaneous Topics in Computer Science (Week 35)

So that students can unwind and see a little of what Computer Science has to offer, I want to do a survey of miscellaneous topics in Computer Science that I think are interesting. I'm not 100% sure which topics I will talk to them about but possible topics are listed below. I don't plan to have students do a lot of work surrounding these topics except listen. Following a few days of exposure, I would like them to pick a topic, research it independently, and convince me that they learned something about it, preferably through a write-up or a

- Depth-first search vs breadth-first search (queue vs stack)
- Game Theory
- Complexity Theory

- Information Theory
- Turing Machines
- Context Free Art / Context Free Grammars
- The Game of Life
- Cryptography
- Robotics / navigation
- Computer Vision
- Artificial Intelligence
- Quantum Computing

Labs / Assignments:

- Students must pick one task and implement it or research one topic and write a paper.

Post-AP Project (Weeks 36-37)

Topics:

- Topics will vary.

Labs / Assignments:

- Tasks will vary according to the students' preferences. More than likely, the students will implement either a card or a board game or a game like .

4 Teaching Strategies

This is to be a very hands-on class. After introducing the material and topic for the day, students are typically immediately given the opportunity to use the concepts for themselves through coding. At first they generally start out with small coding tasks doing little more than explore the definitions of the concepts. Next they work their way into more complex coding tasks, often using topics that we have worked with earlier in the course. For instance, if the topic is arrays, first the students will have a simple task declaring an array, filling in its values and printing them out. Following that, we'll move on to more complicated tasks like implementing the Sieve of Eratosthenes by using nested loops to find all the prime numbers from 1 to `n`.

I am fortunate enough to have a small class this year (19 students) so while they're working I have a chance to check in with everyone and see how they're doing and catch any misconceptions they students may have. I also have the good fortune of having this class right before lunch and it's common for students to stick around after class if they have unresolved questions. The students also work in pairs to do their coding about once a week, so they have opportunities to help each other out too. In addition, I provide tutoring half an hour before school starts upon student request.

Grading for the course falls under two headings: assignments and tests/quizzes/reports/projects. Because it's natural that some students complete the assignments in class faster than others, I provide students with ample extra credit opportunities to extend the assignments. Tests are generally a mix of multiple choice, fill-in-the-blank, and short answer questions. I also have students write code on the tests, usually on the order of a short (5-10 line) method.

5 Resources

Abelson, Ledeen, and Lewis. Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion. Crawfordsville, Indiana. Addison-Wesley Professional, 2008. <http://www.bitsbook.com/wp-content/uploads/2008/12/B2B_3.pdf>

Baker, Greg. *Introduction to Computing Science and Programming I, Fall 2010 Edition.* Burnaby, British Columbia. Simon Fraser University, 2010.

Coyne, Chris and Horigan, John. *Context Free Art.* <http://www.contextfreeart.org>.

Litvin, Maria and Litvin, Gary. *Be Prepared for the AP Computer Science Exam in Java, 5th Edition.* Andover, Mass. Skylight Publishing, 2013.

The College Board. New AP CS A Labs Magpie, Elevens, PictureLab. New York: College Entrance Examination Board, 2013.