Unit 11 - Week 9 :

Assignment 9

The due date for submitting this assignment has passed.
As per our records you have not submitted this assignment.

Due on 2020-11-18, 23:59 IST.

1) 1 point

Law of Demeter design pattern does not recommend a method call to which one of the following types of objects?

a. This object (or self)
b. An object parameter of the method
c. An object attribute of self
d. An object that is an attribute of a called method

a.
b.
c.
d.

No, the answer is incorrect.
Score: 0
Accepted Answers:
d.

2) 1 point

Which one of the following statements is not true of the façade design pattern?

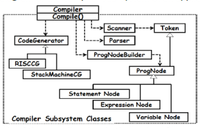
a. Use of façade class in a server package reduces the overall coupling in the design
b. Use of façade class in a server package increases the overall cohesion of the classes in the server package
c. The façade class often does little more than just delegating requests to other classes inside the package.
d. A façade class provides a common interface to the services of the package

a.
b.
c.
d.

No, the answer is incorrect.
Score: 0
Accepted Answers:
b.

3) 1 point

Consider the following class diagram of the compiler sub-system of an application.



What can be said about the role of the compiler class in the package?

a. Façade
b. Observer
c. Expert
d. Controller

a.
b.
c.
d.

No, the answer is incorrect.
Score: 0
Accepted Answers:
a.

4) 1 point

In a push from below mechanism for interaction between the view and model elements, the model object hard codes the references of the view objects and notifies them as soon as an event of interest occurs. Which one of the following is a reason for **not** using a push from below mechanism for interaction between the view and model classes in an application such as network monitoring?

a. View objects are usually transient
b. Sometimes more view objects are added and some view objects are taken off.
c. A change to a view object should not require change to the model object.
d. Model objects may need to be changed at times.

a.
b.
c.
d.

No, the answer is incorrect.
Score: 0
Accepted Answers:
d.

5) 1 point

Which one of the following is achieved through a judicious use of the observer pattern?

a. Loose coupling between the model and view objects
b. Encapsulation of the observers
c. Polymorphic binding between the observer and observable
d. Composition of the observers

a.
b.
c.
d.

No, the answer is incorrect.
Score: 0
Accepted Answers:
a.

6) 1 point

Which one of the following is achieved through a judicious use of the façade pattern?

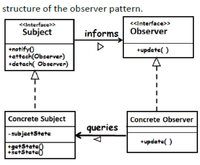
a. Increase in cohesion
b. Reduction in overall coupling
c. Reduction in the number of message exchanges
d. Enforcement of the principle: "do not talk to strangers"

a.
b.
c.
d.

No, the answer is incorrect.
Score: 0
Accepted Answers:
b.

7) 1 point

Consider the following class structure of the observer pattern.



When the state of the concrete subject changes, which one of the following should take place?

a. **Concrete observer** should invoke the **getState** method of the **Concrete subject** class
b. **Concrete observer** should invoke the **setState** method of the **Concrete subject** class
c. **Concrete subject** should invoke its **getState** method
d. **Concrete subject** should invoke its **notify** method

a.
b.
c.
d.

No, the answer is incorrect.
Score: 0
Accepted Answers:
d.

8) 1 point

In the observer pattern, why does the problem of *dangling references* occur?

a. A concrete observer may die unexpectedly
b. The concrete subject may die unexpectedly
c. The concrete subject may fail to invoke the detach method
d. A concrete observer may get created unexpectedly

a.
b.
c.
d.

No, the answer is incorrect.
Score: 0
Accepted Answers:
a.

9) 1 point

While designing the structure of communication between a set of observer objects and a subject, in which one of the following situations is the use of the observer pattern recommended?

a. Number of observers is fixed and the data stored in subject is static
b. Number of observers is fixed and the data stored in subject is dynamic
c. Number of observers is flexible and the data stored in subject is static
d. Number of observers is flexible and the data stored in subject is dynamic

a.
b.
c.
d.

No, the answer is incorrect.
Score: 0
Accepted Answers:
d.

10) 1 point

In the Java observer pattern, when the concrete subject invokes the update method of the concrete observer, why does it pass a reference of itself to the observer? A prototype of this method is: **update(Observable x, Object y)**

a. Observer does not know the reference of the Observable
b. Observer needs to update its local copy of the reference to the Observable
c. Observer ignores this redundant information
d. Observer may be observing multiple observers

a.
b.
c.
d.

No, the answer is incorrect.
Score: 0
Accepted Answers:
d.