1. What are the advantages of Polymorphism?
   - Polymorphism allows for the same method to be used for different types of objects, reducing the need for duplicate code. It also provides flexibility by allowing the same code to be used with different types of objects, making the code more adaptable and easier to maintain. Additionally, polymorphism can make the code more readable and easier to understand, as it allows for a more natural and intuitive representation of the problem domain.
2. How is Inheritance useful to achieve Polymorphism in Java?
   - In Java, inheritance is useful for achieving polymorphism by allowing a subclass to inherit the methods and behaviors of its superclass. This enables a subclass to use the methods of its superclass and also override those methods to provide its own implementation. Through method overriding, different subclasses can have different implementations of the same method, thus achieving polymorphism.
3. What are the differences between Polymorphism and Inheritance in Java?
   - In Java, inheritance is a mechanism for creating new classes based on existing ones, allowing for code reuse and establishing a class hierarchy.
   - Polymorphism is the ability of a reference variable to refer to different types of objects and to invoke methods specific to the type of object it refers to, enabling dynamic behavior at runtime.
   - Inheritance is related to the class hierarchy and code reuse, while polymorphism is related to the ability to perform different actions based on the object being operated upon.