1. We use the extends keyword , For example : class Subclass extends Superclass.

2. Java does not support multiple inheritance for classes. This means a class cannot directly inherit from multiple classes. However, it supports multiple inheritance through interfaces.

3. To invoke an overridden superclass method from a subclass, you can use the super keyword. For example, if you have a method superclassMethod in the superclass, you can call it from the subclass using super.superclassMethod().

4. The method is overridden , because it provides a new implementation for the same method

5.

(a) . False . The superclass's no-arg constructor is not always invoked when invoking a constructor from a subclass.

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

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

(d). False. You cannot always successfully cast an instance of a superclass to a subclass without a risk of a ClassCastException.

6. Use the “default” or package – private access or modifier.

7. Use the “protected” access modifier when you want to allow access subclasses but restrict access from other classes outside the package.

8. Using final is the best way to archieve these goals , as it directly and clearly communicates your intent in code. But :

To prevent a method from being extended , use the final keyword before the class declaration . Example : public final class MyFinalClass{}

To prevent a method from begin overridden , use the final keyword before the method declaration in the superclass. Example :

public class MySuperclass{

public final void myFinalMethod(){}