

SALINE AREA SCHOOLS

COURSE OUTCOMES

COMPUTER SCIENCE

The Course

Computer Science is a one-semester course that introduces the student to the “science” of computer programming. A large part of the course is built around the development of computer programs that correctly solve a given problem. The course also emphasizes the design issues that make programs understandable, adaptable, and, when appropriate, reusable. At the same time, the development of useful computer programs 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. The programming language used will be Java.

Prerequisites

Both an aptitude for sequential, logical thinking and strong symbolic manipulation skills are important for success in Computer Science. Before studying Computer Science, all students must be competent in the basic algebraic concepts of solving equations, evaluating expressions, and evaluating functions. In addition, students should be comfortable with manipulating symbols, recognizing mathematical patterns, and deductive reasoning. The prerequisites for Computer Science can be satisfied by successful completion of Integrated Algebra II, Algebra III or Honors Algebra III.

There are no computing prerequisites for this course. The course is designed for students with no prior computer programming experience. However, students with prior programming experience who take the course can expect to benefit substantially from the study of “good” programming practices and efficient algorithm development.

Philosophy

In this Computer Science course, broad concepts and widely applicable methods are emphasized. The focus of the course is neither the intricacies of the Java programming language nor the memorization of disparate facts about computer science. Rather, the course seeks to lay a strong foundation for looking at computer science as a problem-solving tool whose key elements involve design, implementation and analysis. Structured problem-solving, procedural and data abstraction and careful analytic reasoning are themes throughout the course.

The cohesive nature of the course must be emphasized through incremental development of the techniques and skills of computer science with analysis of pre-existing code, continuous distributed review and frequent cumulative testing. Since an important aspect of computer science is its problem-solving nature, the course must also involve frequent lab projects. Unifying computer science themes must be revisited over and over again, with continual applications to increasingly complicated problems.

Objectives

1. Students understand the basic elements of computer hardware and software and their roles in a computer system.
2. Students master essential elements of Java syntax and good programming style and learn to pay attention to code clarity and documentation.
3. Students are able to draw on data types and variables to represent information used by a program. They can make use of simple data types and aggregate data structures, including int, char, bool, double, strings and arrays. They understand and can specify the scope of variables.
4. Students acquire proficiency with programming constructs, including sequential, conditional and repetitive control structures, functions and input/output.
5. Students understand the concept of an algorithm as a step-by-step procedure and can implement some standard algorithms in Java.
6. Students develop testing strategies and data for their programming.
7. Students distinguish among compile-time, run-time, and logical errors. Students recognize and correct bugs using software debugging tools and techniques.
8. Students acquire skills in designing software solutions to problems from various application areas. They can specify the software's purpose, goals and limitations.
9. Students appreciate OOP (Object-oriented programming) concepts, for its reusability, level of abstraction and its methodological approach.
10. Students gain some practice in using library classes and packages for developing GUI (graphical user interface) applets and applications.
11. Students are aware of the importance of responsible use of computer systems. They recognize ethical and social implications related to the use of computing systems.

Topical Outline

The outline of topics is intended to indicate the scope of the course, but not necessarily the order in which topics are taught. Although the final exam for Computer Science will be based on the topics listed in this topical outline, teachers may enrich the course with additional topics.

I. Introduction to Hardware, Software, and the Internet

- Hardware, software and Internet overview
- Representation of Information in Computer Memory: Numbers; Characters

II. Designing Web Pages with HTML

- Document structure tags; text layout and formatting tags; anchors and links; lists; images; tables; applets.

III. An Introduction to Software Development

- Overview of algorithms; compilers and interpreters, software components and packages; object-oriented programming

IV. Java Classes, Objects, and Events: A Preview

- Classes and objects
- Classes and source files
- Imported classes and packages
- Fields, Constructors, and methods
- Events

V. Java Syntax and Style

- Using comments
- Reserved words and programmer-defined names
- Syntax vs. style
- Statements, blocks, indentation

VI. Data Types, Variables, and Arithmetic

- Declaring fields and local variables
- Primitive data types
- Constants
- Scope of variables
- Arithmetic expressions
- Compound assignment and increment operators

VII. Boolean Expressions and if-else Statements

- If-else statements, and nested if-else, switch statements
- Boolean data type
- Relational and logical operators
- Order of operators

VIII. Strings

- String properties and methods
- Declaring and using literal strings
- String constructors and commonly used String methods

IX. Arrays

- One- and two-dimensional arrays
- Accessing array elements

X. Iterative Statements: while, for, do-while

- Loop structures
- Iterations and arrays

XI. Streams and Files

- Java's I/O Package
- Reading text from the Console
- Opening text files for reading and writing
- Reading and writing lines of text