Transcript: Aliases

>> Now let's consider the concept of aliases. Recall in our last slide we had two string reference variables, name1 and name2, that both referenced the same string object, Steve Jobs. Well, when two or more references refer to the same object, we call them aliases. Here's another example. Suppose we have Scanner scan1 = new Scanner(System.in), so now we have the scanner on the keyboard.

And then we declare Scanner scan2 = scan1. Well, both scan1 and scan2 reference the same objects, so they're aliases. Now what that means is, if you change an object using one reference, it's changed for the other reference as well. So if we said scan2.useDelimiter(","), which means we want to think of the token separated by commas instead of white space.

Well, later if you use scan1, it's now gonna use the same delimiter as scan2 because both scan1 and scan2 really reference the same object. Garbage collection occurs automatically in Java. And when a, when an object is no longer referenced, that is, there's no variables pointing to it, it can't be accessed.

So this happens naturally through assignment. If you create a, a string object, for example, and, and set that equal to, say string one, and then come along and assign something else to string one, well, that first object is no longer accessible. In fact, it's essentially useless. So therefore it's called garbage.

And Java performs automatic garbage collection periodically, which means it returns the object's memory to the system for future use. Now languages such as C and C++ require the programmer to perform garbage collection in the form of allocating and deallocating memory. So after you've allocated memory and used it in C, let's say if you forget to deallocate it that creates an issue.

And, and it's called, it's called a memory leak. So if your program runs long enough and it has a memory leak, you eventually run out of memory. So the fact that Java offers automatic garbage collection means it's done away with an entire class of program error, errors, which is a good thing.