NEWPORT HARBOR HIGH SCHOOL

AP COMPUTER SCIENCE A

Course Description and Goals

AP Computer Science is an introductory course in computer science. Because the development of computer programs to solve problems is a skill fundamental to the study of computer science, a large part of the course is built around the development of computer programs or parts of programs that correctly solve a given problem. At the same time, the development of useful computer programs and program modules is used as a context for introducing other important concepts in computer science, including the development and analysis of algorithms, the development and use of fundamental data structures, and the study of standard algorithms and typical applications. In addition, an understanding of the basic hardware and software components of computer systems and the responsible use of these systems are integral parts of the course.

AP Computer Science is an introductory course intended not just for those with a particular interest in the field of computer science but also for people who will major in other disciplines that require significant involvement with computing. Ultimately, I want all my students to enjoy the class; to find the algorithms and logical challenges stimulating, engaging, and fun. In addition to the many exercises and projects found in ICT's Java Curriculum for AP Computer Science (v. 2.0), I have compiled and/or designed additional projects that I believe students will find interesting and fun and, very importantly, will maintain those students that are very adept at programming, and show uncommon aptitude, interested and engaged.

Students will:

- design and implement computer-based solutions to problems in several application areas.
- learn well-known algorithms and data structures.
- develop and select appropriate algorithms and data structures to solve problems.
- code fluently in a well-structured fashion using the Java programming language.
- read and understand a large program and a description of the design and development process leading to such a program.
- 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.

Computer Language

The AP Computer Science Examination requires the use of Java. The current introductory programming course takes an object-oriented approach to programming that is based on encapsulating procedures and data. The exam will not cover all the features of Java; it will be consistent with the AP Java subset as described by the College Board:

AP Computer Science Course Description

AP Computer Science A – Syllabus

Course Resources	All class materials – including complete lesson notes (ICT's Java Curriculum for AP Computer Science (v. 2.0), in-class and homework assignments, all supplementary projects and lessons, and all files, materials, and narrative pertaining to the GridWorld Case Study – are available to students at all times at the teacher website: **Primary Curriculum Resource:** ICT's Java Curriculum for AP Computer Science (v. 2.0) **Compiler: BlueJ** **Required Texts:** Teukolsky, Roselyn. *How to Prepare for the AP Computer Science Exam*, 3rd Edition (Java). Barron's Educational Series **Recommended and Reference Texts:** Quesenberry, Nancy. *Java Curriculum for Advanced Placement (TM) Computer Science*, version 2.0 revision. Institute of Computer Technology (ICT), 2006. Horstmann, Cay. *Java Concepts*. 4th ed. Hoboken, N.J.: Wiley, 2006. Big Java - C Horstman - http://horstmann.com/bigjava.html Lambert, Ken, and Martin Osborne. *Fundamentals of Java: AP* Computer Science Essentials for the A & AB Exams*. 3rd ed. Boston: Thomson Course Technology, 2006. Savitch, Water. JAVA, An Introduction to Computer Science & Programming. 2nd Ed. **Teacher Website: APCS Teacher's Web Page**
Prerequisites	• Completion of Geometry and Algebra II with a grade of C or better. • Previous math teacher recommendation (signature).
Computer Facilities	Students work independently in the computer lab no less than 3 hours a week. Lab work is extremely important for two reasons. First, it allows me to see students' coding in action. I casually walk from student to student, looking at their progress on assigned programs and getting a good idea of how each person is progressing. I can check for understanding or check coding style (indentations, naming conventions for variables, methods, classes, etc.). My instant feedback enables a student to make corrections "on the spot." Second, students can ask specific questions about concepts with which they are having trouble, and I can give personalized assistance. Also, next year I will have a student aide (he will be a senior) who has completed the course (earned a 4 on the AP exam) and can assist me in helping students who have issues that can be easily resolved and/or are experiencing problems with the computer, compiler, etc. In all cases, students are introduced to the AP Java subset delineated in Appendices A and B of the AP Computer Science Course Description.
Teaching Methods	The methodologies described herein for the course are learned through class discussion, projects that vary in length and level of difficulty, development of concept ideas through deconstructionism (the process of learning concepts by examining a completed product), predicting program segment outcomes, verifying program correctness, and other activities. Students will be evaluated by the quality of the programming assignments, quizzes/homework, hands-on programming tests and lab exercises, unit exams (multiple-choice and free-form), and a Final Exam (1 semester) and Major Project (2 semester). Part of the 1 semester Final Exam requires that all students submit an individual portfolio including all major (and clearly specified) programming assignments (program code and sample output).

Curriculum Components:

Lesson A1, A2, and so forth, refer to the student lessons and topics in *ICT's Java Curriculum (version 2.0).* Similar references are to ICT's Lesson Handouts (e.g., A.1), Worksheets (e.g., A.1), and Lab Assignments (e.g., A.1).

In addition to the many lab projects/programs that form part of this curriculum, students will spend time with the three exemplar labs described in the **AP Computer Science Course Description.**
The first lab (Magpie) can is incorporated early in the course and involves simple string processing and conditional execution. The second lab (Picture Lab) involves 2-dimensional array manipulation in the context of image processing. The third lab (Elevens) provides an example of larger object-oriented program design.

Abbreviations Used Below:

class discussion [cDisc]
student reading [sRead]
lecture [lect]
programming assignment [pAssign]
teacher directed program completion [tDirPC]
teacher directed activity [tDirAct]
discussion circle [dCircle]
student essay / outline / writing piece [sEssOutWrite]
optional project for the advanced student [advPro]

Week(s)	Description of Learning	Outcomes of Learning Assignments and Evaluation
1-2	• Use of computers in the lab • Lesson A1 - Introduction to Object Oriented Programming (OOP) • Lesson A2 - Object Oriented Programming Source Code – Bytecode – Virtual Java Machine (how to write a Java program) • General Computer Knowledge (AP Topic VI. A-D) Piracy and Acceptable Computer Use • Simple I/O	• Identify computer components [sRead] [cDisc] • Understand and articulate the process for writing a Java program [tDirPC] Handout A.1 [sRead] Handout A.2 [sRead] Worksheet A.2 [sRead] • Describe legal issues involving piracy and the acceptable use policies in computer science [sRead] [lect] • Use simple I/O to create a rudimentary Java program Lab A7.1 – GroceryList [pAssign - 2 days] • Student Design [advPro]
3	• Lesson A3 - Primitive Data Types	• Declare variables, store values in them, learn operations to manipulate and use those values, and print out the values using the System.out object. [sRead] [tDirAct] Handouts A3.1, A3.2 [sRead], Worksheets A3.1-A3.4 [tDirAct] Lab A3.1 – Easter [pAssign - 2 days], Lab A3.2 – Coins [pAssign - 2 days]
4	• Lesson A4 – Object Behavior	• Understand Java conventions and rationale for object-oriented programming [sRead] [cDisc] [tDirPC] Lab A4.1 – MPG [pAssign - 2 days] • Lab A4.2 – Rectangle [pAssign - 2 days] • Additional Project [advPro]

5-6	• Lesson A5 – Designing and Using Classes	• Design your own classes and determine object behavior [sRead] [cDisc] [tDirPC] Worksheet A5.1, A5.2 [tDirAct] Lab A5.1 - PiggyBank [pAssign - 2 days] Lab A5.2 – Müller [pAssign - 3 days] • Additional Projects [advPro]
7	• Lesson A6 – Libraries and API's	• Learn how to read the APIs, use pre-written classes, and understand the elements of the AP Java Subset [sRead] [cDisc] [tDirAct] Worksheet A6.1, A6.2 [tDirAct] Lab A6.1 – Taxes [pAssign - 3 days] Lab A6.2 – RegularPolygon [pAssign - 3 days] • Understand the creation of JavaDocs Handout A6.1 [sRead]
8-9	• Lesson A7 – Simple I/O • Lesson A8 - Control Structures	• Use conditional, relational, and logical operators to construct control structures in programs [sRead] [cDisc] [gProj] [tDirPC] Worksheet A8.1, A8.2 [tDirAct] Lab A8.2 – IRS [pAssign - 3 days] • Use the switch statement to control a programs flow • Additional Project [advPro]
10-12	• Lesson A9 - Recursion	• Anticipate the outcomes of recursive methods, understand the process of the recursive method [sRead] [cDisc] [tDirAct] Worksheet A9.1 [tDirAct], Handout [tDirAct] Lab A9.1 – Fibonacci [tDirPC] • Write simple recursive methods to perform tasks Lab A9.2 – KochCurve [pAssign - 1 day] • Additional Projects [advPro]
13-15	• Lesson A10 – Strings	• Understand what an immutable object represents, use a variety of methods from the String class, understand what object references are [sRead] [cDisc] [tDirPC] [dCircle] Worksheet A10.1, A10.2 [tDirAct] • Solve string-processing problems Lab – Exemplar Lab - Magpies [pAssign - 2 days] Lab A10.3 – RomanNumerals [pAssign - 3 days] • Additional Projects [advPro]
16-20	• Lesson A11 – Inheritance • Lesson - Inheritance, Polymorphism, and Abstract Classes	• Use single inheritance, use inheritance to build class hierarchies in a program, use method overriding to modify the behavior of a subclass [sRead] [cDisc] [tDirAct] Lab A11.1 – BackToSchool [tDirAct] Lab A11.2 – GraphicPolygon [pAssign - 2 days] • Explain abstract methods and classes, describe and use polymorphism, read and understand interfaces, create inheritance from predefined classes [sRead] [cDisc] [tDirAct] • Additional Project [advPro] • Lab – Exemplar Lab – Elevens [pAssign - 2 days]

21	• Lesson A13 – Exceptions and File I/O	• Understand and describe exception handling; create try-catch blocks to use exception handling [sRead] [cDisc] [tDirAct] Worksheet A13.1 [tDirAct] Lab A13.1 – ErrorCheck [tDirAct] • Additional Projects [advPro]
22	• Boolean Algebra	• Use of DeMorgan's Laws to solve Boolean algebra problems • Understand and construct Truth Tables [sRead] [cDisc] [tDirAct]
23-25	• Lesson A15/A16 - Arrays and ArrayLists • Lesson A17/A19 - Sorting and Searching • Lesson A17 - Quadratic Sorting Algorithms	• Understand Abstract Data Types (ADTs) and how an array implements the List ADT, create objects of type ArrayList using generics, use methods of ArrayList to access, add and remove elements, learn casting [sRead] [cDisc] [tDirAct] [gProj] [dCircle] Handout [sRead], Worksheet [tDirAct] Lab A15.1 – IrregularPolygon [tDirPC] Lab A15.2 – Permutations [pAssign - 2 days] Lab A15.3 – Statistics [pAssign - 1 day] • Declare and create single-dimension arrays and use them in programs; fundamental algorithms by programming arrays for insertion, deletion, and traversal [sRead] [cDisc] [tDirAct] [gProj] • Use searches to find elements within data structures Worksheet A19.1 [sEssOutWrite] Worksheet A19.2 [sEssOutWrite] Lab A19.1 [pAssign - 2 days] • Trace the execution of quadratic sorting algorithms, write programs to use sorts, determine efficiency of sorting algorithms [sRead] [cDisc] [tDirAct] • Additional Projects [advPro]
26-28	• Lesson A21 – Two-Dimensional Arrays	• Lab – Exemplar Lab – Picture Lab [pAssign - 2 days] • Additional Projects [advPro]
29	• Lesson – Number Systems	• Will be able to convert between binary, octal and hexadecimal number systems [cDisc] [tDirAct] Handout [tDirAct] • Additional Projects [advPro]

Newport Harbor High School

A California Distinguished School

600 Irvine Avenue • Newport Beach, CA 92663
Phone (949) 515-6300 • Fax (949) 515-6370

Michael Vossen	Tina Case	Jack Cusick	Dave Martinez
Principal	Assistant Principal	Assistant Principal	Assistant Principal

Vision Statement: Founded in 1930, Newport Harbor High School uses shared decision-making to create a dynamic and challenging school environment.

Mission Statement: By challenging students to reach their academic and personal potential, we believe we can help all students become productive and successful members of our global society.

WE ARE:	S-Scholarly	P-Persistent
	A-Artistic	R-Respectful
	I-Interconnected	I-Innovative
	L-Leaders	D-Diligent
	O-Outstanding	E-Enlightened
	R-Responsible	

General School Policies*

Attendance:
Proper attendance to this class is required. Classwork missed due to truancy cannot be made up. ANY STUDENT WITH TEN TRUANCIES TO THIS CLASS MAY BE DROPPED FROM IT WITH A FAILING GRADE. (EC: 49067, BP: 5121) Truancy is a class absence not properly cleared in the Attendance Office; multiple truancies will have an impact on your grade in this course. If dropped, students may discuss credit recovery options in the Counseling Office.

Tardy Policy:
Students are expected to report to this class on time. Students who incur multiple tardies may be subject to disciplinary consequences by the teacher and/or school administration, which may include (but are not limited to) detention, Saturday Detention, and unsatisfactory Citizenship marks.

Citizenship Grades:
Students who earn unsatisfactory marks in Citizenship ("U") in any of their classes may find themselves suspended or removed from co-curricular activities and/or placed on a behavior contract for regular review. Two or more unsatisfactory citizenship grades on the quarter reports are automatic removal from participating in co-curricular activities such as: performances, games, etc.

Academic Integrity:
All students are expected to complete their work ethically, honestly, and as directed by their teachers. Academic dishonesty includes (but is not limited to) copying another student's classwork or homework, cheating on tests, and plagiarism; these are serious offenses that inhibit the educational process, and they will not be tolerated. Students who are academically dishonest may see their grades reduced, their citizenship marks affected, their athletic and co-curricular eligibility jeopardized, and their ability to obtain letters of recommendation for college applications from teachers and counselors endangered. All school and department policies regarding Academic Integrity will be followed.

*** these are only the sections that apply to this class**