Sample Quiz

1. Design patterns does not follow the concept of software reuse.
   a) True
   b) False

2. Which pattern prevents one from creating more than one instance of a variable?
   a) Factory Method
   b) Singleton
   c) Observer
   d) None of the mentioned

3. Which of the following is an advantage of partition testing.
   a) Ensures that every possible input is tested
   b) It reduces the likelihood of defects being missed
   c) **It saves time and effort by reducing the number of test cases required**
   d) None of the above

4. **Your team is developing a web application. Explain how you would apply the open/Closed principle (OCP) to ensure that the system is maintainable and extensible.**
   Answer: OCP states that software entities should be open for extension but closed for modification, you should be able to add new functionality to a system without modifying its existing code.
   To apply OCP to a web application, we need to ensure that the application's modules are designed in a way that allows new features to be added without modifying the existing code. Here are some steps we could take:
   1. Identify the parts of the application that are most likely to change in the future. this could include user interfaces, data access layer or business logic.
   2. Design these parts of the application to be modular and loosely coupled, so that changes in one module do not affect the others.
   3. Use well-defined interfaces or APIs to connect these modules. This will make it easier to replace or update a module without affecting other parts of the system.
   4. Use design pattern such as strategy pattern or template method pattern to encapsulate behavior that may vary between modules. This will allow new behavior to be added to the system without modifying existing code.

   By following these steps, we can ensure that our web application is maintainable and extensible, and that we can add new features to the system without introducing bugs or breaking existing functionality.

Which design pattern(s) could be applied to a software application that allows users to perform task management, where users can add tasks and group them into projects with the ability to nest projects arbitrarily deep, and where the estimated completion time is specified when a task is created, and the time to complete a project is the sum of all the times required to complete its tasks and sub-projects?

Answer: One design pattern that could be applied to this software application is the Composite pattern. The Composite pattern allows objects to be composed into tree structures, and these structures can be treated as individual objects or groups of objects.

In the context of this software application, the Composite pattern can be used to treat tasks and projects as individual objects, and groups of tasks and sub-projects as composite objects. This allows for a uniform treatment of tasks and projects, as well as their groups, in terms of time needed for completion.

The Composite pattern provides a way to represent the hierarchical structure of tasks and projects as a tree structure. Each node of the tree can represent either a task or a project, and the leaf nodes represent tasks with no sub-projects. The time needed to complete a project is calculated by recursively summing the times needed to complete all its tasks and sub-projects.

In summary, the Composite pattern is a suitable design pattern for this software application, as it allows for a uniform treatment of tasks and projects, and their groups, in terms of time needed for completion. It also provides a way to represent the hierarchical structure of tasks and projects as a tree structure, which enables the recursive calculation of the time needed to complete a project.

Draw a decision table using the True/False method for the following business rules:

a. **given the hours (hours), and hourly pay rate (rate), the program calculates pay as hours times rate if the values for hours and rate are valid. The program displays the message "invalid hours" if the value for hours is not between 1 and 40 inclusive. The program displays the message "invalid rate" if the value for rate is not between 10 and 25 inclusive. The program always checks both the hours and the rate.**
Answer:

Conditions	R1	R2	R3	R4
Enter valid hour [1-40]	T	T	F	F
Enter valid rate [10-25]	T	F	T	F
Actions				
Calculate Pay	x			
Error msg "invalid rate"		x		x
Error msg "invalid hour"			x	X