Advanced Object-Oriented Design
SCOPE AND SEQUENCE

I. Object Oriented Modeling [9 weeks]

1. Object-Oriented Modeling

OBJECTIVES – The student will be able to:

1.1. Explain what is meant by object-oriented modeling

1.2. Understand the role of the nouns and verbs in a problem description in guiding object-oriented modeling

1.3. Apply object-oriented modeling techniques to the problem of creating an interactive battleship game

2. Review of the Java language

OBJECTIVES – The student will be able to:

2.1. Create a class with instance variables and methods
2.2. Test a java method using a driver
2.3. Utilize unidimensional arrays in a java program
2.4. Utilize multidimensional arrays in a java program
2.5. Write a java method using loops and conditionals
2.6. Throw and catch exceptions in Java
2.7. Write an interactive program using Java classes
2.8. Do calculations within a Java program
2.9. Create a test plan for a Java program

ASSIGNMENT: Battleship assignments 1-8, Test plan assignment

II. Object-Oriented Modeling of Intelligent Agents [3 weeks]

3. Creating Intelligent Agents

OBJECTIVES – The student will be able to:

3.1. Explain how an intelligent agent can be modeled using inheritance

3.2. Create an intelligent agent to play the game of Battleship

3.3. Use scientific optimization methods to improve the performance of an intelligent agent.

ASSIGNMENT: Battleship assignment 9

III. Computer Graphics using Java [3 weeks]

4. Computer Graphics

OBJECTIVES – The student will be able to:

4.1. Create a Java program that inputs text using a text input field.
4.2. Create a Java program that outputs a text label onto a graphical window.
4.3. Create a Java program that draws rectangular shapes in a graphics window.
4.4. Create a Java program that creates circular arcs.
4.5. Create a Java program that can respond to action events in a window.
4.6. Create a Java program with multiple graphics frames.
4.7. Create a Java program that inputs data via a checkbox
4.8. Create a Java program that inputs data via a menu
4.9. Create a Java program that presents buttons to a user.
4.10. Create a Java program that senses and responds to mouse events.

ASSIGNMENT: Java Graphics Assignments

IV. Inheritance Hierarchies [3 weeks]

5. Universal Modeling Language

OBJECTIVES – The student will be able to:

5.1. Identify parts of a UML class hierarchy.
5.2. Use UML to diagram a class hierarchy of mathematical relationships.

ASSIGNMENT: UML assignment

6. Inheritance Hierarchies

OBJECTIVES – The student will be able to:

6.1. Create a function graphing program that enables inheritance.
6.2. Progressively extend the capabilities of a function graphic program using inheritance.

ASSIGNMENTS: Function grapher assignments, tic tac toe assignment.

V. The Group Software Development Process [9 weeks]

7. Software Engineering

OBJECTIVES – The student will be able to:

7.1. Define the term *Software Engineering.*

7.2. Identify the goals of Software Engineering
7.3. Identify the steps of the Software Engineering process
7.4. Describe the activities involved with each step of the Software Engineering process
7.5. Identify the products created during each step of the Software Engineering process

8. Software Development Methodologies

OBJECTIVES – The student will be able to:

8.1. Define the term *software development methodology*.
8.2. Identify widely used methodologies.
8.3. Identify the characteristics of projects that indicate the use of specific methodologies.
8.4. Define the term *agile methodology*.

9. Commercial Software Development

OBJECTIVES – The student will be able to:

9.1. Define the following terms:

- RFP
- Proposal
- Contract
- Deliverables
- Reviews
- Sign-off

9.2. Explain how the following roles fit into the commercial software development process:

- Project Manager
- Lead Engineer
- System Engineer
- Programmer
- Tester
- Technical Writer
- Configuration Management
- Trainer

9.3. Define the term *leadership style*
9.4. Identify the major leadership pitfalls and how to avoid them.

10. Software Schedules

OBJECTIVES – The student will be able to:

10.1. Define the following terms:

- Task
- Milestone

- Resources

10.2. Identify five reasons why software projects do not meet their schedules and explain how to avoid them.

11. Interactive Software

OBJECTIVES – The student will be able to:

11.1. Define the following terms:

- User Interface
- Front End
- Back End
- Command Line Interface
- Menu Interface
- Direct Manipulation Interface
- Icons
- WYSIWYG
- Robust
- Transparent
- The ten-minute rule

11.2. Identify the three main types of user interfaces and when it is appropriate to use them.

11.3. Identify three characteristics of effective user interfaces.

11.4. Identify five common pitfalls of user interface design.

12. Developing a Group Software Project

OBJECTIVES – The student will be able to:

12.1. Contribute effectively to a software development team.

12.2. As part of a software development team, deliver commercial-quality software on schedule, through all steps of the Software Engineering process.

12.3. Contribute to the effective leadership of a software development team.

ASSIGNMENT: Group software project

VI. Pair Programming [6 weeks]

13. Pair Programming

OBJECTIVES – The student will be able to:

13.1. Explain the difference between the pilot and navigator roles, and perform each effectively.

13.2. Specify, design, develop, test, and deliver a commercial-quality graphical software game

ASSIGNMENT: Pair project

VII. Software Maintenance [3 weeks]

14. Software Maintenance

OBJECTIVES – The student will be able to:

13.1. Identify the special problems facing a software maintenance team.

13.2. Plan, implement, and deliver an enhancement release for a software system most of whose developers are not available for questions.

ASSIGNMENT: Maintenance project