Report on Multiple Inheritance and Method Resolution in Python

1. Multiple Inheritance

Definition:
Multiple inheritance is a feature in Python that allows a class to inherit attributes and methods from more than one parent class. This enables a child class to combine functionality from multiple sources.
Example:

```
class A:
pass

class B:
pass

class C(A, B):
pass
```

Key Point:
Python uses the Method Resolution Order (MRO) to determine the order in which it searches for attributes and methods when multiple parents are involved.

2. Child and Parent Having the Same Method

When both the child class and its parent define a method with the same name, the child’s method overrides the parent’s method.
Example:

```
class Parent:
def greet(self):
print('Hello from Parent')

class Child(Parent):
def greet(self):
print('Hello from Child')

obj = Child()
obj.greet() # Output: Hello from Child
```

Reason:
Python starts searching for methods in the child class first. If found, it stops without checking the parent.

3. Two Parents Having the Same Method

If two parent classes define a method with the same name, the child class’s method search order depends on the order of inheritance in the class definition.
Example:

```
class A:
def show(self):
print('From A')

class B:
```

```
def show(self):
print('From B')

class C(A, B):
pass

obj = C()
obj.show() # Output: From A
```

Explanation:
Since A is listed first in 'class C(A, B)', Python checks A before B. The search order is determined by the MRO.
Checking the MRO:

```
print(C.mro())
```

Example output: [, ,]

4. Method Resolution Order (MRO)

Definition: MRO is the sequence in which Python looks for a method or attribute in a hierarchy of classes when multiple inheritance is involved.
Rules:
- Search starts from the calling class.
- Moves left to right in the inheritance list.
- Follows depth-first search, avoiding repeated visits.

5. Summary Table

Scenario	Behavior in Python
Multiple inheritance	Child inherits from multiple parents.
Child & parent have same method	Child’s method overrides parent’s method.
Two parents have same method	First listed parent in child definition takes priority.
Method search order	Determined by MRO.

6. Shallow Copy vs Deep Copy

Shallow Copy:
Shallow repetition is quicker. However, it's 'lazy' — it handles pointers and references. Rather than creating a contemporary copy of the particular knowledge the pointer points to, it simply copies over the pointer value. So, both the original and the copy will have pointers that reference the same underlying data.

Deep Copy:
Deep repetition truly clones the underlying data. It is not shared between the original and the copy. This means that modifications to one will not affect the other.