You have decided to create a blog to raise awareness on good programming practices. You want readers to be able to subscribe to your blog so that they receive notifications whenever you post new content. In order to achieve this, your blog should be able to maintain a list of subscribers and must be able to add, remove or notify subscribers whenever there is new content. You want to use a design pattern to achieve this. You have reached out to your friends who are also programmers, and they helped you narrow down your options to the following three design patterns. Now you need to select the one that would be the most appropriate for the task at hand.

- Visitor Design Pattern
- Abstract Factory Design Pattern
- Observer Design Pattern

a) Which design pattern do you think would best fit this requirement? [02]

[Solution]: Observer Design Pattern (2 ticks)

b) Discuss why you think this is an appropriate design pattern. [04]

[Solution] (4 ticks max):

- Maintaining a list of "subscriber" objects (2 ticks)
- Notifying subscriber objects about changes to the subject (2 ticks)
- Subject needs to share its state with subscribers without knowing who they are (2 ticks)
- Any other valid statement relating the Observer Design Pattern to the scenario (2 ticks)

c) Discuss the limitations that would need to be considered before implementing the design pattern. [04]

[Solution] (4 ticks max):

- The subject holds a reference to the object until it is deregistered, which can lead to memory leaks (2 ticks)
- It is difficult to see the path through the code, especially when there are chains of observers (2 ticks)