If the reference variable you are working with has a null value (meaning it is not referring to an object), it's in a dangerous state.  Specifically, if you access an instance member with it, you will see the dreaded "NullPointerException."  This is an exception created by the JVM and unless special exception handling code is written, will result in your program quitting prematurely.

For example, the following code will result in a NullPointerException:

public static void main(String[] args) {

String s = null;

s = s.toUpperCase(); // Oops! s is null!

}

Static members are different.  While you shouldn't write sloppy/confusing code like this, you should know (especially for the exam) that accessing static members with a null Class reference will **not** throw a NullPointerException.  For example, the java.lang.Math class has a static method called floor:

public static void main(String[] args) {

// Confusing code, but it works.

Math m = null;

double result = m.floor(28.15);

}

I want to stress, while the preceding code won't throw a NullPointerException, you shouldn't write code like that. It is much clearer to use the class name:

public static void main(String[] args) {

double result = Math.floor(28.15);

}

See the "Exception Handling" section for more details on exceptions.