1. What are the advantages of Polymorphism?

- Allows objects of various classes to be handled as objects of a common class, increasing the reusability of code.
- Reduces the quantity of code that needs to be written and maintained, which improves readability and maintainability of code.
- Allows for dynamic binding, which allows the appropriate method to be invoked depending on the object's actual class at runtime.
- Makes it easier to develop generic code that can handle objects of diverse types by enabling objects to be treated as a single type.

2. How is Inheritance useful to achieve Polymorphism in Java?

The ability to carry out a single action in multiple ways is known as polymorphism in Java.

Since inheritance happens when there is a relationship between several classes, inheritance is helpful in achieving Variation in Java: One of Java's strongest features is inheritance. Java Inheritance allows a class to inherit the characteristics and properties of its parent class. We can use these inherited attributes to carry out various activities in Java thanks to polymorphism. As a result, we can do the same task in numerous ways.

3. What are the differences between Polymorphism and Inheritance in Java?

While they are connected, polymorphism and inheritance are two different ideas. While inheritance creates new classes based on preexisting ones by inheriting their methods and properties, polymorphism uses the same name or interface for multiple actions or activities. Overriding or overloading methods can be used to achieve polymorphism, while the extends or implements keywords are used to accomplish inheritance. In addition, inheritance can be single or numerous (with the exception of Java, which supports interfaces), and polymorphism can be either static or dynamic. Inheritance helps you to create a hierarchy of classes to structure your code, while polymorphism allows you to write more succinct and expressive code.