Quiz: Low Level Design

Question 1

1/1 point (graded)

Which of the following are considered to be benefits of the low level design goal 'Encapsulate what varies'?

Select all that apply.

- [ ] it helps decouple implementation from design
- [x] it makes it easier for systems to be extended
- [ ] it improves the reusability of code in a system
- [x] it helps future bug fixes to be more localized
- [ ] it makes systems more dynamic at runtime

✔

Submit You have used 1 of 1 attempt

Question 2

1/1 point (graded)

Which of the following are considered to be benefits of the low level design goal 'Design to interfaces'?

Select all that apply.

- [x] it helps decouple implementation from design
- [ ] it makes it easier for systems to be extended
- [x] it improves the reusability of code in a system
- [ ] it helps future bug fixes to be more localized
- [ ] it makes systems more dynamic at runtime

✔

Submit You have used 1 of 1 attempt

Question 3

1/1 point (graded)

Which of the following statements is NOT true about design patterns?

- ○ they provide solutions for specific evolutionary problems
- ◉ they are guaranteed to improve the design of any system ✔
- ○ they leverage existing design knowledge from past software developers
- ○ they are designed to improve a system's resilience to future change

Submit You have used 1 of 1 attempt

Question 4

1/1 point (graded)

In which scenario would it be most appropriate to use a Singleton pattern?

- we have an object that we want to use sparingly in our system, and we want a single instance of that object
- we have an object that we want to use sparingly in our system, and we want multiple instances of that object
- we have an object that we want to use widely in our system, and we want multiple instances of that object
- we have an object that we want to use widely in our system, and we want a single instance of that object ✔

Submit You have used 1 of 1 attempt

Question 5

1/1 point (graded)

Which of the following statements are true about the design solution in the Strategy pattern?

- [ ] it provides a mechanism for encapsulating algorithms to support future modification
- [x] it provides a mechanism for encapsulating algorithms to support future extension
- [x] it provides a mechanism for varying the states of our program in a static way
- [ ] it provides a mechanism for varying the states of our program in a dynamic way

✔

Submit You have used 1 of 1 attempt

Question 6

1/1 point (graded)

Within the State pattern, how are state transitions handled?

- (•) at run time, a concrete state invokes the setState method on the context object ✔
- ( ) at run time, a client invokes the setState method on the context object
- ( ) at run time, a client invokes the setState method on the concrete state object
- ( ) at run time, a concrete state invokes the setState method on the concrete state object

Submit You have used 1 of 1 attempt

Question 7

1/1 point (graded)

What design trade-off is present in the Facade pattern?

- ( ) it improves the reusability of code in a system, but it violates the open/closed principle
- ( ) it improves the reusability of code in a system, but it violates the dependency inversion principle
- ( ) it provides a simplified view of a complex system, but it violates the interface segregation principle

- (selected) it provides a simplified view of a complex system, but it violates the single responsibility principle ✔

Submit You have used 1 of 1 attempt

Question 8

1/1 point (graded)

In the Decorator pattern, what purpose does the Component serve?

- it wraps a concrete component in various decorator components
- it wraps a decorator component in various concrete components
- (selected) it declares the high level actions that need to be performed ✔
- it controls the interactions between decorator components

Submit You have used 1 of 1 attempt

Question 9

1/1 point (graded)

What attributes do the MVC and MVP patterns have in common?

Select all that apply.

- [x] they encourage views in a system to be lightweight
- [x] they enhance the testability of a system

- [ ] they always use the Observer pattern
- [x] they encourage designers to pull functionality out of the model of a system
- [ ] they use a controller to decouple the views from the business logic of a system

✔

Submit You have used 1 of 1 attempt