AP Computer Science A Syllabus

Course Resources

Primary Text: *Java Software Solutions for AP Computer Science*, John Lewis, Williams Loftus and Cara Cocking, 2004, Addison Wesley.

Objects First With Java: A Practical Introduction Using BlueJ, David J. Barnes and Michael Kölling, 2003, Prentice Hall.

Addison-Wesley's Review for the AP Computer Science Exam in Java*, Susan Horwitz, 2004, Pearson Education Limited.

Barron's How to Prepare for the AP Computer Science Advanced Placement Exam Java* Version, Roselyn Teukolsky, 2003 Barron's Educational Series, Inc.

Video: *Computer Security*. 1999 Films for the Humanities and Sciences.

Video: *Cybercrime: The Invisible Threat*. 2000 Discovery Channel.

Video: *Lifting the lid: How Computers Work*. 2002 Films for the Humanities and Sciences.

AP® GridWorld Case Study.

Java is the primary programming language.

The classroom consists of 24 computers for student use. Every student has use of their own computer for at least 3 hours a week for programming.

AP Computer Science A Topic Outline

I. Object-Oriented Program Design

The overall goal for designing a piece of software is to correctly solve the given problem. At the same time, this goal should encompass specifying and designing a program that is understandable, can be adapted to changing circumstances, and has the potential to be reused in whole or in part. The design process needs to be based on a thorough understanding of the problem to be solved.

A. Program Design *The learner will:*

- A1. Read and understand a problem description, purpose, and goals.
- A2. Apply data abstraction and encapsulation.
- A3. Read and understand class specifications and relationships among the classes ("is-a", "has-a" relationships).
- A4. Understand and implement a given class hierarchy.
- A5. Identify reusable components from existing code using classes and class libraries.

B. Class Design *The learner will:*

- B1. Design and implement a class.
- B2. Choose appropriate data representation and algorithms.
- B3. Apply functional decomposition.
- B4. Extend a given class using inheritance.

II. Program Implementation

The overall goals of program implementation parallel those of program design. Classes that fill common needs should be built so that they can be reused easily in other programs. Object-oriented design is an important part of program implementation.

C. Implementation Techniques *The learner will:*

- C1A. Object-oriented development.
- C1B. Top-down development.
- C1C. Encapsulation and information hiding.
- C1D. Procedural abstraction.

D. Programming Constructs. *The learner will:*

- D1. Primitive types vs objects.
- D2A. Constant declarations.
- D2B. Variable declarations.
- D2C. Class declarations.
- D2D. Interface declarations.
- D2E. Method declarations.
- D2F. Parameter declarations.
- D3. Console output (System.out.print/println)
- D4A. Control using methods.
- D4B. Sequential control.
- D4C. Control using conditionals.
- D4D. Control using iteration.
- D4E. Control using recursion.

E. Java Library Classes *The learner will:*

E1. Utilize the AP Java subset of classes.
(Those included in the A-level AP Java subset.)

III. Program Analysis

The analysis of programs includes examining and testing programs to determine whether they correctly meet the specifications. It also includes the analysis of programs or algorithms in order to understand their time and space requirements when applied to different data sets.

F. Testing *The learner will:*

F1. Test classes and libraries in isolation.
F2. Identify boundary cases and generate appropriate test data.
F3. Perform integration testing.

G. Debugging *The learner will:*

G1. Categorize errors: compile-time, run-time, logic.
G2. Identify and correct errors.
G3. Employ techniques such as using a debugger, adding extra output statements, or hand-tracing code.

H. Understand and modify existing code *The learner will:*

H1. Understand and modify existing code

I. Extend existing code using inheritance *The learner will:*

I1. Extend existing code using inheritance

J. Understand error handling *The learner will:*

J1. Understand runtime exceptions.

K. Reason about programs *The learner will:*

K1. Document pre- and post- conditions
K2. Implement assertions

L. Analysis of algorithms The *learner will:*

L1. Evaluate informal comparisons of running times.
L2. Exact calculation of statement execution counts.

M. Numerical representations and limits *The learner will:*

M1. Evaluate representations of numbers in different bases.
M2. Evaluate limitations of finite representations (e.g. integer bounds, imprecision of floating-point representations, and round-off error.)

IV. Standard Data Structures

Data structures are used to represent information within a program. Abstraction is an important theme in the development and application of data structures.

N. Simple data types. *The learner will:*
- N1. Utilize simple data types: int, boolean, double.

O. Classes *The learner will:*
- O1. Utilize classes.

P. One-dimensional arrays *The learner will:*
- P1. Utilize one-dimensional arrays.

V. Standard Algorithms

Standard algorithms serve as examples of good solutions to standard problems. Many are intertwined with standard data structures. These algorithms provide examples for analysis of program efficiency.

Q. Operations on A-level data structures previously listed *The learner will:*
- Q1. Implement Transversals.
- Q2. Implement Insertions.
- Q3. Implement Deletions.

R. Searching The learner will:
- R1. Implement Sequential Search.
- R2. Implement Binary Search.

S. Sorting The learner will:
- S1. Implement Selection Sort.
- S2. Implement Insertion Sort.
- S3. Implement Mergesort Sort.

VI. Computing in Context

A working knowledge of the major hardware and software components of computer systems is necessary for the study of computer science, as is the awareness of the ethical and social implications of computing systems. These topics need not be covered in detail but should be considered throughout the course.

T. Major hardware components *The learner will:*

- T1. Primary and secondary memory.
- T2. Processors.
- T3. Peripherals.

U. System software *The learner will:*

- U1. Language translators/compilers.
- U2. Virtual machines.
- U3. Operating systems.

V. Types of systems *The learner will:*

- V1. Single-user systems.
- V2. Networks.

W. Responsible use of computer systems *The learner will:*

- W1. Systems reliability.
- W2. Privacy.
- W3. Legal issues and intellectual property.
- W4. Social and ethical ramifications of computer use.

Units of Study **Curricular Requirements (C2)**

Unit 1: Introduction to Computer Systems
Curricular Requirements (C3, C4, C5, C6, C8)

Students will study computer architecture, memory, and peripheral devices. They will explore the BlueJ environment by compiling and running several ready-made programs. Students will then create a program from scratch in order to understand some aspects of the Java programming language such as comments, identifiers, whitespace, syntax, and errors.

Sample Student Activity for Unit 1:
1. Write an essay summarizing the video.
2. Write an application that displays your initials in large block letters. Make each large letter out of the corresponding regular character.

Resources:
1. *Java Software Solutions for AP* Computer Science* - Lewis, Lofting, Cocking
2. Video: *How Computers Work*. 2002 Films for the Humanities and Sciences.

Unit 2: Foundations of Object Orientation
Curricular Requirements (C3, C4, C5, C6)

Students will explore the concepts of objects and classes. They will experiment with the uses for objects and classes, how to create them, as well as how to interact with them. Students will experiment with a "LabClass" program with classes and objects to simulate a class data base. They will also use classes and objects to create a picture of a house using graphics.

Sample Student Activity for Unit 2:
1. Using the LabClass program, create a LabClass object and several student objects. Enter those students into the lab and use the inspector to discover what each field contains.
2. Use the shapes from the Shapes project (a class example) to create an image of a house and a sun.

Resources:
1. *Java Software Solutions for AP* Computer Science* - Lewis, Lofting, Cocking
2. *Objects First With Java* - Barnes, Kölling

Unit 3: Methods and Classes

Curricular Requirements (C3, C4, C5, C6)

Students will write classes that will define objects that can be created and manipulated within a program. Students will explore the details of class definitions, including the structure and semantics of methods and the scope and encapsulation of data. They will explore interfaces, inheritance polymorphism, as well as a variety of standard classes such as wrapper class, math class, random class and object class.

Sample Student Activity for Unit 3:
1. Create a program that designs a rocket with different modules. Each module is generated by a different method (nose cone, body, flame).
2. Create a BankAccount class that includes member fields (balance, password) and member methods (deposit, withdraw).

Resources:
1. *Java Software Solutions for AP* Computer Science* - Lewis, Lofting, Cocking
2. *Objects First With Java* - Barnes, Kölling

Unit 4: Conditional and Repetition Structures

Curricular Requirements (C3, C4, C5)

Students will utilize conditional statements (if, if-else, switch) to create programs that allow for a change in the logical flow of the program based on decision-making ability of these conditional statements. Students will also create programs that make use of repetition structures (for, while, do while). Students will demonstrate the ability to choose the most appropriate control structure based on the program specifications.

Sample Student Activity for Unit 4:
1. Create a DateChecker class that allows the user to create a date object. The class will then validate the month, day and year using appropriate conditional statements.
2. Design and implement a Dice class that creates a 6-sided die object. Roll the die 100 times and keep track of the number of times each side is rolled. Display the results in a table format.

Resources:
1. *Java Software Solutions for AP* Computer Science* - Lewis, Lofting, Cocking
2. *Objects First With Java* - Barnes, Kölling

Unit 5: Advanced Programming Structures and Algorithms
Curricular Requirements (C3, C4, C5, C6)

Students will design programs using advanced programming structures such as single and multi-dimensional arrays, ArrayLists, and Strings. Students will also utilize and compare several sorting algorithms (Insertion Sort, Selection Sort, Merge Sort) and will explore recursion.

Sample Student Activity for Unit 5:
1. Create a program that will test String objects recursively to see if they are palindromes.
2. Create a program that will create a Record object in a RecordCollection class. They will sort the collection (ArrayList) by the attributes of the Record object (artist, song name, album name, date). Students will implement Sort, Selection, and Merge sorting algorithms and compare their efficiency. Students will also search the collection using both Sequential and Binary search algorithms and compare their efficiency.

Resources:
1. Java *Software Solutions for AP* Computer Science* - Lewis, Lofting, Cocking
2. *Objects First With Java* - Barnes, Kölling

Unit 6: Program Design and Analysis
Curricular Requirements (C3, C4)

Students will analyze algorithms for program design, test final product for errors (syntax, run-time, compile-time), and will debug programs as necessary. Students will explore the software development life cycle. They will implement robust code in their programs.

Sample Student Activity for Unit 6:
1. Before every program assignment, student will write a program analysis of the problem and consider different solutions.
2. For each program, create pseudo-code/flowchart as a guide for creating the actual code for the program.
3. For each assignment, properly test code to check for possible errors that may make the program crash or produce invalid results.

Unit 7: AP GridWorld Case Study
Curricular Requirements (C3, C4, C5, C6, C7)

Students will discover aspects of programming such as dealing with large scale programs and robust code by exploring the AP GridWorld Case Study. They will modify and add to several classes contained in the Case Study.

Sample Student Activity for Unit 7:
1. Modify the behavior of an existing bug using inheritance.
2. Create a new type of bug with its own unique attributes and behaviors.

Resources:
1. AP GridWorld Case Study.

Unit 8: The Computer and Society
Curricular Requirements (C9)

Students will explore how the computer has affected society. Topics include the development of computer languages, ethics, computer careers, security and technology's affect on a global society.

Sample Student Activity for Unit 8:
1. Write an essay summarizing the video.
2. Create a web site or PowerPoint presentation on a selected topic from above.

Resources:
1. Video: *Computer Security. Films for the Humanities & Sciences*, 1999.
2. Video: *Cybercrime: The Invisible Threat. Discovery Channel*, 2000.

Unit 9: Getting Ready for the AP Exam

Students work through sample multiple-choice questions and free-response questions. They will take a mock exam to simulate the timing and type of questions to expect on the actual exam.

Resources:
1. AP GridWorld Case Study
2. *Barron's How to Prepare for the AP* Computer Science Advanced Placement Exam*, Java Version. Teukolsky, 2003.
3. *Addison Wesley's Review for the AP* Computer Science Exam in Java*. Horwitz, 2004.

Curricular Requirements

C1 - The teacher has read the most recent AP Computer Science Course Description, available as a free download on the AP Computer Science A Course Home Page.

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. (Note: Students who study a language other than Java in AP Computer Science must also be taught to use Java, as specified in the AP Java subset.)

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.