COURSE OUTLINE

COS113	Object Oriented Programming
Course Number	Course Title
4	3 Lecture/2 Laboratory
Credits	Hours: lecture/laboratory/other (specify)

Catalog description:

Teaches the concepts of object-oriented programming. Topics include data abstraction, encapsulation, hierarchy via composition and derivation (inheritance), and polymorphism. Java used for lab work. Computer Science majors planning to transfer should take this course in order to meet transfer requirements. Spring offering.

Prerequisites: COS102 or permission of department

Corequisites: none

Required texts/other materials:

Just Java™ 2, 6th Edition, Peter van der Linden. Prentice Hall

Last revised: Spring 2005

Course coordinator: M.Hayes / D. Bostain

Information resources: Instructor's website: http://cos231.drbco.com/

Course goals:

The successful student will understand and apply the following concepts:

- Java Application: Java classes, programming concepts and syntax rules, building and testing Java applications
- Object Modeling: Modeling real-world problems, Classification and Functionality, Terminology
- Data: Mapping real world objects to Java data, Java built-in types and data statements, simple operations and process statements
- Control Flow: Mapping real world processes to Java algorithms, statement execution and sequences, decisions and repetitions
- Using Java Classes: Importing classes, creating and using objects, example classes
- Arrays and Strings: Creating and using arrays, including arrays of objects, creating and using Strings
- Developing Java Classes: Design and development, the class definition, operation implementation, internal data

Unit 1

At the completion of Unit 1 the student will be able to discuss the following three major topics:

Introduction to the Java language, history and main paradigms

1. History
2. Main aspects of Java
3. Basic paradigm and organization
4. Scope strengths and weaknesses

Compiling Environment, Books, References

1. Compiler/Interpreter/Applet viewer
2. Java books, references, URLs
3. Creating and running a Java application

Differences between Java and C++

1. Program Structure and Environment
2. The Name Space: Packages, Classes and Members
3. Comments
4. No Preprocessor
5. Unicode and Character Escapes
6. Primitive Data Types
7. Reference Data Types
8. Objects
9. Arrays
10. Strings
11. Operators
12. Statements
13. Exceptions and Exception Handling
14. Miscellaneous Differences

Unit 2

At the completion of Unit 2 the student will be able to discuss and apply the following three programming design topics:

Fundamental Programming Structures in Java

1. Required static main() method
2. Data types: integers, floats, char, boolean
3. Variables
4. Assignments, initializations, and constants
5. Operators
6. Strings
7. Control Flow

Classes and Objects in Java

1. Classes, Objects
2. Constructors
3. Method overloading and multiple constructors
4. The **this** pointer
5. The **static** keyword: class variables and class methods, examples
6. Object destruction, automatic garbage collection, and **finalize()**
7. OO code design strategies

Java Object Design and Programming

1. Review of the main OOD features.
2. Classes, attributes, methods, access control, class members
3. Object creation, members, references
4. Inheritance, polymorphism, interfaces
5. Associations, links, association classes, propagation
6. Aggregations
7. Ternary and n-ary associations
8. Multiplicity design rules for associations and association classes

Unit 3

At the completion of Unit 3 the student will be able to discuss and apply the following three Object-Oriented concepts:

Java Special OO Features

1. Inner classes
2. Anonymous classes
3. Reflections
4. Packages

Java Interfaces

1. What interfaces are
2. Why they are important
3. Syntax
4. When to use them
5. Comparison with multiple inheritance
6. Interfaces are data types
7. Interface implementation
8. Interface constants
9. Extending interfaces
10. Marker interfaces

Java Exception Handling

1. Overview and philosophy
2. Exception Class Hierarchy
3. Catching an exception: try and catch blocks
4. Methods which throw exceptions - the throws clause
5. Compiler support for exception handling: the "catch, throw requirement"
6. After catching an exception - finally blocks
7. Programmatically throwing an exception - the throw statement
8. User defined exceptions

Unit 4

At the completion of Unit 4 the student will be able to apply the following two programming concepts:

Java Threads

1. Threads and their importance: GUI speed-up, flexibility, simplicity, application optimization
2. Constructors, start(), stop(), sleep(), sus-pend(), resume(), isAlive()
3. Creating and launching a Thread in a Java application
4. Features of Java Threads
5. Threads sharing resources: synchronized methods
6. Controlling thread interaction: the wait() and notify() methods

Graphics Programming and Printing

1. Frames and windows
2. Displaying in a window
3. Basic Events, update() and paint()
4. Text and font, colors
5. Drawing and filling shapes
6. Paint mode
7. Images, buffering, image acquisition
8. Printing

Unit 5

At the completion of Unit 5 the student will be able to apply the following two topics:

Java User-Interface, the Abstract Windowing Toolkit (AWT)

1. Historical perspective, 1.1 and 1.2 (Swing)
2. Overview of Java GUI Development
3. Creating a visual design
4. Creating the main Container class
5. Building the Components specified in the visual design
6. Laying out the Components using Layout Managers
7. Implementing event-handling methods
8. Adhering to the MVC-standard of GUI development
9. Component Inheritance Hierarchy and methods
10. Frames, Panels
11. In-depth study of Java Layout Managers: Flow, Grid, Grid Bag, Border
12. Using multiple Layout Managers

AWT Event Handling

1. Overview, philosophy, and architecture
2. Event classes and event listeners
3. Using the AWTEvent/Component/Listener
4. Using Adapters
5. GUI architecture based on the Model-View-Controller design pattern (MVC)
6. Applying the MVC design pattern
7. Semantic Events and Low-level Events

Unit 6

At the completion of Unit 6 the student will be able to discuss and apply the following two major topics:

Applets

1. Embedding an application in a browser
2. Applet's key methods: init(), start(), paint(), stop(), destroy()
3. Creating an Applet and dragging it to a browser
4. HTML tags and attributes for Applets
5. Differences between Applications and Applets: security, programming, audio & video
6. Transforming an Application into an Applet
7. Working with Threads in Applets

Lightweight Components

1. The AWT heavyweight reality: native peers and opaque windows versus low-resource and flexible lightweight components
2. Lightweight containers
3. Lightweight components and Z-order
4. Lightweights and their graphics
5. Lightweights and preferred sizes

Unit 7

At the completion of Unit 7 the student will be able to discuss and apply the following two major topics:

Swing

1. Swing versus AWT
2. Swing as a powerful extension of AWT
3. Main Features and basics
4. Event handling
5. Built-in MVC architecture and the Model-delegate Pattern
6. Containers
7. Visual Components
8. Content Pane and default layout manager
9. Applets and applications
10. Labels, Buttons, Button Hierarchy
11. Check Boxes
12. Text components
13. Lists, Scroll Panes, Combo Boxes
14. Dialogs, Option Panes, Menus, Actions
15. Tables
16. Lightweight versus Heavyweight components
17. Mixing Swing & AWT
18. Swing & Threads

Java Input & Output

1. Streams
2. ZIP file stream
3. Delimiters and tokens
4. Object streams; object serialization
5. Object references
6. Security and versioning

Unit 8

At the completion of Unit 8 the student will be able to discuss and apply the following two major topics:

Advanced AWT

1. Java image manipulation
2. Java color models
3. Image filters
4. Memory image sources
5. Pixel grabbing
6. Data transfer

Java Beans for GUI

1. What Beans have to offer
2. How to write a Bean
3. The BDK and BeanBox
4. Building an image viewer application with Beans
5. Design pattern for Bean properties and events
6. Writing Bean properties
7. Adding custom Bean events
8. Property editors
9. Customizers
10. Advanced use of Introspection
11. Beyond GUI Beans: Beans for entire applications-- the component-based application development strategy

Unit 9

At the completion of Unit 9 the student will be able to discuss the following major topics:

JDBC: Database Connectivity

1. Why JDBC: importance and competition
2. JDBC overall design, communication path, and typical uses
3. SQL
4. Installing JDBC
5. Basic JDBC programming concepts
6. Database URLs, connecting, querying
7. Populating a database and executing queries
8. Metadata

Java Networking

1. Networking Basics:
   - What is Networking?
   - Java support for Networking
   - Client/Server Networking
   - IP Addresses
   - Example & exercise
   - Ports
   - Brief Introduction to CGI
2. Working with URLs:
   - What is a URL?
   - The Java URL class
   - The Java URLConnection class
3. Working with Sockets
   - What is a Socket?
   - Creating a Client program with the Socket Class
   - Creating a Server Program
4. Security Issues with Networking and Applets

Unit 10

At the completion of Unit 10 the student will be able to discuss the following concepts:

Java Servlets

1. Servlet Basics
2. Brief Introduction to HTTP and HTML
3. Java Support for Servlets: The Servlet API
    - Servlet
    - GenericServlet
    - HttpServlets
    - ServletRequest
    - HTTP ServletRequest
    - ServletResponse
    - HTTP ServletResponse
4. Client Interaction: Handling HTTP Requests
5. Servlet Life Cycle
6. Saving Client State: Session Tracking

Evaluation of student learning:

Specific methods for evaluating a students progress through the course is up to the discretion of the instructor. Below is a suggested format which balances the lab and lecture components.

Four one-hour exams	=	60%	of the grade
One comprehensive final exam	=	20%	of the grade
The average of seven lab projects	=	20%	of the grade
		100%	

Academic Integrity Statement:

Mercer County Community College is committed to Academic Integrity -- the honest, fair and continuing pursuit of knowledge, free from fraud or deception. This implies that students are expected to be responsible for their own work and that faculty and academic support services staff members will take reasonable precautions to prevent the opportunity for academic dishonesty.

See http://www.mccc.edu/admissions_policies_integrity.shtml for a complete explanation of policies and procedures regarding academic integrity and academic integrity violations.