Syllabus for
CS 134 – Java Programming

Computer Science Course Catalog 2000-2001: This course is an introduction to object-oriented programming using the Java language. In the first part of the course, Java applets are used to build graphics tools and introduce object-oriented design methods. In the second part of the course, Java applications are used to build a complete graphics interface and illustrate the OOP concepts of polymorphism and inheritance.
Prerequisite: None, Little or no previous programming experience

Textbook: *Java for Programmers, by D Lyon*
In the SOE office...
Computer Usage: Students **MUST** have access to a computer with a Java compiler. Metrowerks CodeWarrior Academic Pro for Windows 95, 98, NT is strongly suggested and will be used in class. In the book store (about $126).
E-mail access is required.
Course Notes: Handouts/diskettes/e-mail, web page
When: Friday, 6:30;
Where: Bannow Science Center, Room 257
Who: Dr. D. Lyon
Phone: (203)641-6293
Fax: (203)877-4187
E- mail: lyon@docjava.com
Web Page: http://www.docjava.com
Office Hours:
Monday, Friday; 6 pm - 6:30 pm in class room (Dr. Lyon)
Monday, Friday; 5 pm - 6:00 pm in Mc108 (Dr. Lyon)
Or by appointment (Dr. Lyon).

Course Objectives:

- Write clear, elementary Java programs (applets and applications)
- Use a Java-enabled browser and/or the appletviewer to execute Java applets
- Use the Java interpreter to run Java applications
- Understand basic HTML World Wide Web programming
- Understand algorithmic thinking and apply it to programming
- Understand problem-solving techniques
- Read, write, and debug Java programs
- Write programs using object-based programming techniques including classes, objects and inheritance
- Program Java keyboard input and screen output
- Code with arithmetic, increment, decrement, assignment, relational, equality and logical operators
- Code control structures (if, if/else, switch, while, do/while, for) and use primitive data types
- Use basic graphical user interface (GUI) components including buttons and text fields
- Write user-defined methods
- Understand and manipulate single-subscripted arrays (arrays are Java objects)
- Pass single-subscripted arrays to methods
- Process strings (strings are Java objects)

Student Activities: Learning a new computer language is very much a hands-on activity, which cannot be learned from lectures or textbook reading alone. It does require those lectures and textbooks, but the real learning results from the laboratory trials and the homework assignments. To achieve the course objectives, the student must have good class attendance and participation, conduct the computer programming tasks during the laboratory periods as well as the assigned homework. Homework assignments and laboratory trials are due at the beginning of the class following the assignments. They are to be placed in an envelope containing the student's name. The contents of the envelope will be a diskette and a paper copy of the requested Java source code.

Course Requirements: The schedule of activities and topics to be covered each week are outlined below. Each week will begin with responses to questions and a brief review on the previous week's topics. The first week will begin with administrative announcements and a review of this syllabus.
Join the List: Every student should join the e-mail list for the course, available on the bottom of the Programming in Java web page at:
<http://www.docjava.com/java/cs134/sw406.htm>

Key Topics

* Java-enabled World Wide Web browsing
* Java Development Kit (JDK) and the Java programming environment (compiler, interpreter, appletviewer, debugger)
* Java virtual machine
* Java Applications Programming Interface (API)
* Differences between the Java programming language and C/C++
* Hypertext Markup Language (HTML)
* Applets and applications
* Input/output streams, sequential-access files and random-access files
* Keywords, operators, data types, and type-wrapper classes
* Control structures
* Methods and method overloading
* Arrays and strings (each are Java objects)
* Basic object-oriented programming concepts including objects, classes, encapsulation, inheritance and software reuse
* Advanced object-oriented programming concepts including interfaces, polymorphism, abstract classes and abstract methods
* Static methods and class variables
* Packages
* Errors and exception handling
* Multithreading

Optional extra topics:
* Basic GUI components from the Swing Set of GUI components including buttons, labels, check boxes, radio buttons, choice buttons, text fields and lists
* Advanced GUI components from the Swing Set of GUI components including panels, frames, text areas, sliders, and many more
* Controlling screen layouts with layout managers
* Event-driven GUI programming

A detailed list of topics follows; The number and depth of the topics covered may vary.

Java for non-programmers

1 - How to set-up
2 - running java
3 - syntax intro
4 - show message.

Java syntax, what is a computer?, phases of programming

Course Objective: To show the basics of computing and programming
Outcome: Students can write and run hello world.
Outcome: Students can display a dialog box, for output
Outcome: Students can use the IDE.
Outcome: Students understand the basic parts of a computer.

Java Data Types

1 - Reserved Words, Case Sensitivity
2 – Reference vs. Primitive Data Types
a – signed, vs. unsigned
b – fixed, vs, floating
c – 8 primitive data types

Operators and Applets

Operators, Applets, graphics context, import, getDouble.

Control Structures

While, sentinal-controlled repitition, unary ops, for, decimal text, switch, if, do-while, break, continue
Outcome: Set-up and solve problems requiring iteration

Wrapper Classes Arrays and Vectors

1 – Methods in their Wrapper Classes,
2 – Casting
3 – Arrays – their construction, passing variables
4 – Vector Class and its methods

Goals: To provide the student with the means to manipulate data within his or her program, converting from one data type to another. To also provide the student with a means of grouping the data into the containers, arrays and vectors.
Outcome: The student will be able to write Java programs containing large amounts of data in an efficient manner. He will also be able to increase and demonstrate the efficiency, in terms of lines-of-code, of O-O programming using the containers.

Outcome: The student will be able to write a Java program that demonstrates a class hierarchy, inheritance, and a simple degree of polymorphism.

Java Basics
1 – Classes, Overloaded Methods, Constructors
2 – Getter and Setter Methods, Casting
3 – Reserved Words, null and super
4 – Modifier Static (methods & variables)
5 – Modifier Abstract (Classes & methods)

Strings and Processing

1 – String Class
2 – String constructors
3 – String methods
4 – StringBufferClass
5 – StringTokenizerClass

Goals: To provide the student with the means to work with text-type data in Java programs.
Outcome: The student will be able to create, read, and parse text using Java programs.

This and Interfaces

1 – Reserved Word, this
2 – Interfaces (implementing in classes and extending other interfaces, more inheritance & polymorphism).
3 – Summary

Goals: To teach the student the concepts of static and abstract objects, and of interfaces, important tools in Java for Object-Oriented programming.
Outcome: The student will be able to write a program that demonstrates the use of static variables and interfaces in object-oriented programming, using a customer billing problem.

Packages and Visibility

1. packages
2. Modifier Visibility, Class Scope
3. Inner Classes (and again class scope)

Goals: To provide the student with the Java concepts that give object-oriented programming its modularity and extendibility benefits.
Outcome: The student will be able to control the scope of a variable or method. He will be able to use the concept of Abstract Methods and Interfaces to amplify and further demonstrate O-O inheritance and polymorphism.

Introduction to JavaDoc

a – Introduction
b – JavaDoc tags
c – JavaDoc in CodeWarrior
d - Summary

Goals: To introduce basic elements of the Java language, which permit the writing and documentation of simple programs.
Outcome: The student will be capable of documenting the program code using JavaDoc.

Exceptions

1 – The try block
2 – throw and throws
3 – The catch block
4 – The finally block

Goals: To expose and explain Java exception handling to the students.
Outcome: The student will be able to create exception handling in Java programs, and he or she will be able to respond to requests for exception handling when using someone else’s code.

Graphical User Interface Components and Listeners

1 – GUI Components: Frames, Labels, Buttons
2 – Event Listeners: WindowListener, ActionListener

Goals: To give the student the components necessary to create computer screen displays and to respond to events occurring on the screen.
Outcome: The student will be able to place GUI components onto a computer screen and to respond to events such as a mouse click or the push of a button.

Container Layouts: FlowLayout, GridLayout

Goals: To provide the means of arranging and manipulating GUI components in desired patterns on the computer screen.

Outcome: The student will be able to design functional and aesthetic displays on the computer screen using Java classes and using CodeWarrior.

File I\O, input & output streams

1 – Class hierarchy of java.io
2 – Input Stream, OutputStream
3 – Reader, Writer

Goals: To teach the student the techniques used in Java to transfer data to and from computer files.
Outcome: The student will be able to create files and read from computer files using Java. In particular, he or she will be able to create or read from files in the ASCI text format.

Additional References:

1 – Sun's Application Programming Interface, available in CodeWarrior.
2 – *Java How to Program*, Deitel & Deitel, Prentice Hall, Third Edition, 1999.
3 – *The Java Class Libraries*, Chan and Lee, Addison Wesley (c) 1997.
4 – *Java Source Book*, Ed Anuff, The John Wiley and Sons, Inc., First Edition, 1996.
5 – *The Java Programming Language*, K. Arnold & J. Gosling, Addison Wesley, Second Edition.

Grading Policy:

Homework and Laboratory Trials	1/3
Midterm Exam	1/3
Final Exam	1/3

Assignments are due at the beginning of class. Assignments handed in during class lose 5 points, after class 10 points. Late submittals lose 10 points per day including weekends and holidays. Missing a test results in a zero unless a written excuse is presented.

Homework requirements:

Print out a listing of the program. Print out the program intput and output. You may need to do this at various levels of detail. Hand in a labeled disk with a printout. Place the disk in a #10 letter envelope and staple the envelope to the printout.