

Bemidji State University

CS 3360: Object-Oriented Software Development

A. COURSE DESCRIPTION

Credits: 3

Lecture Hours/Week: *.*

Lab Hours/Week: *.*

OJT Hours/Week: *.*

Prerequisites: None

Corequisites: None

MnTC Goals: None

Techniques used in object-oriented software development. Key components of these techniques include design patterns, abstraction, encapsulation, modularity, message passing, polymorphism, inheritance, and incremental software development. Students translate a design into software using an object-oriented programming language. Additional topics may include applets, markup languages, multi-threaded programming, and rudimentary network programming. Prerequisite: CS 2322 or equivalent. May not be offered every year.

B. COURSE EFFECTIVE DATES: 08/08/2002 - Present

C. OUTLINE OF MAJOR CONTENT AREAS

1. Design Patterns
2. Abstractions
3. Encapsulation
4. Modularity
5. Message Passing
6. Polymorphism
7. Applets
8. Automated Testing
9. Incremental Software Development
10. Inheritance
11. Java & Javadoc
12. Rational Unified Process

D. LEARNING OUTCOMES (General)

1. demonstrate an understanding of, and how to implement key design patterns: singleton, template, strategy, iterator, factory, composite and decorator.
2. develop an in depth understanding of object oriented design given a problem description.
3. know how to develop automated testing of object-oriented components.
4. know how to implement all of the covered design patterns.
5. understand key aspects of object-oriented design: abstraction, encapsulation, modularity, message passing, polymorphism, inheritance.
6. understand the basics of using UML in RUP design: Object and classes, relationships, state and sequence diagrams.

E. Minnesota Transfer Curriculum Goal Area(s) and Competencies

None

F. LEARNER OUTCOMES ASSESSMENT

As noted on course syllabus

G. SPECIAL INFORMATION

None noted