

31 Lecture - CS304

Important Subjective

1. What is multiple inheritance and how does it differ from single inheritance?

Answer: Multiple inheritance is a feature in object-oriented programming that allows a child class to inherit properties and behaviors from multiple parent classes. In contrast, single inheritance only allows a child class to inherit from one parent class.

What is the diamond problem in the context of multiple inheritance and how can it be resolved?

Answer: The diamond problem is a conflict that arises when a child class inherits from two or more classes that have a common ancestor. It can be resolved using virtual inheritance, which ensures that only one instance of the common ancestor class is included in the object hierarchy.

What is method resolution order (MRO) in multiple inheritance and how is it determined?

Answer: Method resolution order (MRO) is the order in which a Python interpreter searches for methods in a multiple inheritance hierarchy. It is determined using the C3 algorithm, which takes into account the order of parent classes and their method definitions.

What is the role of the super() function in multiple inheritance?

Answer: The super() function is used to call a method of a parent class in a child class. It is often used to access and modify the behavior of the parent class method in a child class.

How can you prevent method name conflicts in multiple inheritance?

Answer: Method name conflicts in multiple inheritance can be prevented by using unique method names, or by using the super() function to modify the behavior of the inherited methods.

What is the difference between mixins and multiple inheritance?

Answer: Mixins are a type of multiple inheritance that are used to add functionality to a class without creating a new class hierarchy. In contrast, multiple inheritance involves creating a new class hierarchy by inheriting from multiple parent classes.

How does multiple inheritance affect code readability and maintainability?

Answer: Multiple inheritance can make code more complex and difficult to read and maintain, especially when there are conflicts between inherited methods and attributes. However, it can also improve code flexibility and modularity when used appropriately.

What is the order of constructor execution in multiple inheritance?

Answer: The order of constructor execution in multiple inheritance is determined by the method resolution order (MRO). The constructor of the first parent class in the MRO is executed first, followed by the constructors of subsequent parent classes in the order specified by the MRO.

What are the potential downsides of using multiple inheritance?

Answer: The potential downsides of using multiple inheritance include increased complexity, potential conflicts between inherited methods and attributes, and reduced code readability and maintainability.

How can you use multiple inheritance to create a more flexible class hierarchy?

Answer: Multiple inheritance can be used to create a more flexible class hierarchy by allowing a child class to inherit properties and behaviors from multiple parent classes. This can help to create more modular and reusable code, as well as allowing for the creation of more complex and diverse behavior. However, it is important to carefully design and implement the class hierarchy to avoid conflicts and maintain code readability and maintainability.