

CSE 331 – Section 11 – Design Patterns

For each question on this page, circle your chosen answer among the listed options.

1. With the benefit of hindsight, which creational pattern would have been a good fit for your graph ADT, considering its subsequent client programs?

Factory Method *Abstract Factory* *Singleton* ***Builder*** *Prototype*

2. If a class's objects never change once initialized but must be initialized incrementally, which creational pattern would let you make the class's objects immutable?

Factory Method *Abstract Factory* *Singleton* ***Builder*** *Prototype*

3. In *Pathfinder* (HW7), would the singleton pattern make sense for the model, given that the program only serves one user per session?

Yes

No

4. In *Campus Paths* (HW9), would the singleton pattern make sense for the model, given that the program (**SparkServer**) may serve multiple users per session?

Yes

No

5. Scenario: You need a generic graph class, similar to what you have created this quarter in CSE 331, but the edges must not have labels. Which structural pattern best fits the task?

Adaptor

Composite

Decorator

Strategy

Proxy

6. Scenario: Your web app, *Campus Paths*, is now very popular; to boost performance, you want the model to cache frequent path queries. Which structural pattern best fits the task?

Adaptor

Composite

Decorator

Strategy

Proxy

7. Which behavioral pattern does the **Comparator** class facilitate?

Command

Interpreter

Observer

Strategy

Visitor

For each sub-question on this page, write your brief answer after the design pattern.

8. For each design pattern, identify an occurrence somewhere throughout the homework series, in either your own code or the starter code. For example, we used the builder pattern to initialize the dataset parsers in HW6 and HW7.

- a. Factory Method

The `CampusPathsParser.initializeReader` method in HW7.

- b. Composite

The generic graph class and the `ModelConnector` class in HW7.

- c. Adaptor

The test drivers and, arguably, the web back-end for *Campus Paths* (HW9)

- d. Iterator

The `Path` class in *Pathfinder* (HW7).

- e. Command

The test drivers in HW5, HW6, and HW7.

- f. Observer

The React front-ends for *Connect the Dots* (HW8) and *Campus Paths* (HW9).

- g. Strategy

Any of the `equals`, `hashCode`, or `compareTo`'s in HW5-7.