

**Course number and name:** **CS 04114: Object Oriented Programming and Data Abstraction**  
**Credits and contact hours:** 3 credits / 3 contact hours  
**Course Coordinator:** Jack Myers  
**Instructional Materials:** Objects First with Java – A Practical Introduction Using BlueJ, David J. Barnes & Michael Kölling, 2017.

### Specific course information

**Catalog description:** Objects and data abstraction continues from Introduction to Object-Oriented Programming to the methodology of programming from an object-oriented perspective. Through the study of object design, this course introduces software engineering and focuses on file I/O, function prototypes, exception handling, decoupling strategies, and other advanced topics.

**Prerequisites:** CS 04113 Introduction to Object Oriented Programming *or* CS 04103 Computer Science & Programming

**Type of Course:**  Required     Elective     Selected Elective

### Educational objectives for the course

1. **abstraction techniques.** Students have successfully utilized abstraction techniques to design abstract data models (e.g., abstract classes / interfaces and decoupling strategies) that model real world behavior in highly extensible ways.
  - ABET (1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
  - ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
  - ABET (6) Apply computer science theory and software development fundamentals to produce computing-based solutions.
2. **advanced Java programming.** Students have demonstrated mastery of advanced Java programming (e.g., GUI development, exceptions and error handling, File I/O, lambda expressions) through coding assignments and tests on such concepts.
  - ABET (1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
  - ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. **uml diagramming.** Students have created UML class diagrams and sequence diagrams to document the design of their applications.

- ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

### **Required list of topics to be covered**

1. Abstract classes
2. Interfaces
3. Decoupling strategies
4. GUI development
5. Exception handling and use of related APIs
6. File access, buffered I/O, and use of related APIs
7. Lambda expressions, streams (sequential and parallel execution), and use of related APIs
8. UML Class and Sequence diagrams
9. Introduction to information management (e.g., graph database)

### Optional list of topics that could be covered

1. Design patterns
2. Recursion
3. Abstract data types\* (\*some exposure to this topic strongly desired)