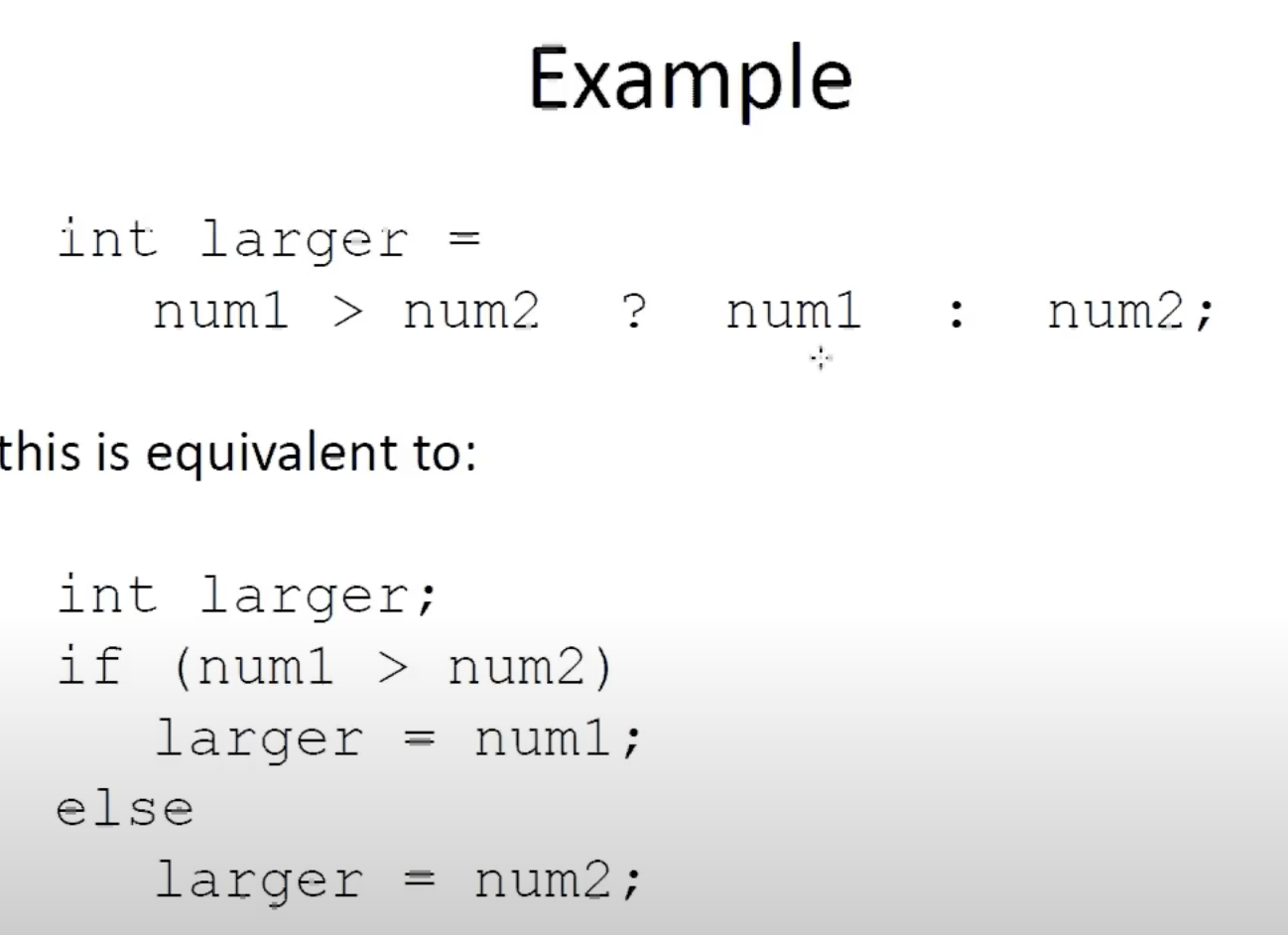
**September 5, 2021: Review**

**Assertions**

* [Programming With Assertions](https://docs.oracle.com/javase/7/docs/technotes/guides/language/assert.html) (Oracle website)
  + Enables you to test your assumptions about your program
  + Each assertion contains a boolean expression that you believe will be true when the assertion executes
  + This allows for debugging within your code - if one of them evaluated to ***false*** you’ll have an assertion error throw
* [Assertion Facility](https://docs.oracle.com/cd/E19683-01/806-7930/6jgp65ikl/index.html)
  + Typically, assertion checking is enabled during program development and testing and disabled for deployment
* [Assertion Facility Chapter from JDK 1.4 Tutorial by Greg M. Travis](https://ccsf.instructure.com/courses/45264/files/6166080/download?wrap=1)

**Review videos**

* [M1 Review 01 Variables and Data](https://www.youtube.com/watch?v=Q1vnIfYf54Y&list=PL5igFWijWBo1D0f-cJHH6jIJujtogqM_D&index=2&ab_channel=jessicamasterssf)
  + Declaring a variable inside of a method means it’s only available within that method
  + Declaring a variable is just giving it a name (the reference variable)
  + Initializing it is giving it a value
  + Java is strongly typed - meaning variables cannot change their types (and it must be stated when you declare them)
  + Objects can be declared as one types but then have a different ACTUAL type (this is polymorphism)
* [M1 Review 02 Conditionals](https://www.youtube.com/watch?v=aGm72WjCkwY&list=PL5igFWijWBo1D0f-cJHH6jIJujtogqM_D&index=3&ab_channel=jessicamasterssf)
  + When using “==” it checks if the aliases are the same - not the underlying value. You’d want to use the ***.equals()*** method to check if two values are the same.
  + The other conditionals like “>” or “<” only work with primitive values → to compare objects, you need to ensure the objects have implemented the **compareTo()** method
  + There’s an alternate syntax to writing conditionals



* [M1 Review 03 Loops](https://www.youtube.com/watch?v=buzbuz7KMqQ&list=PL5igFWijWBo1D0f-cJHH6jIJujtogqM_D&index=4&ab_channel=jessicamasterssf)
  + There are 3 main types: while, do while, for
  + The variables you created inside a while loop are local to that loop (not available outside of it) - further, they’re only available to a certain iteration (they get garbage collected after being used)
  + A **sentinel** value is something you check to see if you should continue the loop (“enter 0 when you are done”)
  + The enhanced for loop is the “For each” loop → for each is implemented in the standard Java arrays and has a leaner syntax than doing the same thing with a for loop
* [M1 Review 04 Methods](https://www.youtube.com/watch?v=HvcaMCev4j8&list=PL5igFWijWBo1D0f-cJHH6jIJujtogqM_D&index=5&ab_channel=jessicamasterssf)