Question 1:When is the best time to use a design pattern? Choose **two** answers.

- For a problem that is unique to your program.
- For a commonly-encountered issue.
- When fixing spaghetti code
- When explaining a solution to your fellow developers

Question 2: What is the purpose of the Singleton pattern? Select the **two correct** answers.

- to provide simple classes with only one method
- to enforce collaboration of a class with only one other class
- to provide global access to an object
- to enforce instantiation of only one object of a class

Question 3: What does it mean to "let the subclass decide" in the Factory Method Pattern?

- the subclass will pass a parameter into a factory that determines which object is instantiated.
- the subclass decides which object to create, but calls a method that is defined in the superclass to instantiate the class
- the subclass defines the methods for concrete instantiation. As such, the type of object is determined by which subclass is instantiated.

Question 4: What do we call the creation of an object, for example, with the 'new' operator in Java?

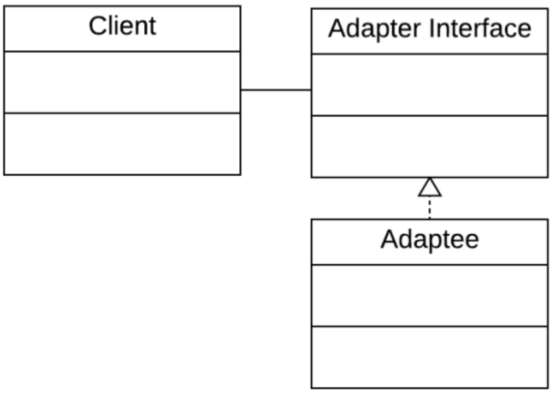
- manifestation
- object realization
- class creation
- concrete instantiation.

Question 5: What are the advantages of the **Facade** pattern? Select the **three correct** answers.

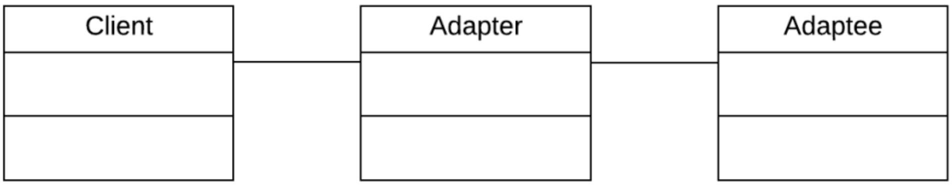
- The Facade class redirects requests as needed
- The complexity of the subsystem is hidden
- The subsystem can handle more clients
- The client and the subsystem are more loosely coupled

Question 6: Which of the following diagrams shows the **Adapter** pattern?

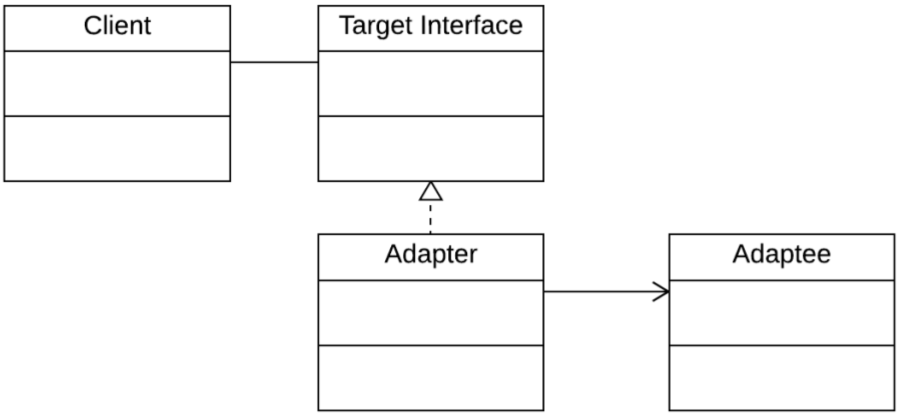
a)



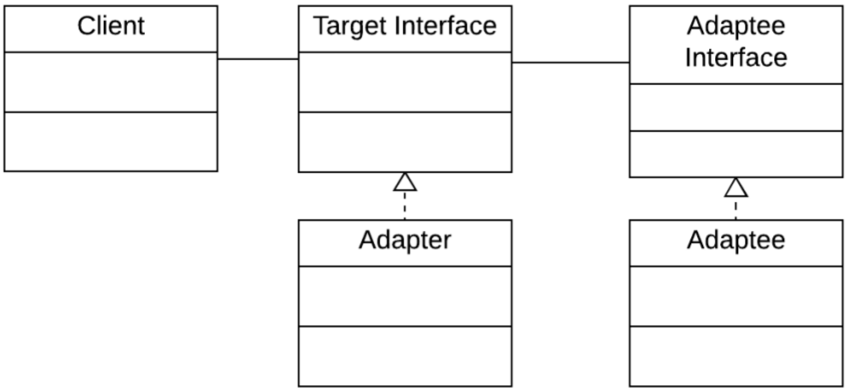
b)



c)



d)



d)
a)
b)
c)

Question 7: Which of these are the best applications for a **Composite** Pattern? Choose the **three correct** answers.

- Files and folders
- Students in a class
- Music in a playlist
- Elements in a user-interface dialog

Question 8: Which of these is **NOT** a common application of the **Proxy** Pattern?

- information proxy
- protection proxy
- remote proxy
- virtual proxy

Question 9: How does a **Decorator** Pattern work? Choose one.

- encapsulates a class to give it a different interface
- expands the methods of a class with inheritance
- builds a behaviour by stacking objects
- adding features to a class with a new class

Question 10: What are the object types that are used in the **Composite** Pattern? Select the **two correct** answers.

- composite
- leaf
- branch
- trunk
- root

Question 11: Many different clients need to create a similar object. You would like to outsource this concrete instantiation to a dedicated class. Which technique will you use, in one word?

Factory

The correct answer is factory. Factories of different types are used to instantiate objects. This could be a simple factory, which is an object which is tasked with concrete instantiation. Factory Methods move concrete instantiation is achieved by a method- that is abstract in the superclass and specified in the subclass.

Question 12: How do you enforce the creation of only one Singleton object? Select the **two correct** answers.

- Give the Singleton class a private constructor
- Specify in the comments that only one Singleton object is to be instantiated.
- Throw an exception if a Singleton object is already instantiated
- Write a method that can create a new Singleton object or return the existing one.