5 Lecture - CS506

Important Subjective

Certainly, here are 10 short subjective questions related to the concept of Inheritance in Object-Oriented Programming along with their answers:

Question 1: What is inheritance in OOP, and why is it important?

Answer: Inheritance is a fundamental OOP concept where a class (subclass/derived) inherits attributes and methods from another class (superclass/base). It promotes code reuse, hierarchy, and the creation of specialized classes while maintaining a common structure.

Question 2: Explain the terms "superclass" and "subclass" in inheritance.

Answer: A superclass is the parent class from which attributes and methods are inherited. A subclass is a child class that extends a superclass, inheriting its characteristics and possibly adding new attributes or behaviors.

Question 3: How does method overriding work in inheritance?

Answer: Method overriding occurs when a subclass provides its implementation for a method that's already defined in its superclass. The method in the subclass has the same name, return type, and parameters, allowing the subclass to customize or extend the inherited behavior.

Question 4: What is the difference between method overloading and method overriding?

Answer: Method overloading involves defining multiple methods in a class with the same name but different parameters. Method overriding is about providing a specific implementation for a method that's inherited from a superclass.

Question 5: How does inheritance support the "is-a" relationship?

Answer: Inheritance models the "is-a" relationship, where a subclass is a specialized version of its superclass. For example, a "Car" is a "Vehicle." This relationship promotes a natural hierarchy in class design.

Question 6: Can a subclass inherit private members (attributes or methods) from its superclass?

Answer: No, a subclass cannot directly access private members of its superclass. Private members are only accessible within the class where they are defined.

Question 7: Explain the term "constructor chaining" in the context of inheritance.

Answer: Constructor chaining refers to the process of invoking constructors of both the subclass and superclass during object creation. The subclass constructor uses the "super" keyword to call the superclass constructor, ensuring proper initialization of both classes.

Question 8: What is the role of the "super" keyword in inheritance?

Answer: The "super" keyword is used to refer to the superclass within a subclass. It is commonly used to call the superclass's constructor or methods, ensuring proper initialization or accessing overridden methods.

Question 9: How does multiple inheritance differ from single inheritance?

Answer: Single inheritance involves a class inheriting from only one superclass. Multiple inheritance allows a class to inherit from multiple superclasses, which can introduce complexities and conflicts when inheriting attributes and methods from multiple sources.

Question 10: What challenges can arise from excessive use of inheritance in software design?

Answer: Excessive inheritance can lead to overly complex hierarchies and tightly coupled classes. It might also result in a situation known as the "diamond problem," where a class inherits from two classes that share a common superclass.