Course Objectives

- Java as a language for OO program implementation.
- Unified Modeling Language (UML) diagrams, an industry standard.
- Object-Oriented Programming (OOP) including...
- To introduce...
- From the functional paradigm to the object-oriented paradigm...
- To transition...
sorting
fundamental algorithms on data structures, such as searching and
lists, stacks, queues, trees, tables, etc.
The formulation and implementation of basic data structures, such as
visitor.
behavioral: command, interpreter, iterator, state, strategy,
structural: adapter, composite, and decorator.
creational: factory and singleton.
common design patterns for various purposes. ..

To introduce (cont.) ..

Course Objectives (cont.)
Course Objectives (cont.)

- Both stream and event-driven I/O.
- Big O notation.
- The rudiments of complexity analysis, such as asymptotic behavior and

To introduce (cont.)
A superclass.

Specifically, it can inherit variables or methods from its parent, or

A class can be derived from, or be an extension of, another class.

A class is a blueprint or prototype that defines the variables and methods

common to all objects of a certain kind.

Objects are said to interact through messages. A message is directed at

The sender knows nothing of how the method is performed.

an object and specifies the method that the object should perform on

Objects are said to interact through messages. A message is directed at

A method is an operation on the object.

an object may model, or be an abstraction of, a real-world object. It is

a collection of fields (variables) and methods.
more specialized its purpose.

In general, the further down in the hierarchy that a class appears, the
desired.

The inheritance tree, or class hierarchy, can be as broad or deep as

subclass are spelled out.

— Only the additional or replacement variables and methods of the
subclass.

— Variables and methods of the superclass are transparently provided to

Simplifies the reuse of existing software.

Inheritance
Exception handling * 
the singleton pattern, the visitor pattern, the interpreter pattern, 
static fields, static methods, *
functional lists and the composite pattern, *
abstract classes, interfaces, inheritance, polymorphism, *
UML diagrams, *
classes, fields, methods, constructors, *
primitive types, String type, *
Transition from functional to object-oriented programming.

The list of topics in (approximate) order:

Schedule
complexity analysis

hashing

priority queues

template pattern, iterator pattern

searching and sorting, quick sort, merge sort, heap sort, bucket sort,

arrays

adapter pattern, decorator pattern

binary trees, binary search trees, self-balancing trees,

mutable lists and the state pattern, stacks, queues,

– Common data structures:

(MVC) paradigm

– Strategy pattern, command pattern, the Model-View-Controller

– Graphical User Interface and Event-driven Programming:

Schedule (cont.)
Course Mechanics

- Administrative
- Assignments
- Tutorials
- On-line lecture notes

Contains

http://www.compmec.rice.edu/~comp212

Course Web site

January 17, 2001
Three equally weighted exams (45%) •

Assignments (10%) • Occasional quizzes during the tutorials and minor programming early.

- There will be graded milestones to help diagnose and correct problems.

- Expect three.

Major programming assignments (45%) •

Assignments
http://www-content.rice.edu/~comp212/01-spring/labs/

• Sign up for a tutorial section at the course web site:

First Assignment