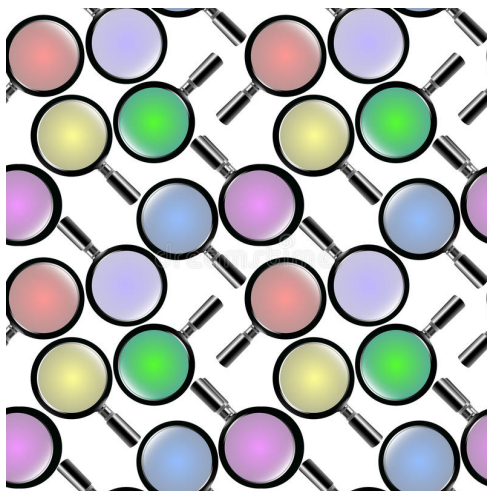


# CS-665: Software Designs and Patterns

## Assignment 5

This document should not be disseminated outside the purview of its intended purpose.

### Finding Patterns in Real-World Repositories (20 points)



#### Application Description

Your assignment is to examine a real-world software project's source code and identify at least three instances of design pattern usage. Write a brief description that outlines where the design pattern is utilized in the code (identifying the relevant components), how it functions as one of the designated design patterns, and its implementation details. You have the option to choose from two open-source projects:

- **Caffeine**

<https://github.com/ben-manes/caffeine>

Caffeine is a high performance, near optimal caching library

- **JUnit 4**

<https://github.com/junit-team/junit4>

JUnit is a simple framework to write repeatable tests. It is an instance of the xUnit architecture for unit testing frameworks.

You can obtain the source code for the project by downloading it directly or using Git to create a local copy.

## Tasks

### Examining the code and identifying design patterns (10 points)

To provide a comprehensive description of the design pattern usage in the project, it is necessary to follow these steps:

1. Identify the class or interface that plays a specific role in the design pattern. For example, if the design pattern is the "Strategy" pattern, identify which class plays the role of the context, which class plays the role of the strategy, etc.
2. Describe the collaboration between the roles as found in the source code. For example, if the "Strategy" pattern is used, describe how the context class delegates the actual implementation of an algorithm to the strategy class.
3. Identify where another class can be easily added and describe why it is useful to do so. For example, if the design pattern is the "Observer" pattern, identify where another observer class can be added and explain why it is easy and useful to add another observer class to the project.

By following these steps, you will be able to provide a comprehensive description of the design pattern usage in the project and its potential for further enhancement.

### UML Class Diagram (10 points)

To produce a UML class diagram that represents the usage of at least three design patterns in the project code, you can follow these steps:

1. Load the project in your preferred IDE, such as Eclipse or IntelliJ.
2. Utilize the IDE's search functions to quickly find and study the code that is relevant to the design pattern in question.
3. Use the IDE's features, such as "search references," to gain a better understanding of the relationships between the classes involved in the design pattern.



4. Consider using IDE plugins, such as ObjectAid, to automatically generate UML diagrams from the code. These tools can save time and effort in producing the diagram.
5. Review the generated UML diagram to ensure that it accurately represents the usage of the design pattern in the project code.

By following these tips, you will be able to produce a UML class diagram that accurately represents the design pattern usage in the project.

## Submission

When you have completed your assignment:

1. Ensure that you have comprehensive descriptions of at least three design patterns used within the chosen package. This should be in your README.md file
2. Create a PDF file of the class diagram for each of the design patterns
3. Submit a zip file containing the README.md file and PDFs of the class diagram. There should be a total of 4 files.

## Late Work

**Late work will not be accepted.** We understand that exceptions can be made in extreme circumstances with proper documentation. For instance, if you provide a doctor/dentist note that verifies you were unable to meet the deadline due to illness, an extension may be granted.

## Academic Misconduct in Programming

In a programming course like ours, it's crucial to understand the line between acceptable collaboration and academic misconduct. Our policy on collaboration and communication with classmates is straightforward: you may not share or receive code through any means, including visually, electronically, verbally, or otherwise. Any other forms of collaboration are permitted.

When it comes to communication with individuals who are not classmates, TAs, or the instructor, it is strictly prohibited. This includes posting questions or seeking assistance on programming forums such as StackOverflow.

When using external resources such as the web or Google, a "two-line rule" applies. You may search for information and access any web pages you need, but you may not incorporate more than two lines of code from an external source into your assignment in any form. Even if you alter the code, such as by changing variable names, it remains a violation to use more than two lines of code obtained from an external source.

**Boston University** Metropolitan College

Ed Orsini | [edorsini@bu.edu](mailto:edorsini@bu.edu)



It is important to properly cite your sources by adding a comment to your code that includes the URL(s) consulted during the construction of your solution. This not only helps to ensure academic integrity but also aids in later recollection of your thought process.