AP Computer Science A

Miss Martinich … "Marty"

Room 231

Office Hours: W & Th 6:30-7:30 AM (or by appt)

rachel.martinich@ncps-k12.org

Course Overview

I proposed to bring AP Computer Science A to New Canaan High School because it is such a valuable course for all students, not just those who want to study Computer Science. It will enhance student problem-solving and abstraction abilities. They build analytical skills that are valuable in computer science, in other courses, and in life. Of course, students also increase their computer science and programming skills, skills that are needed in an ever-increasing array of college courses and workplaces.

The content and objectives of my AP Computer Science A course include the course objectives for AP Computer Science A as described in the AP Computer Science Course Description. This course focuses on an object-oriented approach to problem solving using Java. It includes the study of common algorithms and the use of some of Java's built-in classes and interfaces for basic data structures.

I expect all students to take the AP Computer Science A Examination. We work hard during the year to assure that you will have the opportunity to achieve a passing score on the exam.

Texts

Lewis, Loftus, and Cocking. Java Software Solutions for AP Computer Science 3rd Edition. Boston, Mass. Addison-Wesley, 2011.

Roselyn Teukolsky. Barron’s AP Computer Science A 8th Edition. Hauppauge, New York. Barron’s Educational Series, Inc, 2013.

Course Outline

Weeks 1-17 (Semester 1)

Weeks 1-2	**Computer Systems** Numerical representations; limitations of finite representations; number bases and conversion; hardware (primary and secondary memory); programming languages; and language interpreters and compilers.
Weeks 3-4	**Objects & Primitive Data** Simple data types `(int, boolean, double, char`); declarations (variable and constant); assignment and arithmetic expressions; console output (`System.out.print/println`); primitive types vs. objects; using classes to create objects; references; Java library classes (`String, Integer, Double, Math, Scanner`); and creating random numbers.
Weeks 5-6	**Program Statements – Conditional** Software Development Process; control flow (sequential and conditional); Boolean expressions, laws, and truth tables; using conditional expressions in `if`, `if-else`, and nested `if` statements; and `More` operators (increment, decrement, compound assignment).

Weeks 7-9	**Program Statements – Iteration** Flow of control (iteration); using `while` and `for` statements; infinite and nested loops; and analysis of algorithms (informal comparisons of running times and exact calculation of statement execution counts).
Weeks 10-12	**Writing Classes** Anatomy of classes, constructors, and methods; declarations (class, interface, instance variable, method, and parameter); method overloading; method decomposition; object relationships; reasoning about programs (assertions, pre- and post-conditions); data abstraction and encapsulation; and designing and implementing a class.
Weeks 13-15	**Enhancing Classes** References, exceptions, and class design; == vs. equals; object parameter passing; error handling (runtime exceptions, throwing runtime exceptions); interfaces and abstract classes; Java library classes (Comparable and List interfaces); and identifying reusable components from existing code using classes and class libraries.
Week 16	**Midterm Review**
Week 17	**Midterm**

Weeks 18-38 (Semester 2)

Weeks 18-21	**1D Arrays / 2D arrays / Searching** One- and two-dimensional arrays (creation,

	insertions, deletions, traversals, algorithms); searching algorithms and comparison (sequential and binary); and choosing appropriate data representation and algorithms.
Weeks 22-24	**Lists / ArrayLists / Selection and Insertion Sorts** `Lists` and `ArrayLists` (creation, insertions, deletions, traversals, algorithms); sorting algorithms and comparison (selection and insertion); and choosing appropriate data representation and algorithms.
Weeks 25-27	**Inheritance** Inheritance (subclasses, overriding, hierarchies, using class members, polymorphism, and class hierarchy design); interfaces and abstract classes; Java library classes (`Object`); reading and understanding class specifications and relationships among classes ("is-a" and "has-a"); understanding and implementing a given class hierarchy; extending a given class using inheritance; and applying functional decomposition.
Weeks 28-30	**Recursion / Merge and Quick Sorts** Recursive thinking, programming, and sorting; flow of control (recursion); sorting algorithms (merge and quick) and comparison with other sorts.
Weeks 31-33	**AP Test Practice Exam / AP Review** AP Computer Science A Examination (practice, content, materials, timing, tips).

Week 34	**Ethical and Social Implications of Computer Use; AP Examination** Responsible use of computer systems (system reliability, privacy, intellectual property, legal issues, and social and ethical ramifications of computer use).
Weeks 35-37	**Post-AP Project** Cooperative programming; research; reading code; and comparing strategies and algorithms.
Week 38	**Finals … YOU'RE DONE! :)**

Teaching Strategies

General Approach

AP Computer Science A is a substantial course that requires a meticulous approach from both you and me. The limited class time is fully utilized for discussion and activities, labs, quizzes, review, and tests. Outside class reading and homework is critical to your understanding of the material. Reading and comprehending technical material is a new skill for most students. They need to learn active reading techniques, including how to take notes.

I will typically begin each new unit of material with reading and homework assignments. This is followed by classroom discussion, related activities, and often additional homework such as worksheets. Students complete one or more related lab (programming) assignments. Finally, I have a review and a multiple-choice test. The Reviews utilize Multiple Choice, True/False, and AP-Style Multiple Choice questions that students have already completed as homework.

Differentiated Instruction

Different students learn in different ways. I will use a variety of teaching techniques including role-plays, student presentations, group work, and various multiple response strategies to engage students. Students also learn at different rates, so I utilize tutoring and extra credit assignments to address student needs at both ends of the spectrum.

Tutoring

In order to be successful in AP Computer Science A, it's critical that students learn the material in a timely fashion. Students who don't grasp earlier material don't have the foundation necessary for later material. Beginning after Winter Break, I also try and conduct weekly AP Exam Free Response tutoring sessions. These sessions cover Free Response questions from previous AP Computer Science A exams. These sessions should be mandatory for students who need them the most.

Extra Credit

It's important to keep all students engaged and learning. I provide "mandatory" and challenging extra credit labs for students who finish the normal lab assignments early. You will also have the option to pitch a personal project to me and you can work on that instead.

Lab Component

Writing computer programs is critical to understanding the course material. I assign at least one lab per unit. These assignments are typically completed on an individual basis. I use a program to randomly assign students to computers each day. This encourages collaboration among a variety of students, while at the same time discouraging copying among a few. Many students complete their lab assignments during class. I provide ample open lab time before and after school for students who need or want it.

I have integrated the AP Computer Science A Labs into the course at appropriate times based on their content, which account for a minimum of 20 hours of hands-on lab work (e.g., four hours on Magpie labs, six hours on PictureLab labs, and ten hours on the Elevens labs). You will complete the Magpie labs to help you develop your conditional statement skills. You complete the PictureLab labs to practice 2D-array algorithms. I have distributed the Elevens labs across the school year to complement different portions of the object-oriented curriculum. Students complete all of the required activities of the AP Computer Science A Labs. I will utilize some of the optional sections of the labs for extra credit assignments.

My lab computers have the Oracle Java SDK and the IntelliJ Interactive Development Environment. Program style (commenting/indentation/etc) is 10% of every lab grade and students' code must "compile" to receive any of this 10% credit. All of the Java-specific software we use in the classroom is available at no cost to both Mac and PC users.