The design is better learned than taught, this means that you actively try to solve design problems. And this course is structured around a set of exercises that you should actively work on as you are watching them.

Scenario 1 - Let's jump right in and try to design a text browser application. Imagine the problem of browsing the text in a computer file. Imagine that no graphical user interface, GUI toolkit supplies a single widget to do this. Imagine that you would like to devise a cleanly structured solution. Assume that you do have a GUI library, such as Swing or SWT, it just doesn't have a text browser widget in it. From what atomic GUI components would you build your TextBrowser?

Solution - Okay so for the GUI components of this text browser, we'll need some kind of window to display the text, and I can foresee if the text is too large to fit in that window, we'll have a scroll bar that allows to, to move around in the text document. It turns out that we also need a component that's going to supply us the text. Now, this isn't strictly in the GUI toolkit. But in order to make this to work, we have to access the text somehow. So, we'll call that the file manager component. We're going to make some assumptions. We're going to assume that you cannot hold the contents of the entire file in memory. You're going to have to go to the disk to get it. And assuming you have at the operating level line oriented access to the file. So, in your system libraries, you have a way of, of reading the lines at any time. So, you're going to need to have a module, that when requested can retrieve a limited length, consecutive sequence of the file's lines. And we're also going to assume we don't have to worry about opening the file or closing the file. Just the reading of the file, supplying the lines.

For your window component, we're going to call that a ViewPort, and you need to be able to use it, to display the textual content graphically. And we're going to make some assumptions here, we're going to assume that the ViewPort displays an intern, integer number of lines, and we're going to be, assume that it can be resized to be any length between one and 100 lines. And we’re going to assume that all the text in the same font, is in the same font and has the same point size. So, these are simplifying assumptions to make them, make this particular exercise, you know, small enough to fit in a lesson and allow us to focus on just what the important issues are.

As far as the scrollbar is concerned, scroll bars are one graphical way of supplying numbers to other parts of an application. We’re going to use a traditional scrollbar in which there's it's going to be a vertical scroll bar and it's going to have a movable part of it. That is the user can move a part called the handle which sits in a tray. So, you can move up and down, and we when we use the terms handle and tray to indicate that we can set, the user can set the position of the file by moving this handle up and down on the tray. The handle position denotes that part of the file that should be displayed in the view port. So, when you move it all the way up, you get the start of the file, and when you move it all the way down, you get the end of the file. Also, the size of the handle in proportion to the size of the tray denotes the portion of the file that is visible. So, if all of the file contents fit into the viewport, you'd expect the tray to be, filled up. And if we have a gigantic file, that in a very small window, we'd expect just a thin handle to appear in the scrollbar. So we've, we've come up with three candidate structural elements. Now, let's look at the behavioral side of this TextBrowser. One way to get a handle on behavior is to imagine how the user will use the intended application. We call these descriptions use cases.