1. What are the advantages of Polymorphism?

* Through polymorphism, methods defined in a base class can be overridden by derived classes, allowing for behaviors to be redefined while maintaining a common interface. This promotes code reusability, as one can build upon existing implementations without the necessity to duplicate code.
* **Abstraction of Implementation:** Polymorphism abstracts the implementation details and exposes a uniform interface. As a result, the code interacting with objects can effectively operate on the interface level without concern for the underlying implementations.
* **Increased Maintainability:** It simplifies the maintenance of the codebase by localizing behavioral modifications to specific subclasses, hence minimizing potential side-effects and regression in the system.
* **Extension without Modification:** Existing systems can be extended with new functionality by adding new subclasses and not modifying the existing ones.
* **Decoupling of Components:** Polymorphism aids in the decoupling of system components, which as a result, leads to a reducement in the rigidity and fragility of the software structure.
* **Simplicity:** Polymorphism simplifies complex code, making it more intuitive.

2.

Inheritance lets subclasses inherit attributes and methods of a superclass and having its own attributes and methods. Polymorphism makes use of these methods to operate various tasks.

3.

* **Concept:**  
     - **Inheritance** is all about creating a new class that borrows properties and behaviors from an existing class , while **polymorphism** is about performing a single action in different ways.
* **Purpose:**  
     - Inheritance aims for a logical hierarchical relationship between a base class and derived classes, promoting code reuse and the beauty of a well-organized structure. Polymorphism allows for flexibility and the ability to call the same method on different objects and have each of them respond in their own way.
* **Behavior:**
* With Inheritance, subclasses inherit state and behavior (fields and methods) from the superclass, which means they're tightly coupled in terms of features. Polymorphism is more about behavior than state. It enables objects of different classes to be treated as objects of a common superclass.
* **How They're Achieved:**  
     - Inheritance is achieved in Java using the keyword “Extends” primarily (and “implements” for interfaces). Polymorphism is commonly implemented through method overriding (runtime polymorphism) and method overloading (compile-time polymorphism).
* **Flexibility and Coupling:**  
     - Inheritance can lead to tight coupling if not used judiciously because the base and derived classes are directly linked. Polymorphism fosters loose coupling by allowing inter-class relationships through interfaces and abstract classes, promoting more maintainable and scalable code.