1. **Lecture 1: Class Diagrams**

Given the incomplete class structure depicted in Figure 1:

(a) Add, and discuss, possible associations.
(b) Add, and discuss, possible multiplicity indicators.
(c) Add the class outlined below to the class diagram.

```java
public class Bus extends Car {
    private String manufacturer;
    public void Drive();
    public void CollectTickets();
    private void PayDriver();
}
```

![UML Diagram](image)

**Figure 1: UML Diagram for Problem 1**
2. **Lecture 1: Creational Patterns**

You've been tasked with designing the class structure for the software required to manage a pizza restaurant. First you will be working on the core Pizza / Recipe / Oven / Chef subsystem.

(a) You decide to create a class **Pizza**, with methods such as *Bake()* and a class **Chef** responsible with creating **Pizza** classes. Draw the UML diagram for the two classes, and explain which association you would use.

(b) The *Bake()* method needs to use an oven, modeled by an **Oven** class. You have only one oven in your simulated pizza restaurant. Which design pattern would you use, and how would this be implemented?

(c) Draw the UML diagram explaining how you would support the creation of multiple types of pizza, and name the design pattern you would use.

(d) Which design pattern would you use to enable the recipe of the pizza, captured eg in a **Recipe** class, to be used in the creation of the **Pizza** class. Draw the UML diagram for your design.

(e) You are next told that you need to modify your design to be able to support making pasta, captured eg. by a **Pasta** class. Augment your existing UML diagram to support the request. Which design pattern would you be using?

3. **Lecture 2: Structural Patterns**

You are working on a text encryption mobile app. Using a single UML diagram, and naming the various patterns, explain how your app would support:

(a) The application of multiple types of encryption, potentially successively on the same text.

(b) Writing the encrypted text to both (i) a remote server and (ii) local storage memory.

(c) Both a software and a hardware implementation of your encryption algorithms.

(d) The hardware implementation used in the previous part would only be accessible through an old library, written, potentially in a different programming language.

4. **Lecture 2: Structural Patterns**

Use UML diagrams to:

(a) Explain how the Composite design patterns works.

(b) Explain how the Flyweight design patterns works.

(c) Give an example where you combine the two patterns, i.e. use Composite to facilitate operations on Flyweight-style objects.
5. **Lecture 3: Behavioral Patterns**

The pizza restaurant is requesting more features for their management app. You are now required to add functionality to generate the menu automatically from the list of pizza recipes. You want to reuse as much of the previously created design as possible.

(a) You first need to be able to extract the list of ingredients from each `Recipe` class. Explain, using C++ or pseudocode, how you can do this using the Visitor pattern.

(b) You next define a `RecipeCollection` class to hold a list of possible `Recipes`. Detail, using an UML diagram, the elements of the design pattern would you use to access the element of this collection.

(c) The `RecipeCollection` class is owned by a `Menu` class. Explain, using an UML diagram, which design pattern you would use to save (and potentially restore) the recipe details to a file.

6. **Lecture 3: Behavioral Patterns**

You released your text encryption mobile app on the app store of your choice, but you want to make several improvements:

(a) You did not expect your users to want to be able to decrypt the text, but this seems to be the most requested functionality. Explain how the Chain of Responsibility design pattern would enable you to implement the decryption.

(b) The investor who supported you in writing the app now suggests that your app would be a lot more popular if it allowed users to follow each other’s messages. Detail, using an UML diagram, the class diagram of the design pattern you would use to enable the following functionality.

7. **Lecture 4: Algorithm Design Patterns**

The initial version of your text encryption and decryption app always encrypted the text using a known succession of two encryption algorithms, both using the same password.

You now want to extend this to a user defined, and unknown at decryption time, succession of 8 encryptions, using the same two algorithms, and keeping the same password for all encryptors.

Outline how you would use the backtracking strategy to find the correct chain of decryptors.

8. **Lecture 4: Algorithm Design Patterns**

The pizza restaurant is very happy with the software you created for them, and now wants you to help reduce the amount of food waste they produce. Their idea is to introduce an eco-pizza of the day offer – each day this would allow surplus and previous day ingredients to be used.

Your task it therefore to devise an algorithms that would select, from the set of possible pizza recipes, the one that would use the most leftover and surplus ingredients from the ones currently available. Outline how you might to about accomplishing this.