Report about the difference between shallow and deep copying

Deep Copy :

- A deep copy creates a new compound object before inserting copies of the items found in the original into it in a recursive manner.
- It will first construct a new collection object and then recursively populate it with copies of the child objects found in the original. It means that any changes made to a copy of the object do not reflect in the original object.
- Syntax of python deepcopy → ***copy.deepcopy()***

Shallow Copy :

- A shallow copy creates a new object but retains references to the objects contained within the original. It only copies the top-level structure without duplicating nested elements.
- Changes made to a copy of an object do reflect in the original object.
- Syntax of python shallowcopy → ***copy.copy()***

Comparsion between Shallowcopy and Deepcopy :

- Shallow copies are faster and use less memory but can lead to unintended changes. Deep copies use more memory and processing time but ensure the copied list is fully independent.

Report about Multiple Inheritance

What is Multiple Inheritance?

Multiple Inheritance in programming is like a family tree for your code. You have a bunch of classes, which are like different members of a family. When one class inherits from another, it's like a child inheriting traits from their parents. The child class gets all the **abilities** (methods) and **properties** (attributes) of the parent class, and can also have some of its own. This helps to keep your code organized, saves time, and makes it easier to understand and manage.

Advantages of Multiple Inheritance :

- **Enhanced code reusability**: A class can inherit different attributes and methods from multiple base classes, thus improving code reusability.
- **Flexibility**: Adding multiple inheritances to the code can make a class possess characteristics from multiple parent classes, thereby making the class more flexible and capable of meeting various requirements.
- **Implementing multiple interfaces**: A class can simultaneously implement multiple interfaces, allowing it to be used in different contexts.

Disadvantages of Multiple Inheritance :

- **Increased complexity**: Multiple inheritances can make the relationship between classes more complicated, increasing the cost of understanding and maintaining the code.

- **Naming conflicts**: If multiple base classes have members with the same name, it may lead to naming conflicts and result in code errors.
- **Difficulty in understanding**: Multiple inheritance increases the complexity of the code, making it difficult to understand and debug.

What will happen if the child and the parent have the same method?

If both a parent and a child class define a method with the same name, the method in the **child class overrides** the one in the parent class. This allows the child to customize or extend the behavior of the parent method.

What Happens if Two Parents Have the Same Parent?

This creates what is known as the **diamond problem**. If the same method or attribute exists in a shared ancestor, it could be unclear which version should be used. Programming languages handle this differently. **Python** uses a rule called **Method Resolution Order (MRO)** to determine the order in which classes are searched.