

Started on Friday, 26 March 2021, 4:38 PM

State Finished

Completed on Friday, 26 March 2021, 5:03 PM

Time taken 25 mins 27 secs

Marks 8.00/11.00

Grade 72.73 out of 100.00

Question 1

Correct

Mark 1.00 out of 1.00

A Web based application that integrates HTML code with business logic is in violation of which of the following design principles?

- ☐ a. Liskov's Substitution Principle
- ☐ b. Dependency Inversion Principle
- ☐ c. Loose Coupling
- ☒ d. Single Responsibility Principle
- ☐ e. Open-Closed Principle



Your answer is correct.

The correct answer is: Single Responsibility Principle

Question 2

Correct

Mark 1.00 out of 1.00

High coupling is usually better than loose coupling because it makes the system more integrated.

Select one:

- ☐ True
- ☒ False



The correct answer is 'False'.

Question 3

Correct

Mark 1.00 out of 1.00

Which of the following is NOT an appropriate responsibility for a Manager component?

- ☒ a. Know how data is stored in a database
- ☐ b. Coordinate the use case execution
- ☐ c. Invoke services provided by other components
- ☐ d. Know the sequence of operations carried out in a use case narrative



Your answer is correct.

[Dashboard](#) > [INFS 2044](#) > [Week 4 - Implementation Design](#) > [Quiz 4](#)



Question 4

Correct

Mark 1.00 out of 1.00

Which of the following statements are true for the following design expressed as components and responsibilities: (2 correct answers)

BookingManager component:

- Implement the system operations related to making bookings
- Create/update/cancel bookings
- Store and retrieve bookings from the database
- Calculate the booking fee

Booking component:

- Know the booking time
- Know the booked service
- Know the Member who the booking is for

Service component:

- Know the service description
- Know the price of the service

Member component:

- Know the member's details

Calendar

- Know the bookings
- Add and remove bookings
- Enforce business rules related to making bookings

- ☒ a. This design does not adhere to the Single Responsibility principle, because a change in the database may trigger changes in the BookingManager component and changes in business rules may also trigger changes in that component. ✓
- ☐ b. This design exhibits High Coupling because there are too many components.
- ☒ c. This design does not satisfy the High Cohesion principle because the BookingManager component has multiple unrelated responsibilities. ✓
- ☐ d. This design exhibits High Cohesion because most of the component have focused responsibilities.
- ☐ e. The BookingManager component satisfies the Open-Closed principle because it is possible to create additional components without modifying the code of BookingManager.

Your answer is correct.

The correct answers are: This design does not satisfy the High Cohesion principle because the BookingManager component has multiple unrelated responsibilities., This design does not adhere to the Single Responsibility principle, because a change in the database may trigger changes in the BookingManager component and changes in business rules may also trigger changes in that component.

Question 5

Correct

Mark 1.00 out of 1.00

Consider the following classes with their allocated responsibilities:

BookingManager class:

- Implement the system operations related to making bookings

Booking class:

- Know the booking time
- Know the booked service
- Know the Member who the booking is for

Service class:

- Know the service description
- Know the price of the service

Member class:

- Know the member's details

Calendar class:

- Know the bookings
- Add and remove bookings
- Enforce business rules related to making bookings

Which of the following statements best describes which class should have the responsibility for creating a new Booking instance?

- ☐ a. The responsibility for creating a new booking should be allocated to the Member class, as Booking collaborates with Member.
- ☐ b. The responsibility for creating a new booking should be allocated to the BookingManager, as the BookingManager receives the request.
- ☐ c. The responsibility for creating a new booking should be allocated to the Booking class.
- ☒ d. The responsibility for creating a new booking should be allocated to the Calendar class, as Calendar maintains the set of bookings. ✓
- ☐ e. The responsibility for creating a new booking should be allocated to the database.
- ☐ f. The responsibility for creating a new booking should be allocated to the Member class, as Members make bookings.

Your answer is correct.

The correct answers are: The responsibility for creating a new booking should be allocated to the Booking class., The responsibility for creating a new booking should be allocated to the Calendar class, as Calendar maintains the set of bookings.

Question 6

Incorrect

Mark 0.00 out of 1.00

A component that is intended to be reusable in many different contexts shall be designed to satisfy which design principles? (2 correct answers)

- ☒ a. Low Coupling ✓
- ☒ b. Liskov's Substitution Principle ✗
- ☐ c. High Coupling
- ☐ d. High Cohesion

Your answer is incorrect.

The correct answers are: High Cohesion, Low Coupling

Question 7

Correct

Mark 1.00 out of 1.00

Match the principles to their meaning:

We should design components so that new functions can be added without rewriting existing code

The Open-Closed principle



We should orient dependencies so that the volatile parts depend on stable parts

The Dependency Inversion principle



We should design components so that users of the components don't need to know how the component is implemented

The Liskov's Substitution principle



we should aim to separate independently reusable parts into different components.

The Interface Segregation principle



Your answer is correct.

The correct answer is:

We should design components so that new functions can be added without rewriting existing code

→ The Open-Closed principle,

We should orient dependencies so that the volatile parts depend on stable parts

→ The Dependency Inversion principle,

We should design components so that users of the components don't need to know how the component is implemented

→ The Liskov's Substitution principle,

we should aim to separate independently reusable parts into different components.

→ The Interface Segregation principle

Question 8

Incorrect

Mark 0.00 out of 1.00

Suppose the Calendar class creates an instance of the Meeting class.

Which of the following statements is true in this scenario:

- ☒ a. The Meeting class is coupled to the Calendar class.
- ☐ b. The Calendar must also create an instance of the Attendee class.
- ☐ c. Low coupling demands that there is another object that creates both the Calendar and the Meeting instances.
- ☐ d. The Calendar class is coupled to the Meeting class.
- ☐ e. The scenario violates the Dependency Inversion principle because the Calendar is updated more frequently than the Meeting.



Your answer is incorrect.

The correct answer is: The Calendar class is coupled to the Meeting class.

Question 9

Incorrect

Mark 0.00 out of 1.00

Consider the following classes with their allocated responsibilities:

BookingManager component:

- Implement the system operations related to making bookings

Booking component:

- Know the booking time
- Know the booked service
- Know the Member who the booking is for

Service component:

- Know the service description
- Know the price of the service

Member component:

- Know the member's details

Calendar component:

- Know the bookings
- Add and remove bookings
- Enforce business rules related to making bookings

- ☒ a. The responsibility for calculating the list of bookings for a Member should be allocated to a new class, as none of the classes in this design can take on additional responsibilities. ❌
- ☐ b. The responsibility for calculating the list of bookings for a Member should be allocated to the Member component, as each Member is associated with a Booking.
- ☐ c. The responsibility for calculating the list of bookings for a Member should be allocated to the database, as it stores all the Bookings and Members.
- ☐ d. The responsibility for calculating the list of bookings for a Member should be allocated to the Booking component, as each Booking is associated with a Member.
- ☐ e. The responsibility for calculating the list of bookings for a Member should be allocated to the Calendar component, as Calendar has the needed information.
- ☐ f. The responsibility for calculating the list of bookings for a Member should be allocated to the BookingManager component, as the BookingManager implements the system operations.

Your answer is incorrect.

The correct answer is: The responsibility for calculating the list of bookings for a Member should be allocated to the Calendar component, as Calendar has the needed information.

Question 10

Correct

Mark 1.00 out of 1.00

Consider the following classes with their allocated responsibilities:

BookingManager class:

- Implement the system operations related to making bookings

Booking class:

- Know the booking time
- Know the booked service
- Know the Member who the booking is for

Service class:

- Know the service description
- Know the price of the service

Member class:

- Know the member's details

Calendar class:

- Know the bookings
- Add and remove bookings
- Enforce business rules related to making bookings

Which of the following statements best describes which class should have the responsibility for retrieving Bookings from the database?

- ☐ a. The responsibility for retrieving Booking information from the database should be assigned to the BookingManager class.
- ☐ b. The responsibility for retrieving Booking information from the database should be assigned to the Calendar class.
- ☐ c. The responsibility for retrieving Booking information from the database should be assigned to the Booking class.
- ☒ d. The responsibility for retrieving Booking information from the database should be assigned to new class responsible for accessing this data. ✓

Your answer is correct.

The correct answer is: The responsibility for retrieving Booking information from the database should be assigned to new class responsible for accessing this data.

Question 11

Correct

Mark 1.00 out of 1.00

Match the principles to their definitions:

We should try to minimise the degree to which components in a systems are directly connected

Loose Coupling ✓

We should allocate responsibilities that change for different reasons to different components

The Single Responsibility principle ✓

We should aim for components with strongly related responsibilities

The High Cohesion principle ✓

Your answer is correct.

The correct answer is:

We should try to minimise the degree to which components in a systems are directly connected

→ Loose Coupling,

We should allocate responsibilities that change for different reasons to different components

→ The Single Responsibility principle,

We should aim for components with strongly related responsibilities

[Dashboard](#) > [INFS 2044](#) > [Week 4 - Implementation Design](#) > [Quiz 4](#)



