CS 240: Design Principles - Single-Responsibility Principle Transcript

[00:00:00] **INSTRUCTOR:** The next principle of good design is called the single responsibility principle. Now the idea with the single responsibility principle is that each class that you create in your code should represent one well-defined concept.

[00:00:16] You don't want to create classes that represent two concepts or three concepts, but you really just want to have a class represent one well-defined concept. And then all the features on that class, the variables and the methods, would be highly related to the concept that that class represents.

Start visual description. Slide titled Single Responsibility/Cohesion. Text reads:

- *Each class and method should have a single responsibility*
- *Each class should represent one, well-defined concept*
    - *All variables and methods in a class should be highly related to the class' single responsibility (cohesion)*
- *Each method should perform one, well-defined task*
    - *Unrelated or loosely related tasks should be in different methods*
- *Methods that need to perform multiple tasks should delegate tasks to sub-methods that each perform a single task*
- *Cohesive classes and methods are easy to name*

End visual description.

[00:00:36] So if you find yourself creating a class that does multiple unrelated things or represents multiple unrelated or even loosely related concepts, you want to take that class and break it up into pieces so that each class just represents one idea or one concept.

[00:00:53] And that's the notion with the single responsibility principle. You can also apply the single responsibility principle to methods or functions in your code. Each function should do one thing. It should do one thing and do it well.

[00:01:08] And it should have a good name that describes what it does. You don't want to write functions that do multiple things. So if I need to accomplish three different tasks, I would rather write three different methods, each of which performs one task, rather than writing one function that does all three tasks.

[00:01:26] And so each function does one thing, does it well, has a good name. Now if I do have a function that actually needs to perform multiple tasks, what I should do is I should break those multiple tasks into individual functions as I described.

[00:01:41] And then the function that needs to perform multiple tasks can then call or delegate to those other functions that only do one thing. And so that's how we keep our code in line with the single responsibility principle.

[00:01:56] Classes and functions that really do one thing, they do and they have a good name that describes what they do or what they are. One of the benefits of single responsibility principle is that if you apply this principle, your class names and your method names become very easy to come up with.

[00:02:15] How do you name something? You name it after the one thing that it is or the one thing that it does. Whereas if you have violations of this principle in your code, it becomes hard to name things because you are not sure, okay, here is a class that does four different things, how should I name it?

[00:02:30] It is not clear what to even call it. And so if you find yourself having trouble naming your classes and your functions, you might want to step back and check to make sure that you are actually applying the single responsibility principle.