CS 240: Implementing an Interface Transcript

This video shows a split screen of Professor Wilkerson on the right and a PowerPoint screen on the left. Any text displayed or action performed that is not verbalized will be included in italics as visual descriptions.

[00:00:00] **JEROD WILKERSON:** Now we'll learn some details about how to implement an interface.

[00:00:04] The way you implement an interface is just use the 'implements' keyword.

[00:00:07] If I create a Person that I want to have implement the Moevable interface, I will just declare the Person class and then use 'implements Moevable', and what that says is this Person now has to either provide all the methods for Moevable, which in our case is just the go method, or it has to be declared as abstract.

[00:00:28] If I want to create an instance of a Person, I'm going to have to create an actual implementation for the go method.

[00:00:36] Now in some cases, we might want a class to implement an interface but not provide the implementation, and we had an example of that in our CitySimulation.

[00:00:48] The Vehicle class, if you remember from the very beginning when I first introduced that example, we wanted it to have a go method, but we didn't know what code to write.

[00:00:56] Well, there's nothing about interfaces that make us suddenly know what code to write, and so we still don't know what code to write for Vehicle.

[00:01:03] This is a case where we need our Vehicle to implement Moevable, so its children, its subclasses, can go in the Moevable array, and we also want programmers who create subclasses of Vehicle to know that they need to write that go method.

[00:01:18] We can make all that happen by having the Vehicle class implement Moevable, but still declaring it abstract.

[00:01:25] That will satisfy the compiler.

[00:01:27] We won't have to write the go method that we don't know how to write for Vehicle, but then programmers who create subclasses of Vehicle will receive that compile error that will tell them that they need to write a go method because their subclasses are also implementing that interface.

[00:01:43] Here is how we can extend multiple interfaces.

[00:01:48] I'm sorry, this shows how we can implement multiple interfaces.

The following code shows how to implement multiple interfaces

```
public class Employee extends Person implements Moveable, Comparable {

    public void go() {

        \\Code make person go

    }

    public int compareTo(Object obj) {

        \\Code to compare two employees

    }

}
```

End of code.

[00:01:52] Now we have our Employee class which extends Person, and it implements two interfaces.

[00:01:57] It implements Moevable and Comparable.

[00:01:59] Notice it’s not abstract.

[00:02:01] That means I have to write implementations for all the methods from all the interfaces that it implements.

[00:02:08] I have to write a go method to satisfy the Moevable interface, and Comparable is an interface that gets used quite a bit.

[00:02:16] It has a compareTo method, and so I would have to write that.

[00:02:19] That allows you to compare two objects to see how they should be sorted.

[00:02:23] When you get into sorting, you’ll learn about the Comparable interface.