Spring 2013 Final Exam:

2(a) In the interview, Alistair Cockburn stated that business people and developers should work together when writing use cases. Why?
Answer: Business people tend to write use cases very vague and ambiguous. Developers tend to go on the other side. By having them work together, they can come to a common language that both of them can understand.

2(b) What is wrong with the following use case?
When the user clicks the checkout button a new page appears with the details of his delivery. After he fills it in, he clicks on the left button and a confirmation dialog appears asking him to confirm his purchase.
Answer: The use case is written in terms of the GUI.

13(a) You are working at a company that builds software solutions for small and medium sized hotels.
Your team is responsible for the backend part, where you have to handle the business logic. You have to implement the following scenario:
The system must allow users to log into the system and reserve a room. Some hotels also allow you to reserve special items (like skis, snowboards, bikes etc). These are added to the room reservation and show on the bill as separate items. The customer can then change her reservation or cancel it before the deadline. Cancellation and change depends on a cancellation policy. At the end of the stay, the customer receives an itemized bill for the services she used.
Each hotel can have different options of extra items, so the manager must be able to manage these. Some of the extra items may be available year round, while some may be available only during a certain time frame (e.g., you can only rent skis during winter).

Using the brief use case format, write the use cases for the above system.

Answer:
1. User creates a reservation
2. User changes a reservation
3. User cancels a reservation
4. (1-3) User manages a reservation
5. Receptionist bills a customer
6. Manager adds/edits/removes (manages) extras and the time when they are available.

Spring 2012 Final Exam:

3(a) Does the primary actor of a use case also need to be a stakeholder for it? Explain.
Answer: Actors are always stakeholders, but not all stakeholders are actors, since they "never interact directly with the system, even though they have the right to care how the system behaves." For example, "the owners of the system, the company's board of directors, and regulatory bodies such as the Internal Revenue Service and the

Department of Insurance" could all be stakeholders but are unlikely to be actors.

3(b) Should the system implement at least one use case for which the stakeholder needs to be an actor? Explain.
Answer: No, because a stakeholder may play both an active and an inactive role. If the stakeholder is an inactive actor, then there is no interaction with the system and therefore, no use case.

3(c) Describe two differences between user stories and use case briefs.
Answer:
1. XP stories (and similar things, often called features) break requirements into chunks for planning purposes. Stories are explicitly broken down until they can be estimated as part of XP's release planning process. While use cases organize requirements to form a narrative of how users relate to and use a system. Hence they focus on user goals and how interacting with a system satisfies the goals.
2. User stories provide a small-scale and easy-to-use presentation of information, with little detail, thus remaining open to interpretation, through conversations with on-site customers. While Use case flows describe sequences of interactions, and may be worded in terms of a formal model. A use case is intended to provide sufficient detail for it to be understood on its own.

Spring 2012 Midterm Exam:

3(a) In the interview with the author Alistair Cockburn, he described several characteristics of good use cases. List two of them.
Answer: Use cases should hold Functional Requirements in an easy to read, easy to track text format.They should also represent the goal of an interaction between an actor and the system. The goal represents a meaningful and measurable objective for the actor.

3(b) Describe a scenario when use cases are not suitable for capturing the requirements.
Answer: Use cases are not well suited to capturing non-interaction based requirements of a system (such as algorithm or mathematical requirements) or non-functional requirements (such as platform, performance, timing, or safety-critical aspects). These are better specified declaratively elsewhere.

Spring 2010 Final Exam:

3(a) Use cases are not always an effective technique. Describe a situation or system where use cases do not work or cannot easily be used.
Answer: Use cases are not well suited to capturing non-interaction based requirements of a system (such as algorithm or mathematical requirements) or non-functional requirements (such as platform, performance, timing, or safety-critical aspects). These are better specified declaratively elsewhere.

There are also some limitations to use cases that are mentioned in the wiki page. You can also mention any of these:

Use case templates do not automatically ensure clarity. Clarity depends on the skill of the writer(s).
For some products and systems, use cases are complex to write and to understand, for both end users and developers.
As there are no fully standard definitions of use cases, each project must form its own interpretation.
Some use case relationships, such as extends, are ambiguous in interpretation and can be difficult for stakeholders to understand.
In Agile development, especially Extreme programming, simpler user stories are preferred to use cases.
Use case developers often find it difficult to determine the level of user interface (UI) dependency to incorporate in a use case. While use case theory suggests that UI not be reflected in use cases, it can be awkward to abstract out this aspect of design, as it makes the use cases difficult to visualize. In software engineering, this difficulty is resolved by applying requirements traceability, for example with a traceability matrix. Another approach to associate UI elements with use cases, is to attach a UI design to each step in the use case. This is called a use case storyboard.
Use cases can be over-emphasized. Bertrand Meyer discusses issues such as driving system design too literally from use cases, and using use cases to the exclusion of other potentially valuable requirements analysis techniques.
Use cases are a starting point for test design, but since each test needs its own success criteria, use cases may need to be modified to provide separate post-conditions for each path.

3(b) Students in a software engineering course are creating a web application that allows photographers to post their photos for clients to see. While browsing the project's wiki page, you see the following use case: "Create a database to hold photographs." Why is this not a use case?
Answer: Because use cases describe what the user tries to accomplish with the system, not what a developer does. Creating a database is a task the developers of the system need to do, not the users.