

## 63.152: Computer Science I Using Java SYLLABUS

### Course description (FSC catalog)

An introduction to problem solving using the Java programming language. The course stresses algorithms, object-oriented programming in graphical environments, documentation, testing, and debugging. Topics include hardware basics and number systems, classes, methods, control structures, types, virtual-machine concepts, Internet and client-server computing, human-computer interaction, social, professional, and ethical issues, and general features of programming languages.

*Corequisite:* 43.123 College Algebra **or** minimum score of 2 on the math placement examination.

### Overview

Computer science is a scientific discipline with mathematical and empirical concerns. Computation is the manipulation of abstract symbols but can only occur in the physical world. We use abstraction and reasoning when we design solutions to computational problems, and we test these solutions on actual computing devices.

Computer science is much more than programming, but many courses in computer science use computer programming to present many of the concepts of computer science. Java is an object-oriented programming language widely used in the software industry.

This course looks at a number of other languages and encourages a comparison among them. We will begin with a brief introduction to the HTML markup language used to construct web pages, followed by a short look at the Javascript language, which is similar to Java in some ways but which is embedded in HTML web pages. HTML and Javascript are easy to test with a web browser.

We will also introduce some hardware concepts of computer science, centered around the notion of a processor that executes instructions in machine language. A human-readable language that is otherwise close to machine language is assembler language.

Programming begins with an emphasis on design. This course includes an introduction to the design tools of flowcharts, pseudocode, module hierarchy charts, and the Unified Modeling Language (UML). We direct attention to design of both algorithms and non-algorithmic, interactive computational processes.

You will think logically to plan a solution to a problem; then you will code a program, test it, and correct errors, documenting your program with comments and meaningful identifiers.

This course is challenging and may be of special interest because it may demand new ways of thinking about problem solving, including logical inference, the creation of data types, and the creation of subprograms (programs within programs). Exercises will require extensive use of a Java compiler.

This course emphasizes hands-on learning in the lab or on your computer. Our tools will include a Java compiler (Sun's Java Development Kit) and a graphical software development environment, BlueJ.

The central organizing element of a Java program is the *class*, an abstract specification of objects as aggregates of properties and associated operations. The second organizing principle is the *method*, a subprogram that implements operations for an object (class). We will instantiate, implement, and design classes in Java. The notions of classes and objects are the central ones in the object-oriented design and programming paradigm.

The notion of data type (e.g., integer, real number, string, class) is presented, along with the three control structures used in algorithm design: sequences, branches and loops. Structured types include classes (aggregates of items of different types) and arrays (aggregates of items of the same type).

### Required readings

Textbook: Cay Horstmann, *Big Java*, 3<sup>rd</sup> edition (Wiley, 2008), chapters 1-11.

For topics 1-3, chapter-length handouts by the instructor will be used.

*Slides:* See handouts and course web site.

Reading and studying text material related to the course is a must. The slides and the study questions provide a guide to which concepts are most important to this course, but to understand this material, in-depth reading is required as well, reinforced by exercises.

### To contact instructor:

Office hours (Hemenway Hall 318A):

M 12:30-1:20 p.m., W 10:30-11:20 p.m.,

Th 2:30-3:20 p.m., others by appointment

Office: Hemenway Hall 318A

Telephone: (508) 626-4724

Email: [dkeil@frc.mass.edu](mailto:dkeil@frc.mass.edu)

URL: [www.framingham.edu/faculty/dkeil](http://www.framingham.edu/faculty/dkeil)

### Expected skills on entry

- Some experience using a computer;
- Competence in high-school algebra as shown by score of “2” in FSC math placement test, or by participation in College Algebra (43.123);
- Readiness to read a college textbook and ask questions in and out of class.

### Classroom format

Format will center on group work and problem solving, lecture (with slides), and discussion. Your questions and participation are important. Part of our responsibility is to express doubt.

We will write some program code on the board; other code we will compile and run using a projector. Frequently we will divide into in-class teams to solve discussion problems.

We will normally have one conversation in the classroom and an environment similar to that of a focused workplace governed by attention to business and by mutual respect.

### Web aspect of course

This course is listed under <http://framingham.blackboard.com>. All course materials will be available there. The site hosts a private discussion board for students enrolled in the course.

### Group work

Different problems will be assigned to groups. Each student will present a problem solution in class and groups will post solutions at the Discussion Board. Quizzes will be based in part on these problems.

### Programming project

A Java programming project is assigned that is intended to be constructed step by step as the course proceeds, making use of control structures, procedural modules, file input/output, and class design. It will

provide experience in documentation of process, documentation of code, correctness, and testing.

### Grading

Group work	25
Major quizzes (2)	20
Mini-quizzes (8)	20
Final exam	20
Project final submission	10
Participation	<u>5</u>
	100 %

### Performance objectives

1. To trace a looping and branching computation specified in a flowchart;
2. To recognize the concepts of an algorithm and of the three basic control structures;
3. To recognize the hardware components of a general-purpose computer and how they interact in the fetch-execute cycle;
4. To specify, design, compile, test, and debug programs;
5. To write comments to document code;
6. To code screen output and keyboard input;
7. To declare variables and assign them expressions as values;
8. To manipulate binary numerals;
9. To design solutions to simple programming problems;
10. To recognize the basic vocabulary of structured and object-oriented programming;
11. To recognize basic syntax rules of Java;
12. To use the standard Java data types *int*, *float*, *double*, *char*, *boolean*, and *String*;
13. To solve simple word problems by designing and coding programs, using valid branch and loop statements, including nested loops;
14. To define and use simple Java classes and methods.
15. To program with arrays and collections.
16. To build a simple graphical user interface;
17. To build a simple application that manages and updates file data.

### Accommodations

Students who seek accommodations during the semester because of disabilities should meet with the instructor after class or during office hours early in the semester.

## Course Plan

<b>Dates</b>	<b>Topic</b>	<b>Reading</b>
1/28-1/30	Introduction	
2/4-2/11	1. HTML and JavaScript	<i>Handout</i> : "Introduction to JavaScript"
2/13-2/25	2. Computer organization and hardware	<i>Handout</i> : R. Johnson, D. Keil, "A model processor and its assembler language"
2/27-3/5	3. Problem solving and program design	<i>Handout</i> : R. Johnson, D. Keil, "Problem solving and program design"
3/10	<i>Quiz 1</i> (topics 1-3)	
3/12-3/27	4. Java programming: statements, types, objects, input/output, methods	Horstmann, Chs. 1-2, 4 <i>Handout</i> : "Using Java development tools"
3/28-4/4	5. Control structures: Branch and loop statements	Horstmann, Chs. 5-6
4/7-4/14	6. Methods and classes	Horstmann, Ch. 3
4/16	<i>Quiz 2</i> (topics 4-6)	
4/18-4/28	7. Arrays and collections	Horstmann, Ch. 7
4/30-5/5	8. Building GUIs and file-maintenance applications	Horstmann, Ch. 8-11
5/7	<i>Semester review</i>	
May 12, 9:00 a.m.	<i>Final exam</i> (Topics 1-8)	