

Course: **CS-4811 Java Programming (Section 101)**  
This *summer session* runs for 6 weeks (through Wed. July 2, 2003).

Lecture: M, W: 4:00pm - 6:50pm (CC-43)  
Instructor: Dr. Jeffrey Blessing  
E-mail: [blessing@msoe.edu](mailto:blessing@msoe.edu)  
Web: <http://www.msoe.edu/~blessing/cs4811>  
Office: B-305  
Hours: The hour before and after this class, and by appointment  
Phone: 277-7194  
Text: *Core Java 2 - Volume 1 - Fundamentals*, by Cay Horstmann and Gary Cornell,  
Sun Microsystems Press, Prentice-Hall, 1999. (ISBN #0-13-081933-6)

Description: In just a few short years, Java has become one of the most important programming languages in computing. It combines a general purpose programming language with a wealth of libraries to provide for all aspects of modern-day computing, in one standardized language. With Java, programs can be written which will run in a stand-alone environment or on a web page from an Internet browser. All forms of networking are possible, as well as multi-threaded applications that take advantage of parallel, multi-process architectures. Graphics and windowing applications are also incorporated into the language, as well as database access using all standard data models. The Java language continues to grow, to encompass virtually every purpose for which people write programs. All this is done in an operating system and architecture-neutral way, which promises programmers that applications written for Java will run on virtually any platform. In short, all the programming tasks, which used to require several languages to implement, can now be done in one language: Java.  
Credits: (2-2-3)

Objective: This course is designed for experienced programmers who have not had extensive exposure to object-oriented programming methodologies. A general overview of all language elements, including control statements and operators, is included to completely describe the language. C programmers will be most familiar with the basic language elements of Java, since they are virtually the same in both languages. Likewise, C++ programmers will be equally comfortable with Java's inheritance and polymorphism mechanisms, although there are important differences between the two languages.

Prerequisites:

1. A *thorough* understanding of high level language programming constructs (expressions, data types, operators, control stmts, functions, etc.) and object-oriented programming features (objects, classes, methods, inheritance, polymorphism, etc.). The C++ language is the basis for the Java language.
2. A *fundamental* understanding of data structures (arrays, strings, lists, trees, graphs, etc.) and the algorithms that manipulate them.
3. A *basic* understanding of the Software Engineering process, including the analysis, specification, design, implementation, and testing of software systems.
4. Access and knowledge of using browsers on the World Wide Web.

Grading:	Programs	40%
	Weekly Quizzes	30%
	Final (week 11)	30%

No incomplete grades or makeup exams will be given (without very extenuating circumstances)

Attendance: Although not reflected in the grading for this course, attendance is strongly encouraged. If chronic absenteeism is a problem, attendance will be taken. Exam material will come from both the text and class discussions. **Note:** Students will NOT be automatically dropped from the roster if 3 successive classes are missed.

Late Policy: Assignments are due at the beginning of the class period that was designated as their due date. A 10% penalty will be assessed for each day an assignment is late.

Topics:

- An Introduction to Java*
  - History, features, the web, the enterprise
- The Java Programming Environment*
  - The JDK, other development environments for Java
- Fundamental Programming Structures in Java*
  - Expressions, data types, variables, operators, control statements, classes, methods, arrays
- Objects and Classes*
  - OOP basics, using existing classes, building your own classes, packages
- Inheritance and Polymorphism*
  - Java's use of inheritance, polymorphism, casts, abstract classes, run-time issues
- Interfaces and Inner Classes*
  - Multiple Inheritance, interfaces, use of inner classes
- Graphics Programming*
  - Windows, the AWT, platform dependent GUI's
- Event Handling*
  - The AWT event model, GUI's, multicasting, advanced topics
- Graphical Interfaces with Swing components* (platform independent GUI's)
  - Model-view-controller, layout managers, text fields, choice boxes, scroll bars, menus, dialog boxes, etc.
- Applets*
  - HTML tags, multimedia, contexts, simple networking (client-server model)
- Exception Handling*
  - Errors, exceptions, try-catch blocks, debugging techniques
- Streams and Files*
  - File-based I/O, streams, zip file streams, object streams, file mgmt.