Question 1: What are the advantages of Polymorphism?

- Improve code usability as objects of multiply child classes can be treated as objects of a common parent class.
- Reducing amount of code that need to be implement in multiple classes and instead, we can group it and put it into a parent class and have its child inherit it.
- Allow object's behaviors can be changed at runtime depending on the context and method uses.
- Reduce complexity of code by allowing the use of the same method name for related functionality, make the code easier to read and maintain.
- New child classes can be created to extend the functionality of the superclass without modifying the existing codes.
- Increase efficiency with compile-time polymorphism.

Question 2: How is Inheritance useful to achieve Polymorphism in Java?

Inheritance allows subclasses to inherit attributes and methods of a superclass as well as having attributes and methods of its own. Polymorphism takes advantage of this to use those methods to perform different tasks.

Question 3: What are the differences between Polymorphism and Inheritance and Java?

- Inheritance is a mechanism where a new class is derived from an existing class.
- Polymorphism is a mechanism where objects of different classes can be treated as objects of a common superclass.
- Inheritance creates a structured class system based on relationships between general and more specific classes, while polymorphism uses this structure to enable flexible and interchangeable object interactions.
- Inheritance uses a defining relationship, for example, a dog and a cat are both considered animals.
- Polymorphism use an implementation relationship, for example, a man can be treated as a husband or an employee.
- Inheritance is static and defined at the time of class creation.
- Polymorphism is dynamic and its methods are determined at runtime.