

14 Lecture - CS304

Important Subjective

1. **What is composition in object-oriented programming?**

Answer: Composition is a way of creating complex objects by combining simpler objects or data types. It is a type of association between classes where one class contains an instance of another class as a member variable.

How is composition different from inheritance?

Answer: Composition is a type of association between classes, while inheritance is a way of inheriting properties and behaviors from a parent class. Composition allows for more flexibility and is often used to create objects with complex behavior.

What is the purpose of using composition in object-oriented programming?

Answer: The purpose of using composition is to create objects with complex behavior by combining simpler objects or data types. This allows for greater flexibility and code reuse.

How is composition represented in a UML class diagram?

Answer: Composition is represented in a UML class diagram with a dashed line and an arrow pointing to the contained class.

Can a class have multiple instances of another class as member variables in composition?

Answer: Yes, a class can have multiple instances of another class as member variables in composition.

How does composition affect memory management?

Answer: When using composition, the lifetime of the contained object is managed by the containing object. This means that the contained object is automatically destroyed when the containing object is destroyed.

What happens to the contained object when the containing object is destroyed in composition?

Answer: The contained object is automatically destroyed when the containing object is destroyed in composition.

What is the difference between a strong and weak composition relationship?

Answer: In a strong composition relationship, the containing object has exclusive ownership of the contained object, and the contained object cannot exist without the containing object. In a weak composition relationship, the containing object has a reference to the contained object, but the contained object can exist independently.

How does composition support encapsulation?

Answer: Composition supports encapsulation by allowing the containing object to encapsulate the behavior and data of the contained object.

What are some real-world examples of composition?

Answer: Real-world examples of composition include a car's engine and transmission, a

computer's motherboard and processor, and a house's rooms and furniture.