# Course: AP® Computer Science “A” 2007-2008

Description and Syllabus

## Description of Course:

AP Computer Science A is designed to:

- Train students in programming methodology to produce quality computer-based solutions to real problems. This is achieved by helping students learn to combine technological knowledge and programming skills with problem-solving skills.
- Challenge students with hands-on lab exercises and worksheets to enable them to become experienced Java programmers.
- Provide clear and concise lessons in topics such as object-oriented programming, algorithms, static and dynamic data structures, sequence, repetition, conditions, methods, one-dimensional arrays, and recursion.

After completing this course, students should be able to do the following:

- Write code in the Java programming language.
- Understand and use object-oriented programming with appropriate data structures to solve problems.
- Use current methodologies to analyze a problem, develop program specifications, design the solution using top-down methodology and stepwise refinement, implement the solution using a high-level language (Java), utilize program verification, and create program documentation.

The time frame of this course is one year divided into three 12-week trimesters. Please note the exam will occur in the 33rd week.

## Textbooks:


Students will be given access to ICT’s curriculum, composed of student lessons and supplementary materials (in Microsoft Word© format) and/or will be users of a password-protected website for the student lessons (in web-page HTML format).


Each student will have a copy of the textbook and access to the online Wiley plus system or teacher website.


A classroom set of these books will be available.
Prerequisites: A knowledge of basic algebra and experience in problem solving. Recommended course, Introduction to Programming in Alice. This is a one trimester course that uses the Alice software to teach the fundamentals of Object Oriented programming.

Description of Computer Facilities:

Students will meet and work in the computer lab 3 times a week for 45 minute periods and once a week for a 90 minute period. Most days students are using the computers for at least part of the period. Time is divided into lectures, guided practice, independent or cooperative programming projects and exams or quizzes. In all cases, students are introduced to the AP Java subset delineated in Appendices A and B of the AP Computer Science Course Description.

There are 20 computers in the computer lab. Each is equipped with Window XP, Java JDK 1.5, BlueJ, JCreator, notepad, MS Office, Inspiration and internet access. Class size is usually between 6 – 10 students.

Teaching Methods:

The methodologies described herein for the course are learned through class discussion, interactive role play, visual simulation, development of concept ideas through deconstructionism (the process of learning concepts by examining a completed product), creating solutions to problems via written algorithms, predicting program segment outcomes, verifying program correctness, and other hands on manipulative activities.

Students will be evaluated by the quality of their programming assignments including documentation, planning, creativity, and efficiency, quizzes/homework/class exercises, unit tests (objective (MC) and free-form). All programming assignments are listed in this document and will be assigned as time permits.

Curricular Requirements

C2: The course includes all of the topics listed in the "Computer Science A" 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 Appendixes A and B of the AP Computer Science Course Description.

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.
**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). Ch 1, Ch 2, and so forth, refer to the Java Concepts book, 4th edition. RE refers to Review Exercises, PE refers to Programming Exercises in the Java Concepts book. SG 1, SG 2 and so forth, refer to chapters in the Advanced Placement Computer Science Study Guide. SGMC refers to multiple choice questions in the study guide.

**Elements of the required GridWorld case study are also called out:**

AP Topic I, AP Topic II, and so forth, refer to the AP Computer Science Course Description Topic Outline posted on the College Board Advanced Placement website. C2, C3, C4, C5, C6, C7, C8, C9 are the Curricular Requirements for the AP Audit.

Tests are highlighted in Blue. All tests contain a multiple choice section and a free response section.

**Abbreviations Used Below:**

class discussion [cDisc]
student reading [sRead]
lecture [lect]
programming assignment [pAssign]
These come from the ICT labs, Java Concepts Programming Exercises and my own creations.
teacher directed program completion [tDirPC]
teacher directed activity [tDirAct]
group project [gProj]
homework assignment [hwAssign]
Test or Quiz [Test]
student essay / outline / writing piece [sEssOutWrite]

<table>
<thead>
<tr>
<th>Week(s)</th>
<th>Description of Learning</th>
<th>Outcomes of Learning Assignments and Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| C2 C6 C8 C9 | • Ch 1 – Introduction, SG 2 (AP Topic VI. A-D)  
             What is programming  
             The anatomy of a Computer  
             The Java programming Language  
             Types of errors (AP Topic III B)  
             Lesson A22 - General Computer Knowledge (AP Topic VI. A-D)  
             Piracy and Acceptable Computer Use  
             Lesson A1 - Introduction to Object Oriented Programming (OOP) (AP Topic III. C-D)  
             (AP Topic II A)  
             Lesson A2 - Object Oriented Programming  
             Source Code – Bytecode – Virtual Java Machine (how to write a Java program) (AP Topic I. A)  |
|         | • Log in to the Wiley Plus website for Java Concepts book  
             • Read Ch 1 and Lesson A22 [sRead]  
             • Discuss Ch 1 and Lesson A22 [lect] [cDisc]  
             • Assign as homework RE Ch 1: 1-7, PE Ch1: 2-6 [hwAssign]  
             • Introduction to BlueJ – Instructions for setting up at home HelloTester.java program  
             • Do as a class RE Ch 1: 8-12, PE Ch1: 1 [tDirAct]  
             • Visit ACM code of ethics and The computer Ethics Institute “The Ten Commandments of Computer Ethics” online. [cDisc]  
             • describe legal issues involving piracy and the acceptable use policies in computer science [sRead] [cDisc]  
             • Assign SGMC Ch 2: 1-10 [hwAssign]  
             • understand and articulate the process for writing a Java program [tDirPC]  
             • Handout A.1 [sRead]  
             • Worksheet A.1 [tDirAct]  
             • Lab A1.1 – DrawHouse [pAssign - 1 day]  
             • Handout A.2 [sRead]  
             • Worksheet A.2 [sRead]  
             • Lab A2.1 – Benzene [pAssign - 1 day]  
             • Test on Ch 1, Lessons A22, A1 and A2 [Test]  |
| 3-5 | **Ch 2 – Using Objects**  
(AP Topic III. C-D,H)  
(AP Topic II A)  
**Lesson A3 - Primitive Data Types**  
Ch 4 Fundamental Data Types  
(4.1 – 4.5)  
(AP Topic II B.1)  
(AP Topic IV A)  
(AP Topic III B)  
(AP Topic III H)  
**Lesson A6 - Libraries and APIs**  
(AP Topic II C) | **Read Ch 2.1 – 2.4 [sRead]**;  
**Discuss Ch 2.1 - 2.4 [lect]**;  
**RE Ch 2: 1-5 [hwkAssign]**  
**Read & Discuss Ch 2.5 – 2.10 [sRead] [lect]**  
**Explore the API online [tDirAct]**  
**learn how to read the APIs, use pre-written classes, and understand the elements of the AP Java Subset [sRead] [cDisc] [tDirAct]**  
**RE Ch 2: 6-12; SGMC Ch 3 1-10 [hwkAssign]**  
**PE Ch 2: 1, 4, 7, 9 [pAssign]**  
**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 [hwkAssign]**  
**Lab A3.1 – Easter [pAssign - 2 days]**  
**Lab A3.2 – Coins [pAssign - 2 days]**  
**Worksheet A6.1 [hwkAssign], A6.2 #2 [pAssign]**  
**RE Ch 4: 1-4,6-10, 14,15, 16 a-h; SGMC Ch 5 1-10 [hwkAssign]**  
**Worksheet A22.1- Base conversions**  
**Test on Ch 2.Ch 4.1-4.5, Lesson A3 and A6 [Test]** |
|---|---|
| 6-7 | **Ch 3 Implementing Classes**  
**Lesson A4 - Object Behavior**  
Java Software conventions  
(AP Topic II. B.1-2)  
(AP Topic I B) | **Objects role play [tDirAct]**  
**understand Java conventions and rationale for object-oriented programming [sRead] [cDisc] [tDirPC]**  
**Create a simple Calculator class using Javdocs [tDirAct]**  
**Lab Exercise Ch 3 – Vending machine and Post Card [tDirAct]**  
**RE Ch 3 1-8, 12; Worksheet A4.1 – Design Sphere class [hwkAssign]**  
**Lab A4.1 – MPG [pAssign - 2 days]**  
**develop a program**  
**Lab A4.2 – Rectangle [pAssign - 2 days]** |
| 8-10 | **Lesson A5 - Designing and Using Classes**  
(AP Topic I B.1,3)  
**Lesson A6 - Revisited**  
**GridWorld Narrative Chapter 1**  
(AP Topic – Case Study) | **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] (or Extra Credit if no time)**  
**understand the creation of JavaDocs**  
**Handout A6.1 [sRead]**  
**Lab A6.1 – Taxes [pAssign - 3 days]**  
**Lab A6.2 – RegularPolygon [pAssign - 3 days]**  
**SGMC Ch 4 1-10 [hwkAssign]**  
**observe and experiment with the GridWorld GUI test Actors’ methods and determine the purposefulness of classes use the drop-down to call methods of executable classes GridWorld narrative [sRead]**  
**Do You Know Questions – Set 1, page 6 of GridWorld [cDisc]**  
**GridWorld Exercise Set, page 8 [tDirAct]**  
**Test Ch 3, Lesson A4, A5, Gridworld Ch 1 [Test]** |
| 11 | **Lesson A7 - Simple I/O**  
Ch 4.7 – Scanner class, Formatting (AP Topic Optional) | **use simple I/O to create a rudimentary Java program**  
**Lab A7.1 – GroceryList [pAssign - 2 days]**  
**Worksheet A7.1, A7.2 [hwkAssign]** |

**End Trimester One Grading**
| 12 | **Lesson A8 - Control Structures**  
    Ch 6 Decisions  
    (AP Topic II B.4)  
    (AP Topic II A.1)  
    (AP Topic IV B) | use conditional, relational, and logical operators to construct control structures in programs  
   Worksheet A8.1, [hwkAssign]  
   RE Ch 6: 1, 2, 5, 6, 8, 9, 11, 17 [hwkAssign]  
   PE Ch 6: 1, 4 [pAssign]  
   Lab Biorhythm Calculator – After getting the user’s day of birth and current date, calculate their 3 biorhythms and display. Create and use a simple date class and use Gregorian calendar algorithm to calculate the days alive. (algorithm is given in class) [pAssign]  
   SGMC Ch 6 1-10 [hwkAssign]  
   Test Ch 6 and Lesson 8 [Test] |
|---|---|
| 13-15 | **Lesson A12 – Iterations**  
    Ch 7 - Iteration  
    (AP Topic II B.4)  
    **Lesson A14 - Boolean Algebra and Loop Boundaries**  
    (AP Topic II B.4)  
    The game of Craps program is the first larger program students write. They begin with planning and pseudocode. It uses 2 interacting classes, a reusable Dice class that is tested in isolation and a craps class that can be played interactively with output or run multiple times to determine the win percentage without output.  
    (AP Topic I B, III A) | write an iterative block of code using the for control structure, nested loops, while loop, do-while control structure  
   Worksheet A12.1,2,3 [hwkAssign]  
   RE Ch 7: 1-6, 8, 11 [hwkAssign]  
   Lab A12.1 - FunLoops  [pAssign - 1 day]  
   Lab A12.2 - Pictures  [pAssign - 1 day]  
   Choose one of the following:  
   Lab A12.3 - Loan Table  [pAssign - 1 day]  
   Lab A12.4 - Grades  [pAssign - 2 days]  
   Lab A12.5 - Payments  [pAssign - 1 day]  
   Lab A12.6 - ParallelLines  [pAssign - 1 day]  
   Lab A12.7 – GameLand  [pAssign - 1 day]  
   write negations of Boolean assertions, use DeMorgan's Laws to solve Boolean algebra problems, establish post condition assertions for conditional loops, build boundary conditions for conditional loops using assertions and DeMorgan's Laws.  
   Worksheet A14.1 [hwkAssign]  
   Lab Craps  [pAssign - 3 day] (see explanation on left)  
   SGMC Ch 7 1-15 [hwkAssign]  
   Test Ch 7, Lessons 12, 14 [Test] |
| Winter Break |  |
| 16 – 17 | **Lesson A11 – Inheritance Intro**  
    (AP Topic III F)  
    (AP Topic III A)  
    (AP Topic VI D)  
    (AP Topic II B.2)  
    **GridWorld Narrative Chapter 2**  
    (AP Topic – Case Study) | use single inheritance, use inheritance to build class hierarchies in a program, use method overriding to modify the behavior of a subclass  
   GridWorld narrative [sRead]  
   Lab A11.1 – BackToSchool  [tDirAct]  
   Lab A11.2 – GraphicPolygon [pAssign - 2 days]  
   Do You Know Questions – Set 2, Page 11 of GridWorld [cDisc]  
   GridWorld Exercise Set, Page 12-13: 1,2,3 [pAssign - 3 days] |
| 18-19 | - Lesson A10 – Strings  
     Ch 4.6 Strings  
     (AP Topic I B.1,3,4)  
     (AP Topic III A, F)  
     (AP Topic IV B)  
     - 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 [hwkAssign]  
     - solve string-processing problems  
     Lab A10.3 – RomanNumerals [pAssign - 3 days]  
     - Hangman Lab [pAssign - 4 days] – Design and implement the game of hangman using a hangman class and a display class (one graphical or one using println – same graphical interface)  
     - Test Ch 4.6, Lessons 11 &10, GridWorld Ch 2 [Test] |
| 20  | - Lesson A13 - Exceptions and File I/O  
     (AP Optional – I/O)  
     (AP Topic III B, E)  
     - Reference Ch 15 & 16  
     - understand and describe exception handling; create try-catch blocks to use exception handling. Read data from simple text files. [sRead] [cDisc] [tDirAct]  
     Worksheet A13.1 [tDirAct]  
     Lab A13.1 – ErrorCheck [tDirAct]  
     Lab A13.2 - Average [pAssign - 1 day] |
| 21  | - Lesson A9 – Recursion  
     (AP Topic II B.4.d,e)  
     - anticipate the outcomes of recursive methods, understand the process of the recursive method [sRead] [cDisc] [tDirAct]  
     Worksheet A9.1 [tDirAct]  
     - Extra practice Recursion Worksheet [hwkAssign]  
     - SGMC Ch 15 1-6 [hwkAssign]  
     Lab A9.1 – Fibonacci [pAssign] – Compare iterative and recursive solutions both in time and space.  
     - Show Towers of Hanoi problem and explain solution [tDirAct]  
     - write simple recursive methods to perform tasks (AP optional)  
     Lab A9.2 – KochCurve [pAssign - 1 day]  
     - Quiz Recursion and Exceptions [Test] |
| 22-23 | - Lesson A16 - Single Dimensional Arrays  
     (AP Topic IV C)  
     (AP Topic V A,B)  
     - 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] [gPro]  
     Handout A16.1 [sRead]  
     Worksheet A16.1 [hwkAssign]  
     Lab A16.1, Statistics [pAssign - 2 days]  
     - Ch 8 Arrays and ArrayLists  
     - demonstrate the concept of arrays using “arrays of candy” with bottle tops as the physical container. Illustrate array initialization and indexing.  
     - declare and create single-dimension arrays and use them in programs; fundamental algorithms by programming arrays for insertion, deletion, and traversal [sRead] [cDisc] [tDirAct] [gPro]  
     Handout A16.1 [sRead]  
     Worksheet A16.1 [hwkAssign]  
     RE Ch 8: 6, 9,11, 12, 17, 18 [hwkAssign]  
     SGMC Ch 8 1-14 [hwkAssign]  
     Lab Blackjack: This lab uses 3 classes, a card class, a deck class and the Blackjack class. The first two classes are almost entirely given to the students. The Blackjack class has only method stubs. Both an ArrayList of Cards and arrays are used. In addition to the lab, students are asked to explain how the data structures might change if we have more than just the dealer and 1 player but many players. They must also write the algorithms for at least two poker hand outcomes given an arraylist of 5 cards as a parameter. [pAssign - 4 days] [sEssOutWrite] |
| 24  | Lesson A20 - Inheritance, Polymorphism, and Abstract Classes (AP Topic I A.5) (AP Topic I B) (AP Topic III) Reference Ch 11 & 13 – Interfaces, Polymorphism & Inheritance | explain abstract methods and classes, describe and use polymorphism, read and understand interfaces, create inheritance from predefined classes [sRead] [cDisc] [tDirAct] Lab A20.1 Farm [pAssign - 4 days] RE Ch 11: 1 – 4, 6, 7, 8; RE Ch 13: 1 – 12 [hwkAssign] SGMC Ch 11 & 12 1-10, [hwkAssign] Test Ch 8, Lesson 15, 16, and 20 |
| 28  | Lesson A17 - Quadratic Sorting Algorithms Ch 19 Sorting and Searching (AP Topic V C.1,2) (AP Topic III G) | Physically sort students to illustrate each of the 3 quadratic sorting algorithms. Examine graphical sort applet on the web to demonstrate each sort. [tDirAct] trace the execution of quadratic sorting algorithms, write programs to use sorts, determine efficiency of sorting algorithms [sRead] [cDisc] [tDirAct] Worksheet A17.1,2,3 [tDirAct] Lab A17.1 Sort comparison [pAssign - 2 days] Graph Results [sEssOutWrite] |
| 29  | Lesson A18 - Merge and MergeSort Ch 19 Sorting and Searching (AP Topic V C.3) (AP Topic III G) | write a function to merge data, write a recursive mergeSort routine, determine efficiency of sorting algorithms [sRead] [cDisc] [tDirAct] Worksheet A18.1 [tDirPC] Lab Assignment A18.1 Merge [pAssign - 2 days] Lab Assignment A18.2 Recursive Mergesort [tDirAct] Graph Results and summarize [sEssOutWrite] |
| 30 C2 | Lesson A19 - Searches - Sequential and Binary  
Ch 19 Sorting and Searching (AP Topic V B) | use sequential and binary searches to find elements within data structures. Compare searches and situations for the best use of each  
Worksheet A19.1 [sEssOutWrite]  
Worksheet A19.2 [sEssOutWrite]  
Lab A19.1 Store [pAssign - 2 days]  
Lab A19.2 Search [pAssign - 2 days]  
Lab A19.3 Countwords [pAssign - 2 days] (time permitting)  
Test Ch 19, Lesson 17, 18, and 19 [Test] |
|---|---|
| 31-32 C3 | Review for the exam  
Review of converting between binary, octal and hexadecimal number systems, understand how numbers are stored in a computer, and recognize the limitations of finite representations [sRead] [cDisc] [tDirAct]  
Worksheet A21.1 [tDirAct]  
Lab A21.1 [pAssign - 2 days] (time permitting) |

AP Exam  
End of Timester Three grading