Reading Assignment

Please answer three questions below:

- What are the advantages of Polymorphism?

- How is Inheritance useful to achieve Polymorphism in Java?

- What are the differences between Polymorphism and Inheritance in Java

- **Advantages of Polymorphism:**
  - **Flexibility and Extensibility:** Polymorphism allows for flexibility in code design by enabling the use of a single interface to represent different types. This makes it easier to extend and modify code without affecting existing functionality.
  - **Code Reusability:** Polymorphism promotes code reusability as methods can be written to accept the base type, and they can work with any derived type that inherits from the base type.
  - **Readability and Maintenance:** Code that uses polymorphism tends to be more readable and maintainable. It allows developers to work with abstractions rather than concrete implementations, leading to cleaner and more modular code.
- **How Inheritance is Useful to Achieve Polymorphism in Java:**
  - Inheritance is a key mechanism for achieving polymorphism in Java.
  - Through inheritance, a subclass can inherit the properties and behaviors of its superclass.
  - Polymorphism allows a reference variable of a superclass to refer to an object of its subclass, enabling the use of a single interface (superclass) to represent different types (subclasses).
  - The overridden methods in the subclasses provide the actual implementation for polymorphic behavior.
- **Differences between Polymorphism and Inheritance in Java:**
  - **Definition:**
    - **Polymorphism:** It refers to the ability of a single entity to take different forms. In Java, it often involves methods being able to perform different actions based on the object they are acting upon.
    - **Inheritance:** It is a mechanism where a new class inherits properties and behaviors (fields and methods) from an existing class. It establishes a relationship between a superclass and its subclasses.
  - **Purpose:**
    - **Polymorphism:** It enhances code flexibility and allows objects of different types to be treated as objects of a common type.
    - **Inheritance:** It facilitates code reuse by allowing a new class to use the properties and behaviors of an existing class.
  - **Usage:**
    - **Polymorphism:** It is achieved through method overriding, interfaces, and method overloading.
    - **Inheritance:** It is the mechanism by which one class inherits properties and behaviors from another class.
  - **Relationship:**
    - **Polymorphism:** It can be achieved without inheritance, using interfaces and method overriding.

- **Inheritance:** It is often a prerequisite for achieving polymorphism, especially in the context of method overriding.