COURSE PROFICIENCY OUTLINE ADVANCED PLACEMENT COMPUTER SCIENCE - 1343

5 Credits

Purpose

Advanced Placement Computer Science is a course whose goals are comparable to those of a first semester course offered in college by college and university Computer Science departments.

The course will place emphasis on object-oriented programming (OOP) methodology with a concentration on problem solving and algorithm development. The concepts of OOP include class declaration, inheritance, and polymorphism. Applications of these concepts are used in student assigned labs.

While studying OOP, students will be able to understand program design by applying data abstraction and encapsulation. Focused data structures will include simple data types (int, boolean, and double), classes, and one dimensional arrays. Applications of these data structures are studied in standard searching and sorting algorithms with recursive and non-recursive techniques. General concepts of sequential, conditional, and iterative executions are reviewed.

Approximately one fourth of the AP Computer Science test includes questions regarding the current case study. This class will apply all OOP concepts to the case study and assign labs that will demonstrate the students' understanding of the OOP topics discussed.

I. Student Outcomes 4.3, 4.4, 4.5

The major goal of the Advanced Placement Computer Science course is successful student results on the AP Computer Science examination administered by the College Entrance Exam Board.

A. The student will be able to use OOP concepts to code fluently in a well-structured fashion using the latest programming language for Advanced Placement study.
B. The student will be able to design and implement computer-based solutions to the problems in several application areas.
C. The student will learn well-known algorithms and data structures.
D. The student will be able to develop and select appropriate algorithms and data structures to solve problems.
E. The student will be able to recognize the ethical and social implications of computer use.

II. Content 4.3, 4.4, 4.5

A. Object-Oriented Program Design
1) Program Design
a) Read and understand a problem description, purpose, and goals.
b) Apply data abstraction and encapsulation.
c) Read and understand class specifications and relationships among the classes ("is-a" "has-a" relationships).
d) Understand and implement a given class hierarchy.
e) Identify reusable components from existing code using classes and class libraries.

2) Class Design
a) Design and implement a class.
b) Choose appropriate data representation and algorithms.
c) Apply functional decomposition.
d) Extend a given class using inheritance.

B. Program Implementation

1) Implementation techniques a) Methodology
   - o Object-oriented development
   - o Top-down development
   - o Encapsulation and information hiding
   - o Procedural abstraction

2) Programming constructs a) Primitive types vs. objects b) Declaration
   - o Constant declarations
   - o Variable declarations
   - o Class declarations
   - o Interface declarations
   - o Method declarations
   - o Parameter declarations c) Console output d) Control
     - o Methods
     - o Sequential
     - o Conditional
     - o Iteration
     - o Recursion

3) Java library classes

C. Program Analysis

1) Testing
   a) Test classes and libraries in isolation.
   b) Identify boundary cases and generate appropriate test data.
   c) Perform integration testing.

2) Debugging
   a) Categorize errors: compile-time, run-time, logic.
   b) Identify and correct errors.
   c) Employ techniques such as using a debugger, adding extra output statements, or hand-tracing code.

3) Understand and modify existing code.

4) Extend existing code using inheritance.

5) Understand error handling.
   a) Understand runtime exceptions.

6) Reason about programs
   a) Pre- and post-conditions.
   b) Assertions.

7) Analysis of algorithms
   a) Informal comparisons of running times.
   b) Exact calculation of statement execution counts.

8) Numerical representations and limits
   a) Representations of numbers in different bases.
   b) Limitations of finite representations (e.g., integer bounds, imprecision of floating-point representations, and round-off error)

D. Standard Data Structures 1) Simple data types (int, boolean, double) 2) Classes 3) One-dimensional arrays

E. Standard Algorithms

1) Operations on A-level data structures previously listed.
a) Traversals b) Insertions c) Deletions

2) Searching a)
Sequential b) Binary

3) Sorting a)
Selection b) Insertion c)
Mergesort

F. Computing in Context

1) Major hardware components a) Primary and secondary memory b) Processors c) Peripherals

2) System software a) Language translators/compilers b) Virtual machines c) Operating systems

3) Types of systems a) Single-user systems b) Networks

4) Responsible use of computer systems a) System reliability b) Privacy c) Legal issues and intellectual property d) Social and ethical ramifications of computer use

III. Materials

A. Text: Java Concepts John Wiley & Sons, Inc. Advanced Placement
B. Study Guide: Computer Science Study Guide John Wiley & Sons
C. Barron's AP Computer Science
D. Previous AP Tests

IV. Evaluation

A. The student will be expected to complete classwork, homework, keep a notebook and take tests and quizzes. These will be checked and reviewed by the teacher.
B. The student will be expected to demonstrate an acceptable level of proficiency in the objectives and content of this course.
C. The student will be expected to demonstrate at all times appropriate classroom behavior such as self-control, respect for others, respect for property and a mature attitude.
D. The student will be expected to adhere to the school rules and regulations for behavior and the district policy for attendance.
E. Students will be required to successfully pass the High School Proficiency Assessment as mandated in the graduation law (N.J.S.A. 6:8-4.2).
F. Students who fail the HSPA examination will be placed in a Basic Skills Math class as required by N.J.S.A.6:8-4.2. There will be no exceptions to this requirement.
G. The final grade represents the teacher's professional judgment of the student's performance and all of the aforementioned activities and/or requirements are included in the evaluative process.

September 2007 Reviewed and
Revised August 2008
Reviewed August 2010
Reviewed August 2011

Reviewed and revised: August 2010