Report 2: Multiple Inheritance in Python

Introduction

Inheritance is a core concept of object-oriented programming that allows a class to acquire properties and methods from another class. Multiple inheritance is a feature in Python where a class can inherit from more than one parent class. This enables code reuse and the combination of behaviors from multiple sources but can also lead to method resolution conflicts.

When Child and Parent Have the Same Method

If the child class defines a method with the same name as one in its parent class, the child’s version overrides the parent’s method. This is known as method overriding. Python will always look for the method in the child class first before searching the parent classes.

When Two Parents Have the Same Method

If both parent classes define a method with the same name and the child does not override it, Python determines which method to use based on the Method Resolution Order (MRO). The MRO follows a left-to-right search order as declared in the class definition.

When Two Parents Share the Same Parent (Diamond Problem)

In cases where both parent classes inherit from a common ancestor, ambiguity can occur—commonly referred to as the Diamond Problem. Python resolves this ambiguity using the C3 linearization algorithm, which defines a consistent MRO. Developers can inspect the MRO using ClassName.**\_\_mro\_\_**.

Conclusion

Multiple inheritance in Python is a powerful feature that allows for greater flexibility in code reuse. However, it can lead to method conflicts, especially in cases where different parents define the same method or share a common ancestor. Python’s MRO and the use of super() help manage these conflicts, making multiple inheritance both practical and predictable.