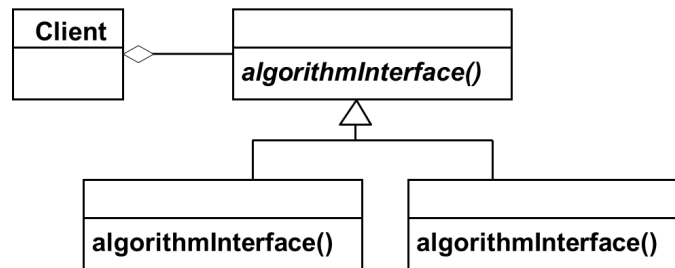# Question 1 (30 marks)

Answer the following 6 short answer questions.
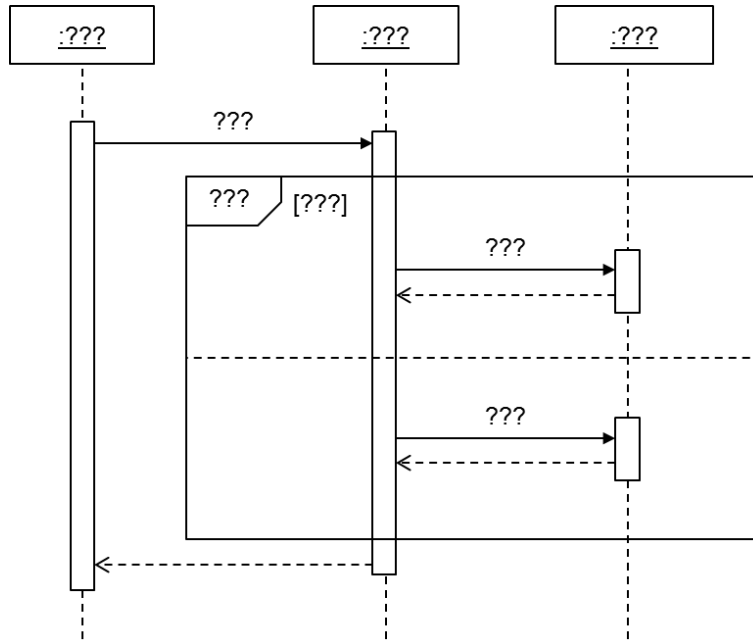
Question 1.1 (5 marks)

Name the *design pattern* shown below. Describe <u>one</u> use of the design pattern with a concrete example.

Question 1.2 (5 marks)

Name the various parts of the ***combined fragment*** shown below. Give <u>two</u> examples of different types of combined fragments.

Question 1.3 (5 marks)

Why is the following *class diagram* incorrect? Give the correct solution. Assume that a Book is always written by one Author.
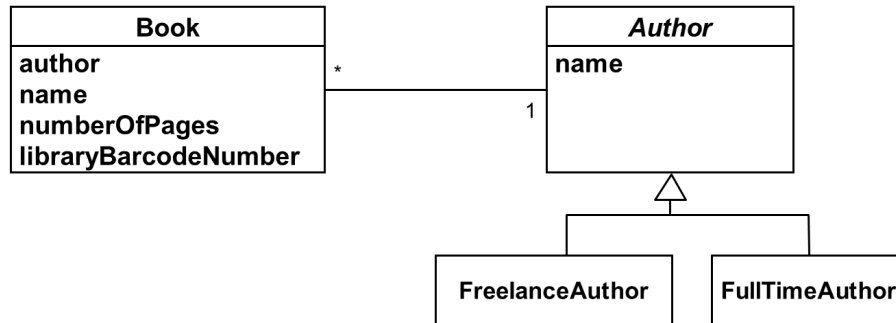
Question 1.4 (5 marks)

Which **modeling constructs** does Umple add to a programming language like Java? Give <u>one</u> example of <u>each</u> modeling construct.

Question 1.5 (5 marks)

When specifying constraints, which **collection types** may result from navigating associations?

Question 1.6 (5 marks)

Why are **composite states** useful?

Question 2 (25 marks)

Given the problem description of the Restaurant Billing System (RBS) on page 14, specify the **domain model** of the RBS with a class diagram. Describe all mentioned concepts of the proposed system and their relationships. You do <u>not</u> need to show (a) operations and (b) types of attributes.

Question 3 (25 marks)

Given the problem description of the Restaurant Billing System (RBS) on page 14, specify the ***state machines*** for (i) a reservation, (ii) a customer, (iii) an order, and (iv) an individual item of an order. You are required to show all states and transitions including events and guards. You do <u>not</u> need to show actions on transitions or actions in states <u>except</u> when an action describes an interaction between two state machines or an action manipulates data related to a guard. Make sure that the state machines are <u>consistent</u> with the domain model from Question 2.

Question 4 (10 marks)

Given the problem description of the Restaurant Billing System (RBS) on page 14, describe (i) the ***user interface*** for adding an item to an order in a <u>few</u> sentences and (ii) the Controller method called by the user interface to add an item to an order with the help of a ***sequence diagram***. If necessary, use comments in the sequence diagram for those statements that cannot be expressed with a message between two objects in a sequence diagram. Make sure that the sequence diagram is <u>consistent</u> with the domain model from Question 2.

Question 5 (10 marks)

Given the problem description of the Restaurant Billing System (RBS) on page 14, specify three *constraints* that are not covered by the domain model from Question 2.

Student Name		McGill ID								
Signature		Date								

Problem Description for Questions 2 to 5

Restaurant Billing System (RBS)

A restaurant allows customers to reserve tables for a specific date, time, and number of persons. To hold the reservation, the restaurant requires the name of the customer as well as a phone number or email address as contact information. In return, the customer is given a reservation number. A customer may cancel the reservation at any time. However, if a customer cancels two reservations less than 24hrs before the reservation date or is a no show for two reservations, the customer is not allowed to make a reservation for 6 months. The restaurant also accepts walk-ins.

At the restaurant, a waiter takes the order of each customer seated at a table. A waiter identifies each table by its number. The customers have the choice of getting one bill per table or individual bills. Shared items such as a bottle of water or wine are divided equally among the sharing customers on individual bills. The price of an item already includes all taxes (note: taxes are out of scope for the RBS). If a party is large enough to require several tables, one bill for several tables may also be issued. Once an order has been placed, the kitchen is automatically notified and starts preparing the dishes. When a dish is ready, the waiter is informed automatically to pick up the dish from the kitchen and serve it to the customer. Similarly, the bar is notified of drink orders and the waiter is informed when a drink is ready to be served. It is possible for a customer to change an order by cancelling one item or adding another. The whole order may also be cancelled.

For legal and accounting reasons, all orders including cancelled orders are retained by the system. Reservations, however, are not retained when they are not needed anymore. The system also has to keep track of whether an order has been paid or not. Payment details, however, are out of scope for the RBS.