LAB03 READING ASSIGNMENT

1. **What are the advantages of Polymorphism?**

- *Code Reusability*: Methods in the superclass can be reused by different subclasses, reducing code duplication.
- *Flexibility*: Code can work with objects of various types through a common interface, making it adaptable.
- *Maintainability*: Reduces code repetition, so updates and fixes are easier to apply.
- *Extensibility*: New classes can be added without changing existing code, making the system easier to expand.
- *Simplified Code*: Polymorphism reduces the need for complex conditionals or repetitive code, making the program easier to read and understand.

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

Inheritance allows a subclass to inherit methods and properties from a superclass. It enables different subclasses to be treated as instances of the superclass, which is a key to achieve *polymorphism*.

In short, inheritance helps achieve polymorphism in Java by allowing subclasses to override methods of a parent class. A parent class reference can point to subclass objects, and at runtime, the correct overridden method is called, making the code more flexible, reusable, and maintainable.

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

	Inheritance	*Polymorphism*
Definition	Allows a subclass to inherit properties and behaviors from a superclass.	Allows objects of different classes to be treated as objects of a common superclass.
Purpose	To promote code reuse by allowing subclasses to share the same implementation of methods and properties.	To provide flexibility, allowing the same method or function to behave differently based on the object type.
Usage	Used to create a class hierarchy and share common behavior across multiple classes.	Used to enable different classes to have their own implementation of the same method, providing runtime flexibility.
Type	A structural concept.	A behavioral concept.
Dependency	Inheritance can exist without polymorphism.	Polymorphism depends on inheritance.
Main concept	"Is-a" relationship — a subclass is a more specific version of the superclass.	"Many forms" — allows one method or interface to be implemented differently depending on the object.