GRS CS 611:
Object Oriented Software Principles and Design (in Java)
Department of Computer Science

Boston University – Fall 2024

Description: An **advanced** course in Software Engineering with a specific focus on the design and development of software solutions using the Object Oriented Software paradigm.

This course is aimed on developing a deep understanding of the fundamental principles and techniques of object-oriented programming. Focus will be on the specification, programming, and analysis of reliable, and reusable software using object-oriented design principles. Topics include Classes, Objects and memory models, Abstraction, Inheritance and Polymorphism, Interfaces and Generic typing. Emphasis will be placed on using OO principles to design software against failure, and to design software for reusability through *well-known design patterns*.

All features of OO will be tied to Java specific implementations. We will study class definitions, designing a class inheritance structure, interfaces, generics, iterators and exception handling. A specific case study will be done to overlay the OO model on Java's built-in graphical user interface.

Prerequisites: Working knowledge of a programming language; Data Structures, SQL.

Goals and Objectives: The objective of the course is to provide the students with a strong foundation in object oriented principles and design, and to apply those skills in the context of an enterprise software development environment.

Instructor: Christine Papadakis-Kanaris, cpk@bu.edu, CGS 943
Teaching Assistants: Patrick Kuzdzal (pkuzdzal@bu.edu), **Alzahrani, Abdulaziz (zahrani@bu.edu), Junsun (Lucas) Yoon (lyoon02@bu.edu)**

Meeting Times and Places
Lecture: T/Th 2:00 p.m. – 3:15 p.m., CAS 522; **Labs:** as scheduled

Office Hours: Available through course Blackboard site

Course Materials

Textbooks:

- Program Development in Java, Abstraction, Specification, and Object-Oriented Design Barbara Liskov, John Guttag, Addison Wesley
- Design Patterns: Elements of Reuseable Object-Oriented Software Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides Addison-Wessley

Optional: Introduction to Java Programming: Daniel Liang, Prentice Hall
Research Papers: as provided

Course Resources Blackboard (https://learn.bu.edu) will be used as a repository for the lecture notes from each class, for homework, and for general announcements. Piazza.com will be our tool for questions and discussion.

Requirements and Grading

Independent Assignments & lecture quizzes	40%
Midterm Exam #1	35%
• Written (60%)	
• Practicum (40%)	
Final Project	25%

Independent Assignments: There will be (at least) three multi-part independent assignments throughout the course of the semester. The objective of each assignment is to put into practice the material and practices of OO design as presented in lecture. Your assignments will be graded in accordance to the mastery of the material as shown in practice. Completion of an assignment does not constitute mastery. Specifically, each assignment will be reviewed ***qualitatively*** on how well the principles of Object-Oriented design have been put in practice.

Quizzes: There may be several pop-quizzes throughout the semester, The number, content, and timing of the quizzes will not be pre-determined. Make-up quizzes will not be given, the lowest scored quiz will be dropped.

Exams: There will be one comprehensive midterm exam. The midterm exam will be in two parts and administered over two class periods: the first part will be an in class written exam focused on ensuring your understanding of Object Oriented Theory and Design. This part will account for 60% of your midterm grade. The second part will be an in class timed programming assignment. This part will account for 40% of your midterm grade.

Final Project: The course will culminate in a final project designed to **emulate a real-life development environment**. The objective for each student will be to investigate, understand and apply the software design process using object oriented constructs and techniques to a specific problem as presented by the instructor. Students will be randomly assigned to small development teams. Each team will be guided through the software development process, namely: analysis, specification and design, development, testing, and implementation. It is up to each team to use instructor resources to understand the system requirements.

Development teams will work independently and be encouraged to think creatively, but all teams must adhere to the problem objectives and specifications. Development will be done in the Java Programming language and the system developed is expected to provide an end to end solution, including use of local database and client graphical user interface. Students will be expected to ***formally*** present their design specification, project plan, and final solution.

Academic Misconduct: It is expected that you understand BU's Academic Conduct Code. The University and College take cheating very seriously. Cheating and plagiarism will not be tolerated in any course. Cases will be referred to the Dean's office and may result in loss of credit for an exam or assignment or other disciplinary action.

Please read the college's policy at: http://www.bu.edu/academics/policies/academic-conduct-code

Prohibited behaviors include:

- copying all or part of someone else's work, even if you subsequently modify it; this includes cases in which someone tells you what you should write for your solution,
- viewing all or part of someone else's work (with the exception of work that is being done by other members of your group),
- showing all or part of your work to someone other than a peer in your assigned group,
- consulting solutions with those found online or in books,
- posting your work where others (outside your group) can view it (e.g., online),
- **accessing solutions in a GitHub account of a prior student,**
- **use of ChatGPT or other on-line resources for solutions.**

The following schedule is a guide to the organization of the material and assignments throughout the semester, but is subject to change at the discretion of the instructor.

Week	**Weekly Topic Objective**	**Lecture Schedule**	**Assignment**
Week 0	Course Overview	Course Introduction and Overview: Evolution of Computers, Languages, and Software Design The Java Programming Environment and model	Assignment 0: Evaluation Assignment
Week 1	Understanding the Object Design Model: Applying the principles of *Abstraction* and *Encapsulation*. Case Study	Designing for scalability using object decomposition vs. algorithmic decomposition Defining classes; The private interface; The public interface. Understanding the need for both.	
Week 2	Understanding the Object Design Model: Applying the principles of *Inheritance* and *Polymorphism*.	Strong vs. Weak coupling; Designing for scalability through extendibility. Applying the principles of *Inheritance* and *Polymorphism.* The binding process: static vs. dynamic.	Assignment #1, Part I: Designing for Usability, Reliability, Correctness, and Modifiability
Week 3	Understanding the Object Design Model: Abstraction by Specification. Understanding the (ADT)	Interfaces, Java's alternative to multiple inheritance.	Assignment #2, Part I: Designing for Usability, Reliability, Correctness, and Modifiability

Week 4	Understanding the Object Design Model: Parametric Polymorphism: Using generic types to further abstraction Research Study	Generic methods, interfaces, and classes; Bounded vs. Unbounded types. Java Collection Library: *Iterators and Iterables, Comparators and Comparables.*	
Week 5	Midterm exam	*Written Exam* *Practicum Exam*	
Week 6	Classes run on a Monday schedule	*No class meeting Tuesday* *Midterm review*	
Week 7	Understanding Design Patterns	Behavioral Patterns (Strategy, Composite)	Assignment #3, Part I: Designing for Extendibility
Week 8	Understanding Design Patterns	Creational Patterns (Singleton, Factory) Creational Patterns (Abstract Factory)	
Week 9	Understanding Design Patterns	Structural Patterns (Composite, Adaptee, Proxy, Facate), Null Pattern	Pair Assignments Assignment #3, Part II: Working together to evaluate an object design, test for extendibility.
Week 10	Software Engineering:	Event Driven Programming and GUI: Designing an independent object structure in a dependent environment.	*Team Assignments:* The final project

Week 11	Software Engineering:	Event Driven Programming and GUI: Designing an independent object structure in a dependent environment.	Importance of Group work, trust and communications.
Week 12	*Excerpts from the Mythical Man Month: Aristocracy, Democracy and System Design* Software Engineering: Understanding the Design Phase and clarity of objective.	Understanding Software Development as an engineering discipline: Life Cycle, Quality Software, and the development team	
Week 13	Software Engineering: The development phase, making adjustments and polishing the presentation.	Final Project	
Week 14		Final Project	