

**Brooklyn College**  
**Department of Computer and Information Science**

**CISC 3150 [26] Object-Oriented Programming**

3 hours lecture; 3 credits

Use of inheritance and polymorphism. Advanced object-oriented programming techniques.  
Introduction to object-oriented design.

**Objectives**

By the end of this course, students will be able to:

1. Understand and apply the concepts of class, object, instantiation and methods.
2. Understand and apply the concepts of data abstraction, inheritance (single and multiple) and polymorphism.
3. Write applications in an object-oriented language.
4. Understand and apply the concepts of run-time exceptions, file I/O, multi-threading and event-driven programming.
5. Evaluate the appropriateness of a specified class hierarchy for a given task.

**Syllabus**

1. Introduction (2 hours/ 2 hours)
  - a. Object-oriented programming vs. procedural programming
  - b. Basic terminology: class, object, instance, method, message
2. The life cycle of an object-oriented program (2 hours / 4 hours)
  - a. Objects in execution
  - b. Message passing and method invocation
  - c. Object creation and destruction
3. Introduction to Java (2 hours / 6 hours)
  - a. Features in common with C
  - b. Primitive type model
  - c. Cross-platform issues
4. Java support for object-orientation (3 hours / 9 hours)
  - a. Reference types and objects
  - b. Constructors
  - c. Class definition
    - i. Instance variables
    - ii. Method definition
    - iii. Data access
5. Other Programming Features (2 hours / 11 hours)
  - a. Final and static instance variables
  - b. Static methods
  - c. Arrays, Vectors and basic iteration
6. Inheritance I (3 hours / 14 hours)
  - a. Extending functionality
  - b. Composition
  - c. Inheritance: subclass, superclass, protected access, upcasting
7. Polymorphism (3 hours / 17 hours)

- a. Dynamic vs. static types and binding
- b. Method overriding
- c. Downcasting
- d. Class hierarchies
- 8. Inheritance II (4 hours / 21 hours)
  - a. Inheritance vs. composition
  - b. Factoring out common functionality through inheritance
  - c. Abstract methods and classes
  - d. Interfaces
- 9. Basic applet programming (6 hours / 27 hours)
  - a. Introduction to the AWT
  - b. The applet execution environment
  - c. Events and event handling
  - d. Display maintenance
- 10. Advanced programming techniques (8 hours / 35 hours)
  - a. The Java class libraries
  - b. Collections and iteration
  - c. Elimination of type-testing using polymorphism and subclassing
  - d. Application frameworks
  - e. Introduction to network and client-server programming
- 11. Introduction to Object-Oriented Design (3 hours / 38 hours)
  - a. Responsibility-driven programming
  - b. CRC (class-responsibility collaborator) cards
  - c. Basic UML concepts
- 12. Model/View/Controller - Document/View (4 hours / 42 hours)
  - a. The problem with application / display coupling
  - b. Separating the application and the view
  - c. The model / view/ controller and document/view paradigms
  - d. MVC applied to non-GUI applications
- 13. Exams (3 hours)

**Textbooks:**

Gosling, et. al. - *The Java Programming Language*  
Horstmann – *Core Java, Vols. I & II*