1. The **diamond problem** occurs when a class inherits from two parent classes that share a common base class. This creates ambiguity about which version of a method to use.

In Java, this issue is avoided because **multiple inheritance is not allowed** with classes. However, it can occur when using interfaces. So using interface implement the following scenario:

**Base Class/ Interface**: Product (attributes: name, price; method: showDetails()).

**Derived Classes/Interfaces**:

Electronics: Adds warranty and overrides showDetails().

Clothing: Adds size and overrides showDetails().

**Child Class/Interface**: ElectronicsClothing inherits both Electronics and Clothing.

1. Imagine a hierarchical structure for a library system:

Base class: LibraryItem (attributes: title, author).

Subclasses:

Book (adds numberOfPages attribute).

Magazine (adds issueNumber attribute).

Write a program to demonstrate polymorphism using this hierarchy.