In my program, I exemplify Object-Oriented Programming (OOP) through its use of encapsulation, inheritance, and polymorphism, three core principles of OOP.

Encapsulation: The program encapsulates data and methods within classes like Entity, Hero, Creature, and Bag. Each class bundles related attributes (like name, health, damage) and methods (such as attack(), level\_up()) into a single unit. This encapsulation secures data and functionalities within classes, allowing controlled access and modification.

Inheritance: My program demonstrates inheritance, where Hero and Creature classes inherit from the Entity class. This means they acquire all attributes and methods of Entity. Inheritance is also seen in the Hero class’s ability to extend or modify inherited behaviors, like in the overridden level\_up() method. This showcases the 'is-a' relationship, with Hero and Creature being specialized forms of Entity.

Polymorphism: Polymorphism in this program is observed in the method overriding. The Hero class overrides the level\_up() method from the Entity class, providing specialized behavior while maintaining a consistent interface. This allows for flexibility in treating different objects (Hero, Creature) as general Entity objects, enabling the use of generic methods like attack() on them, regardless of their specific class types.

By integrating these OOP concepts, the program effectively creates a structured and flexible codebase where objects interact with each other through well-defined interfaces, demonstrating the power and utility of the OOP paradigm.