1. **Advantages of Polymorphism**:
    - **Code Reusability**: Polymorphism allows us to perform a single action in different ways. So, we can write methods that behave differently based on the object that it is acting upon.
    - **Code Flexibility**: With polymorphism, we can design and implement systems that are easily extensible. New classes can be added with little or no modification to the existing code, making the system more flexible.
    - **Separation of Interfaces from Implementation**: Different classes can be used with the same interface, each providing their own implementation. This separation of interfaces from their implementation can enhance code robustness and hide complexity from the user.

2. **Inheritance and Polymorphism in Java**:
    - Inheritance in Java is a mechanism where one class acquires the properties (fields) and behaviors (methods) of another class. The class which inherits the properties of other is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class).
    - Inheritance is useful for polymorphism because it allows a subclass to be treated as its superclass. This is the key principle that allows for polymorphism. In Java, a superclass reference variable can point to a subclass object. This allows Java to make method calls at runtime based on the actual object, rather than the reference type, a feature known as dynamic method dispatch.

3. **Differences between Polymorphism and Inheritance in Java**:
    - **Purpose**: Inheritance is a concept that allows the creation of hierarchical classifications. It provides a way to create a new class from an existing class. On the other hand, Polymorphism is a concept that deals with allowing a method or an operator to be used in multiple ways, depending on its input.
    - **Functionality**: Inheritance allows properties and methods from one class to be inherited by another class. Polymorphism, on the other hand, allows methods to be used in multiple ways, depending on the object that is calling the method.
    - **Usage**: Inheritance is used when there is an “is-a” relationship between classes, while Polymorphism is used when there is an “it can be” relationship.
    - **Method Overriding and Overloading**: Method overriding is a perfect example of polymorphism, and it occurs when a subclass provides a specific implementation of a method that is already provided by its parent class. Method overloading is not an example of polymorphism, but it allows different methods to have the same name but different parameters.