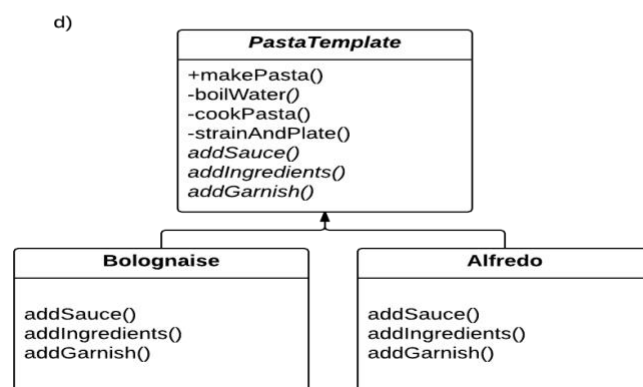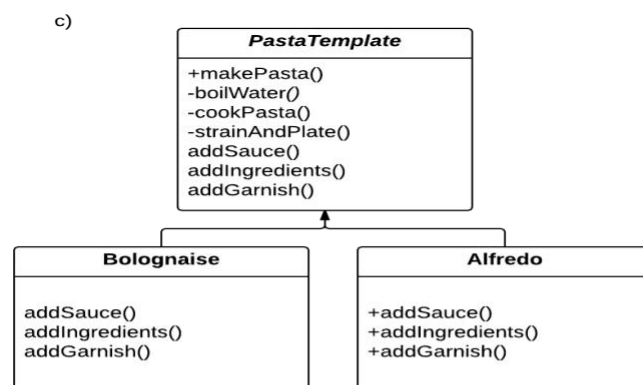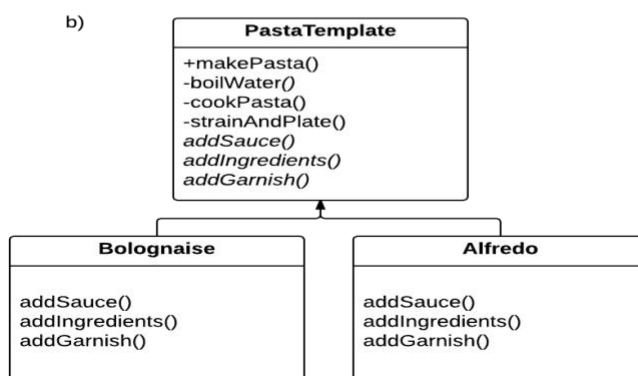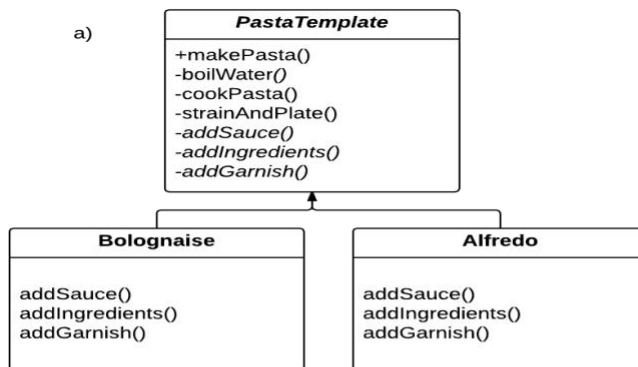Question 1: Choose the most appropriately implemented Template pattern.
Some UML reminders that will help you:

1. a private method or variable is denoted by a - as in -boilWater().
2. a method, variable, or class that is abstract is denoted by italics (as in PastaTemplate)

a)

PastaTemplate
+makePasta()
-boilWater()
-cookPasta()
-strainAndPlate()
-addSauce()
-addIngredients()
-addGarnish()

Bolognaise
addSauce()
addIngredients()
addGarnish()

Alfredo
addSauce()
addIngredients()
addGarnish()

b)

PastaTemplate
+makePasta()
-boilWater()
-cookPasta()
-strainAndPlate()
addSauce()
addIngredients()
addGarnish()

Bolognaise
addSauce()
addIngredients()
addGarnish()

Alfredo
addSauce()
addIngredients()
addGarnish()

c)

PastaTemplate
+makePasta()
-boilWater()
-cookPasta()
-strainAndPlate()
addSauce()
addIngredients()
addGarnish()

Bolognaise
addSauce()
addIngredients()
addGarnish()

Alfredo
+addSauce()
+addIngredients()
+addGarnish()

d)

PastaTemplate
+makePasta()
-boilWater()
-cookPasta()
-strainAndPlate()
addSauce()
addIngredients()
addGarnish()

Bolognaise
addSauce()
addIngredients()
addGarnish()

Alfredo
addSauce()
addIngredients()
addGarnish()

d)

Question 2: What is the correct situation for the use of a Chain of Responsibility pattern?
- You need a set of objects to each contribute information on responding to a request.
- ==You have multiple potential handlers, but only one will deal with the request.==
- You need to pass a message to multiple receivers.
- You need to delegate a set of tasks to a hierarchy of objects.

Question 3: What is the purpose of encapsulating state in an object in the State Pattern? Choose the **three** that are correct.
- ==it turns the context into a client of the state.==
- ==it allows the current state object to decide how to achieve behaviours specific to the state of the context.==
- ==it removes large conditionals that are difficult to maintain.==
- it allows the current state to be copied from one instance to another

Question 4: What design principles is the Command Pattern using?
- ==Encapsulation, generalization, loose coupling==
- Encapsulation, information hiding, loose coupling
- Encapsulation, generalization, information hiding
- Generalization, information hiding, loose coupling

Question 5: Which are the minimum requirements of the Observer pattern? Choose the **three** that are correct.
- ==update method in observers==
- ==methods to add or remove observers==
- a state variable to determine if observers have been notified.
- ==method to notify observers==

Question 6: When are you most likely to need a Mediator pattern?
- When your class is sending a request that might be handled by one of several handlers.
- ==When you are coordinating the activities of a set of related classes.==
- When you want to de-couple a class that is requesting a service from one that is providing it.
- When you have two classes with different interfaces that you must connect.

Question 7: Marlon is coding part of the software that follows a similar sequence of steps. Depending on the type of object, these steps will be implemented in slightly different ways, but their order is always the same. Which design pattern could Marlon use?
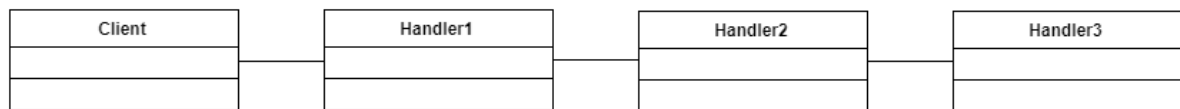- ==Template pattern==
- Mediator pattern
- Command pattern
- State pattern

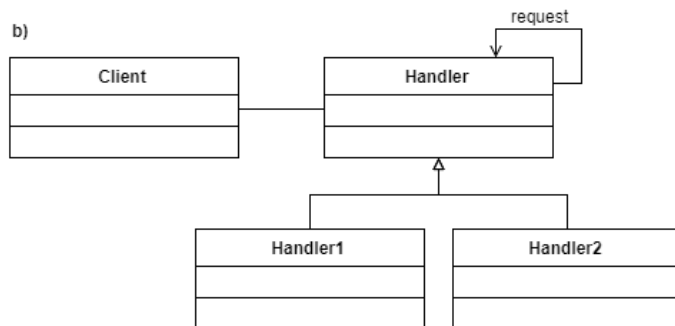Question 8: What are the important roles in the Command Pattern?
- Sender, Receiver, Invoker
- Delegate, Command, Requester
- ==Command, Receiver, Invoker==
- Command, Queue, Receiver

Question 9: Select the best UML class diagram representation of the Chain of Responsibility pattern.
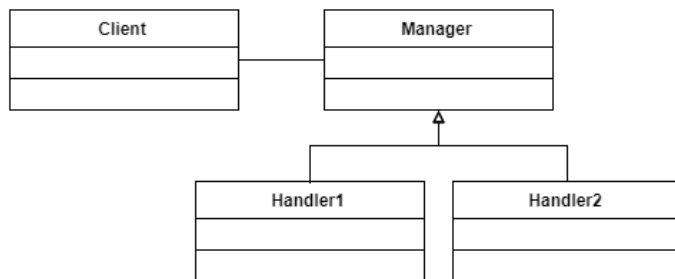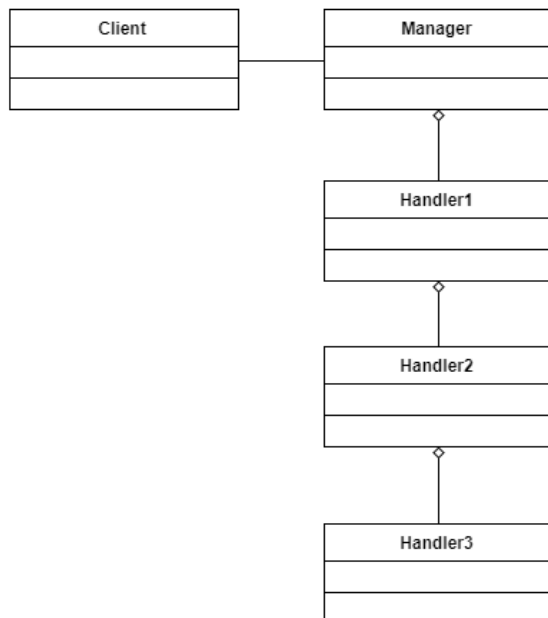
a)

Client

Handler1

Handler2

Handler3

b)

request

Client

Handler

Handler1

Handler2

c)

Client

Manager

Handler1

Handler2

d)

Client

Manager

Handler1

Handler2

Handler3

b)

Question 10: You have a machine performing a complex manufacturing task, with different sensors and different components of the machine represented by different classes. Which design pattern will you use to arrange the parts?
- Template
- Command
- Mediator
- Chain of Responsibility

Question 11: You have a security system class, and it has 3 modes: normal, lockdown, and open. Which pattern would you use to model the behaviour in these different modes?
- Observer
- State
- Mediator
- Template

Question 12: One of your classes represents a mailbox, while another is the owner of the mailbox. The person would like to know when new mail arrives. Which design pattern will you probably use?
- Mediator
- State
- Command
- Observer