

23 Lecture - CS304

Important Subjective

1. **What is the scope resolution operator and how is it used to access base class member functions in a derived class?**

Answer: The scope resolution operator (::) is used to specify the scope of a member function or variable. To access a base class member function in a derived class, the derived class can use the scope resolution operator followed by the name of the function and the base class name.

Can a derived class access a private member function of its base class? If so, how?

Answer: No, a derived class cannot access a private member function of its base class directly. It can only access the base class member function through a public or protected member function or by declaring the derived class as a friend class of the base class.

How does method overriding work in a derived class?

Answer: Method overriding is when a derived class defines a member function with the same name as a member function in the base class. The derived class can modify the behavior of the base class member function by providing a new implementation. When the derived class object calls the member function, the derived class implementation is executed.

Can a derived class modify a base class member function?

Answer: No, a derived class cannot modify a base class member function directly. It can only modify the behavior of the base class member function by providing a new implementation in the derived class.

What is virtual function and how is it used in a derived class to override a base class member function?

Answer: A virtual function is a member function in a base class that can be overridden by a member function in a derived class. To override a base class virtual function, the derived class should provide a new implementation of the virtual function with the same signature as the base class virtual function.

How is the order in which base classes are specified in a derived class declaration related to the order in which their constructors are called?

Answer: The order in which base classes are specified in a derived class declaration determines the order in which their constructors are called. The base class constructors are called in the order they are specified in the derived class declaration, regardless of the order in which they are inherited.

Can a derived class access private member variables of its base class?

Answer: No, a derived class cannot access the private member variables of its base class directly. It can only access them through public or protected member functions or by declaring the derived class as a friend class of the base class.

What is the difference between public, private, and protected inheritance?

Answer: Public inheritance makes public members of the base class accessible in the derived

class, protected inheritance makes protected members of the base class accessible in the derived class, and private inheritance makes both public and protected members of the base class private in the derived class.

How does a derived class call a base class constructor?

Answer: A derived class can call a base class constructor explicitly in its own constructor by using the base class name followed by the constructor arguments in the member initializer list of the derived class constructor.

Can a derived class have its own private member functions and variables in addition to those inherited from the base class?

Answer: Yes, a derived class can have its own private member functions and variables in addition to those inherited from the base class.