Pre Lab – 4 Mohammed sheikh 1221541 3-12-2023

- 1 - No, you cannot create an instance of an interface using new A() .

- 2 - Yes, you can declare a reference variable with type A for an interface A .

- 3 - True, which is required by the Comparable interface.

- 4 - No, a class cannot invoke super.clone() if it does not implement java.lang.Cloneable.

- 5 - Yes, Therefore, you can invoke clone() on a Date object.

- 6 - Multiple Inheritance: Java supports multiple interface inheritance, allowing a class to implement multiple interfaces. However, a class can extend only one abstract class. This flexibility in inheritance is a significant advantage of interfaces.

  - Implementation Flexibility: Since a class can implement multiple interfaces, it provides more flexibility in adding functionalities to a class without the limitations of single inheritance.

  - Separation of Concerns: Interfaces separate the specification (what a class must do) from the implementation (how it does it). This promotes a clear separation of concerns and makes it easier to understand and maintain code.

- 7 - Similarities between abstract classes and interfaces:

  - Both can declare abstract methods that must be implemented by concrete subclasses or implementing classes.

  - Both can have methods with a default implementation (with the default keyword in interfaces or non-abstract methods in abstract classes).

  Differences between abstract classes and interfaces:

  - An abstract class can have instance variables (fields), whereas an interface cannot have instance variables until Java 8 (with the introduction of default methods).

  - A class can extend only one abstract class, but it can implement multiple Interfaces. - Abstract classes can have constructors, while interfaces cannot.

  - Abstract classes can have access modifiers for their methods, while methods in interfaces are implicitly public.