**Boston University** Metropolitan College

BOSTON UNIVERSITY

# Software Design and Patterns

Spring 2025 – MET CS-665
On Campus

Ed Orsini
edorsini@bu.edu
Office hours: by appointment

Meets Thursdays at 6:00PM – 8:45PM EST (Starting on 1/23)

### Course Description

This course is an introduction to software design and patterns. You will learn about general software development principles and software design patterns. A design pattern is a general reusable solution to a commonly occurring problem within a given context in software design. Patterns are formalized best practices that the programmers should implement in the applications. This course covers the most common object-oriented software design patterns. For each pattern, it will discuss the underlying object-oriented design principle it is based on; the specific object-oriented design problem it solves; the context it can be applied and the consequences of applying it; how the different components of the pattern collaborate with each other.

### Recommended Books

The following books are recommended, but are not required.

**Eric, J. Braude. (2003). *Software Design: From Programming to Architecture.***
Publisher: Wiley
ISBN-13: 978-0471204596
ISBN-10: 0471204595
This book can be purchased from Barnes and Noble at Boston University.

**Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Grady Booch. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.***
Publisher: Addison-Wesley Professional
ISBN-13: 978-0201633610
ISBN-10: 0201633612
This book can be purchased from Barnes and Noble at Boston University

**Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra. (2004). *Head First Design Patterns*.** Publisher: O'Reilly
ISBN-13: 978-0596007126
ISBN-10: 0596007124
This book can be purchased from Barnes and Noble at Boston University.

**Robert C. Martin. (2008). Clean Code: *A Handbook of Agile Software Craftsmanship.***
Publisher: Prentice Hall
ISBN-13: 978-0132350884
ISBN-10: 0132350882
This book can be purchased from Barnes and Noble at Boston University.

**Courseware**

Blackboard site: https://onlinecampus.bu.edu/

**Course Prerequisites**

Students should have a solid background in object-oriented programming. The following classes are required/recommended:

- MET CS 341 or MET CS 342 (Data Structures with C++ or Java) is required for understanding the contents of this course.
- METCS520 or METCS521
- MET CS 526 (Data Structures and Algorithms) or MET CS 622 Advanced Programming
  Techniques

The course emphasizes the object-oriented design pattern concepts using Java. All examples in class use Java. Students need to know the Java programming language to implement the homework and project assignments.

**Class Policies**

1. **Attendance & Absences** – I will not be taking attendance. It is the student’s responsibility to keep up with the material covered in class.
2. **Assignment Completion & Late Work** – **Late work will not be accepted**. We recognize that emergencies occur in professional and personal lives. If one occurs that prevents your completion of homework by a deadline, please make

your instructor aware as soon as possible. This must be done in advance of the deadline (unless the emergency makes this impossible, of course), and should be accompanied by particulars that back it up. Additional documentation may be requested. Late submissions without reasons will result in grade deduction: we want to be fair to everyone in this process, including the vast majority of you who sacrifice so much to submit your homework on time in this demanding schedule.

3. **Academic Conduct Code** – Cheating and plagiarism will not be tolerated in any Metropolitan College course. They will result in no credit for the assignment or examination and may lead to disciplinary actions. Please take the time to review the Student Academic Conduct Code: http://www.bu.edu/met/metropolitan_college_people/student/resources/conduct/cod e.html.

   NOTE: [This should not be understood as a discouragement for discussing the material or your particular approach to a problem with other students in the class. On the contrary – you should share your thoughts, questions and solutions.

**Grading Criteria**

The grading percentages for the course is determined by the following:

| | |
|---|---|
| Assignments | 40% |
| Weekly Quizzes | 10% |
| Final Class Project | 20% |
| Final Exam | 30% |

Translation between letter grades and percentages:

| | | |
|---|---|---|
| A | 95-100 | Excellent |

| | | |
|---|---|---|
| A- | 90-94.99 | Excellent; minor improvement evident |
| B+ | 87-89.99 | Very good |
| B | 83–86.99 | Good |
| B- | 80-82.99 | Good mostly some significant improvements needed |
| C+ | 77-79.99 | Satisfactory; some significant improvements needed |
| C | 73-76.99 | Satisfactory; significant improvements needed |
| C- | 70-72.99 | Satisfactory; significant improvements required |
| D | 65-69.99 | Many improvements required |
| F | 0-64.99 | Fail |

## Graded Items

- Assignments: There are a total of 6 assignments. Assignments are software design and implementation assignments using UML diagrams and Java programming languages.
- We will also use some additional tools and frameworks like Apache Maven, FindBugs, CheckStyle, and JUnit. You will learn about these tools in this course.
- You can access and submit the assignment in the "Assignments" area.
- Final Class Project: The Final Class Project is similar to the assignments, but the topic is open and you can develop your own ideas.
- One part of the final project is to present your final project.
- Submission direction: Access and submit the Final Class Project in the "Assignments" area.
- Quizzes: Quizzes are designed to assess your understanding of the material, on a module-by-module basis. There is one graded quiz in each of the six modules. Quiz taking direction:
    - The results for your quiz will be released as soon as possible after the quiz closes. When the quiz results are released, you will be able to see the questions, your answers, and the correct answers

- Access and take the quiz in the "Assessments" area.

**Class Meetings, Lectures & Assignments**
List in a legible format all of the class meetings, lectures, and assignments. One example, based on a computer science course:

*Lectures, Readings, and Assignments subject to change, and will be announced in class as applicable within a reasonable time frame.*

| Date | Topic | Readings | Due @6PM |
|---|---|---|---|
| 1/23 | Introduction and Design Principles, Unified Modeling Language (UML), Class Diagram, Sequence, Use Case, State Diagram | Module 1 | |
| 1/30 | Introduction to Design Patterns, Strategy Pattern, Observer Pattern | Module 1 | Quiz 1 |
| 2/6 | Project Template, Git, and GitHub | | |
| 2/13 | Factory Method, Abstract Factory, Iterator | Module 2 | Assignment 1<br>Quiz 2 |
| 2/20 | Singleton, Facade, State | Module 2 | |
| 2/27 | Pattern Categories, Adapter, Proxy, Decorator | Module 3 | Assignment 2 |
| 3/6 | No Class | | |
| 3/13 | Composite Pattern, Command Pattern, Mediator Pattern | Module 3 | |
| 3/20 | Visitor, Template Method | Module 4 | Assignment 3<br>Quiz 3 |
| 3/27 | Architecture and Frameworks | Module 5 | Assignment 4<br>Quiz 4 |
| 4/3 | MVC, Refactoring 1 | Module 5 | Quiz 5 |
| 4/10 | Refactoring 2, Agile Methodology | Module 6 | Assignment 5 |
| 4/17 | JUnit Testing and TDD | Module 6 | Quiz 6 |
| 4/24 | Presentations 1 | | Assignment 6 |
| 5/1 | Presentations 2, Final Exam Review | | Final Project |
| 5/8 | Final Exam (in class) | | |