Formating Instructions: Please use this provided LaTeX template to complete your homework. When including figures such as UMLs, it is highly encouraged to use tools (graphviz, drawio, tikz, etc..). Figures may be hand draw, however, these may not receive credit if the grader cannot read it. For ease of grading, when including code please have it as part of the same pdf as the question while also including correct formatting/indent, preferably syntax highlighting. Latex includes the minted, or lstlisting package as a helpful tool. For this assignment all code must be full code (no pseudo-code) and be written in either Java or C++. However, implements may ignore all logic not relevant to the design pattern with simple print out statements of "[BLANK] logic done here"

Question Instructions: In this homework assignment, you will apply one or more Structural patterns discussed in the lectures.
This is a group assignment that requires two students per group.
For each question:

1. Give the name of the design pattern(s) you are applying to the problem.
2. Present your reasons why this pattern will solve the problem. Please be specific to the problem and do not give general applicability statements. If there is an alternative pattern, explain why you preferred this one..
3. Show you design with a UML class diagram. If the pattern collaborations would be more visible with another diagram (e.g. sequence diagram), give that diagram as well.
   (a) Your diagram should show every participant in the pattern including the pattern related methods.
   (b) In pattern related classes, give the member (method and attribute) names that play a role in the pattern and effected by the pattern. Optionally, include the member names mentioned in the question. You are encouraged to omit the other methods and fields.
   (c) For the non-pattern related classes, you are not expected to give detailed class names etc. You may give a high-level component, like "UserInterface" or "DBManagement"
4. Give Java or C++ code for your design showing how you have implemented the pattern.
   (a) Pattern related methods and attributes should appear in the code
   (b) Client usage of the pattern should appear in the code
   (c) Non-pattern related parts of the methods could be a simple print. (e.g. "System.out.println()", "cout")
5. Evaluate your design with respect to SOLID principles. Each principle should be address, if a principle is not applicable to the current pattern, say so.

1. (14 points) We are designing a game where players navigate around a map and collect orbs and gemstones. These collectibles are represented by a Collectible interface, a class with only virtual methods. Each type of orb and gemstone has different effects and can be used for trading within the game. There are vast quantities of orbs and gemstones spread all over the game map. Each Collectible item is represented by a separate object either as an instance of Orb class or Gem class. The Orb class has the following attributes: position (2D), size (1-5), rarity (common, rare, scarce), effect (healing, increaseSize), bearer (null or player), and texture (an image). The effect acts on the bearer if there is one. The Gem class has the following attributes: position (2D), name (ruby, sapphire, emerald), size (1-5), rarity (common, rare, scarce), effect (strength, health), bearer (null or player), and texture (an image). The effect acts on the bearer if there is one. Collectible items can draw themselves, activate their effect, move, calculate value using rarity, size, and name of the gem or effect of the orb. It also has a setBearer(Player) method.

   Everything looks good but we have run into a problem: the game is too resource-intensive and runs out of memory due to the vast number of Collectible objects. Your task is to propose a design solution that reduces the game's memory usage while maintaining the game's visual appeal and functionality. (address all items 1-5)

2. (14 points) We are developing a software component for printing different types of library items on different platforms. The library has books, movies, audio tapes, maps, and newspapers. In the first iteration, I want to print books both on the web and on paper (platforms). Printing a book on web requires different formatting than printing a book on paper. In the next iteration, I also need to print movie information (not the video) in a slightly different format on both the web and on paper. Then, I need to print similar information for other item types in the library, such as audio tapes. Currently, these requirements have resulted in an explosion of media and formatting combinations: BookHTMLFormatter, TapeHtmlFormatter, BookPaperFormatter, and so on. Any new combined derivative (for example, MovieHtmlFormatter) will necessarily need to introduce redundancies with the existing implementation. Suggest a structural design pattern to avoid this explosion of media and formatting combinations. (address all items 1-5)

Table 1: Grading Rubric for **14** points questions

1 (1 point)	0	missing or incorrect
	+1	correct pattern
2 (1 point)	0	missing
	+1	the reason provided correctly describes an advantage of the pattern and is specifically beneficial to this scenario
3 (5 points)	0	missing
	+2	includes all participants (including client) that play a role in the pattern
	+2	all class relations are correct
	+1	includes all class members that are related to the pattern
4 (5 points)	0	missing
	+1	includes all pattern related methods and attributes
	+2	includes client usage
	+2	correctly implements and uses all pattern related methods
5 (2 points)	0	missing
	+2	correctly lists multiple ways the pattern benefits a user