

Java Programming

Mr. Shubert, instructor

An elective for high school students who have taken the two required Technology Courses, this course presents the great ideas of computer science. These great ideas are the important models and concepts which enable people to understand the field of computer science. We use a problem-solving approach and write many programs in the Java programming language, and study basic computer architecture, operating systems, networks, and the possibilities and limitations of computing in general.

Text:

Great Ideas in Computer Science with Java, by Alan W. Biermann and Dietolf Ramm. MIT Press, 2002. 0-262-02497-7.

Materials:

1-subject notebook for computing journal

USB flash memory stick for storing files, or email account for transferring files

Expectations:

Each student will keep a computing journal, keeping study questions, definitions, procedures, and class and project notes, which will be checked periodically.

Assignments: Some assignments will be completed in class, and some will be taken home and completed by the due date. (For work at home, the Java programming tools are free and run on almost any computer.)

Participation: Each student will be a valuable member in classroom discussions and presentations. Adequate participation consists in reading the material and having worthwhile things to say. Excellent participation goes beyond this to include asking appropriate questions and encouraging other students to share their ideas.

Project: A larger programming assignment, on a subject of each student's choice.

Tests: One midterm and one final exam, with additional quizzes at the instructor's discretion.

Academic Honesty:

Our reputation depends on maintaining the highest standards of intellectual honesty. All of us will conduct ourselves so that together we establish a community of trust where no one plagiarizes, cheats, or obtains unauthorized academic materials.

Grading:

Computing journal/Participation	15%
Assignments	25%
Project	20%
Tests and quizzes	40%

Course Outline:

Week 1. Read ch. 1.

Review of HTML. Creating web pages. Java is not JavaScript. Introduction to the lab. Installing Java software and documentation. Short paper: Evaluating my computer and programming experiences and interests.

Weeks 2-3. Read ch. 2.

Java program structure: applets and applications. Compiling and executing programs. Decision trees (branching). Simple input and output. Interactive programs: buttons and text boxes. Debugging techniques, including the IDE (integrated development environment). Lab: Design and build an application and an applet, and compare. Design and build a decision-tree program.

Weeks 4-5. Read ch. 3.

Functions, arrays, and numerical computation. Simple calculations. Using functions to find sums and statistics. Lab: Design and build a program that uses arrays and performs numerical calculations.

Weeks 6-8. Read ch. 4.

Designing and building larger programs: top-down programming. Lab: Design and build a program which interacts with a simple database class.

Week 9-10. Read ch. 5.

Graphics, classes, and objects. Class hierarchies and inheritance. Lab: Design and build a program with a graphical interface that draws a collection of house images.
Review and midterm exam.

Weeks 11-12. Read ch. 6.

Simulation. How do different playing strategies affect game outcomes? Introduction to threads. Lab: Design and build a program with an improved strategy simulating a race car, or modify another simulation program.

Week 13. Read ch. 7.

Software engineering. Different approaches to large-scale programming projects. Short paper: investigate a large software project and lessons learned from it.

Week 14. Read ch. 8.

Machine architecture. Writing programs in a simplified machine language and running them on a simulator.

Week 15. Read ch. 9.

Language translation. Writing programs in a simplified Java-like language and translating them into machine language.

Week 16. Read ch. 10.

Operating systems. Working with different levels in computer systems: user applications, operating systems, and hardware. Multi-threading.

Week 17. Read chs. 11-12.

Security and privacy. Defending against malicious code. Encryption. Computer communications and networks. Lab: modify a networked chat program to encrypt messages for security.

Week 18. Read ch. 13.

Program execution time; "hard" problems. Review and final exam.