

OBJECT-ORIENTED PROGRAMMING

LESSON PLAN and STUDENT WORKBOOK

**Richard Conn, Instructor
University of Cincinnati
Department of Electrical and Computer Engineering
Course Number 20-260-635-901**

Contents

1. Software and Software Engineering	3
2. Design Methodologies and Graphical Notation	4
3. The Object Model	5
4. Object-Based Programming Languages - Ada	6
5. Object-Oriented Programming Languages - C++	7
6. Class Project	8

**Object-Oriented Programming
Lesson Plan and Student Workbook**

This Lesson Plan and Student Workbook gives the reading assignments, homework assignments (homework problems and papers), and study guides for this course. If you miss any classes, you can catch up by following this Lesson Plan and Student Workbook.

Reading Assignments, Course Text, and Lecture Notes

All reading assignments are from the course text and the course notes. The course text is:

Grady Booch, *Object-Oriented Design with Applications*, Benjamin-Cummings
Publishing Company, 1991, ISBN 0-8053-0091-0

The course notes were prepared by Richard Conn and are available at Kinko's. Ask for them by course number.

Each section of the Lesson Plan and Student Workbook includes a homework assignment, which is to be turned in for a grade, and a series of study guide questions.

Homework

Place your name and the identification number, e.g., 4-1, on the first page of your homework assignment or paper.

Schedule, Grading, and Materials List

The Cover pages to the Course Notes contain a copy of this information. In addition, each student is given a copy of these Cover pages and this Lesson Plan and Student Workbook on the first day of class.

**Object-Oriented Programming
Lesson Plan and Student Workbook**

1. Software and Software Engineering

Objectives of this Module

- Present and discuss the idea that software is much more than just code -- *engineered* software is composed of *controlled configuration items* which include documents, data, and code
- Present and discuss the history of software development, including its evolution into a business
- Present and discuss many of the problems with doing software development, particularly when there is more than one person involved
- Present and discuss several different software engineering paradigms, showing different methods for developing engineered software:
 - Classic "waterfall" method
 - Rapid prototyping
 - Spiral method
- Present and discuss some of the technologies used in the support of software engineering:
 - *Computer-Aided Software Engineering* (CASE)
 - Ada programming language
 - C++ programming language
- Introduce the concepts of complexity, Object-Oriented Requirements Analysis (OORA), and Object-Oriented Design (OOD)

Reading Assignment

Course Notes, *Module 1*
Booch, *Chapter 1*

Homework Assignments Due at the Beginning of the Next Module

Course Notes, first half of *Workbook on Ada Programming*

Study Guide

- ✓ What is *software engineering*?
- ✓ What is *software* from the perspective of Software Engineering?
- ✓ Explain the failure curve for software. Why is it so different from the failure curve for hardware?
- ✓ What are the various *software configuration items* that compose an item of engineered software? When are they usually produced during the classical software engineering processes?
- ✓ Explain the five aspects identified in the course of the industrial view of software.
- ✓ What are the phases of the classic "waterfall" model of the software engineering development process? What are the problems with this model?
- ✓ What are the phases of the prototyping model of the software engineering development process?
- ✓ What are the phases of the Spiral Model of the software engineering development process? How do other models fit into the Spiral Model?
- ✓ What three phases of activity are common to all software engineering development processes?
- ✓ Describe the *Capability Maturity Model for Software*.
- ✓ Why was the Ada language developed? What are some of its key features which help it meet its goals?
- ✓ Why was the C++ language developed? What are some of its key features which help it meet its goals?
- ✓ What is *industrial-strength software*? What are its attributes?
- ✓ Explain the canonical form of a complex system.
- ✓ What are the purpose and elements of the design of a complex system?

2. Design Methodologies and Graphical Notation

Objectives of this Module

- Present and discuss many of the common diagram notations used during requirements analysis and design:
 - Data Flow Diagrams (DFD's)
 - Function Diagrams
 - State Transition Diagrams (STD's)
 - Entity Relationship Diagrams (ERD's)
 - Object Interaction Diagrams (OID's)
 - Booch Diagrams
- Present and discuss the concept of the Data Dictionary and its content
- Present and discuss several common design methodologies
 - Data Flow-Oriented Design
 - Data Structure-Oriented Design
 - Object-Oriented Design
 - Real-Time Design

Reading Assignment

Course Notes, *Module 2*

Booch, *Chapter 5*

Homework Assignments Due at the Beginning of the Next Module

Course Notes, second half of *Workbook on Ada Programming*

Study Guide

- ✓ What do *Data Flow Diagrams* tell us? What are the elements of a DFD?
- ✓ What does a *Data Dictionary* tell us?
- ✓ What do *Function Diagrams* tell us? What are the elements of a Function Diagram?
- ✓ What do *State Transition Diagrams* tell us? What are the elements of an STD?
- ✓ What do *Entity Relationship Diagrams* tell us? What are the elements of an ERD?
- ✓ What do *Object Interaction Diagrams* tell us? What are the elements of an OID?
- ✓ What do *Booch Diagrams* tell us? What are the elements of a Booch Diagram?
- ✓ Compare and contrast the Data Flow-Oriented Design, Object-Oriented Design, and Real-Time Design methodologies.

3. The Object Model

Objectives of this Module

- Present and discuss the concept of the Object Model and its evolution
- Introduce the terms Object-Oriented Analysis (OOA), Object-Oriented Requirements Analysis (OORA), Object-Oriented Design (OOD), and Object-Oriented Programming (OOP)
- Present and discuss the elements of the Object Model
- Present and discuss the benefits and applications of the Object Model, particularly in the creation of industrial-strength software
- Obtain a more complete understanding of classes and objects, the basic building blocks of software systems developed using object-oriented methods
- Present and discuss the nature of an object
- Present and discuss the concept of relationships between objects
- Present and discuss the nature of a class
- Present and discuss the concept of relationships between classes
- Present and discuss relationships between classes and objects and the role of classes and objects in object-oriented design
- Present and discuss the concept of building classes and objects (abstractions) with quality in mind, including heuristics for choosing operations, relationships, and implementations of classes and objects

Reading Assignment

Booch, *Chapters 2, 3, and 4*

Homework Assignments Due at the Beginning of the Next Module

Course Notes, *Workbook on C++ Programming*

Study Guide

- ✓ How do the different generations of programming languages differ in a topological sense?
- ✓ What is an *object*?
- ✓ What is meant by the terms *Object-Oriented Programming*, *Object-Oriented Design*, and *Object-Oriented Analysis*? How are these activities related?
- ✓ Explain why no single programming paradigm is best for all kinds of applications.
- ✓ What is the *Object Model*? What are the elements of the *Object Model*? Be able to go into detail on these topics.

4. Object-Based Programming Languages - Ada

Objectives of this Module

- Examine the features of Ada in terms of the following:
 - Encapsulation mechanisms
 - Modularity support
 - Abstraction mechanisms
 - Data typing
- Present and discuss the concept of declaring classes in Ada
- Present and discuss the concept of member data and functions in Ada
- Present and discuss an overview of the features of Ada

Reading Assignment

Course Notes, Lecture Notes

Course Notes, *Workbook on Ada Programming*

Homework Assignments Due at the Beginning of the Next Module

None

Study Guide

- ✓ What are the four program units of Ada?
- ✓ Which of the four program units of Ada are used as the mainline?
- ✓ What program units in Ada are used to declare object classes?
- ✓ Be able to write a simple program in Ada.

5. Object-Oriented Programming Languages - C++

Objectives of this Module

- Examine the features of C++ in terms of the following:
 - Encapsulation mechanisms
 - Modularity support
 - Abstraction mechanisms
 - Data typing
- Present and discuss the concept of declaring classes in C++
- Present and discuss the concept of member data and functions in C++
- Present and discuss an overview of the features of C++

Reading Assignment

Course Notes, Lecture Notes

Course Notes, *Workbook on C++ Programming*

Homework Assignments Due at the Beginning of the Next Module

None

Study Guide

- ✓ What are the three program units of C++?
- ✓ What program unit in C++ is used to declare object classes?
- ✓ Be able to write a simple program in C++.

*Object-Oriented Programming
Lesson Plan and Student Workbook*

6. Class Project

The class project is described in the Software Development Plan. This plan includes a timeline with milestones (due dates) for various elements of the project. Each time a milestone is reached, an IV&V Report is also due.