Mouath Sami Masalmah

No: 1220179

Date: 14/10/2023

“Solution for Pre\_Lap 2 ”

Dr: Fadi khalil

Q1:

“extends”

Q2:

Java supports multiple inheritance for interfaces but not for classes.

Q3:

To invoke an overridden superclass method from a subclass in Java, you use “ **super.methodName() ”** within the subclass method. This calls the superclass method and can be used to extend or modify its behavior in the subclass.

Q4:  
If a method in a subclass has the same signature and return type as a method in its superclass, it is an overridden method, not overloaded. Method overriding allows the subclass to provide its own implementation while maintaining the same method signature and return type from the superclass.

Q5:

(a) False. Superclass's no-arg constructor is invoked only if not explicitly called in the subclass constructor.

(b) False. You cannot override a private method from a superclass in Java.

(c) True. You can always successfully cast a subclass instance to a superclass.

(d) False. Casting a superclass instance to a subclass may result in a runtime exception unless it's a valid subclass instance.

Q6:

To allow access to a class by classes in the same package but restrict access to classes in different packages, you should use the default (package-private) access modifier.

Q7:

You should use the "protected" access modifier.

Q8:

To prevent a class from being extended, use the **`final`** keyword in the class declaration, like this: **`final class ClassName`.**

To prevent a method from being overridden, use the **`final`** keyword in the method declaration in the superclass, like this: **`final void ‘methodName()’.**