

COURSE DESCRIPTION

Department and Course Number	CS 202	Course Coordinator	Jiang Guo
Course Title	Introduction to Object-Oriented Programming	Total Credits	5

Current Catalog Description:

Algorithm development for Object Oriented Programming; designing, coding, and documenting programs. Laboratory activities on problem analysis and software development.

Textbook:

Deitel and Deitel. *Java How to Program, Sixth Edition*, Prentice Hall, 2004.

References:

Liang, Daniel., *Introduction to Java Programming, 4th Edition*, Prentice Hall, 2002.

Course Goals:

At the end of the course, students are able to

- Implement encapsulation and data hiding
- Use inheritance polymorphism
- Build GUIs that handle events
- Manipulate graphics context and objects
- Handle Java exceptions
- Create, read, and write files and streams
- Design recursive methods

These course goals contribute to the success of **Student Learning Outcomes 1.a, 1.d, 1.e, 5, and 6.**

Prerequisites by Topic:

Introduction to Programming

Major Topics Covered in the Course:

- Classes and Objects
- Inheritance
- Polymorphism
- GUI Components
- Graphics and Java 2D
- Exception handling
- Files and Streams

- String

Laboratory Projects (specify number of weeks on each):

Each week (except exam weeks) students complete a 3-hour lab project on a selected topic:

- Week 1: Design a program using classes
- Week 2: Practice with classes and inheritance
- Week 3: Build a program using class, inheritance, and polymorphism
- Week 4: Create a student information system with Java GUI
- Week 5: Exam
- Week 6: Implement a simple game with Java 2D
- Week 7: Design a student information system with files and streams
- Week 8: Design a student information system with strings
- Week 9-10: Finish a big project with GUI and files

Estimate Curriculum Category Content (Quarter Hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms	.75		Data Structures	.75	
Software Design	1.5		Prog. Languages	2.0	
Comp. Arch.					

Oral and Written Communications:

Written documentation of software built in labs and homework assignments.

Social and Ethical Issues:

No significant component.

Theoretical Content:

- Inheritance (1 week)
- Polymorphism (1 weeks)

Problem Analysis:

In the first part of the course, students learn the basic concepts of object-oriented programming, such as classes, inheritance and polymorphism. In the latter part of the course, students learn how to use OO skills to write programs based on the classes provided by Java library, such as GUI, Java 2D, and files packages.

Solution Design:

Solution design in this course mostly involves choosing appropriate OO techniques to accomplish certain programming objectives, such as creation of a class, an inheritance structure, and polymorphic functionality. The course also introduces students to the use of the GUI, file, and stream classes and to the use of exceptions when writing programs, especially big programs.