## [Introduction and Prerequisites](https://app.pluralsight.com/course-player?clipId=e5395cfb-b473-47a9-92b6-34b8c46d1094)

### [Version Check](https://app.pluralsight.com/course-player?clipId=e5395cfb-b473-47a9-92b6-34b8c46d1094)

### [Introduction](https://app.pluralsight.com/course-player?clipId=42a62500-befc-422c-94fe-023010a35a2e)

[So you've just been in an interview and someone asked you to describe a design](https://app.pluralsight.com/course-player?clipId=42a62500-befc-422c-94fe-023010a35a2e&startTime=1.79) [pattern or maybe you were just describing a problem to a coworker and they](https://app.pluralsight.com/course-player?clipId=42a62500-befc-422c-94fe-023010a35a2e&startTime=6.08) [said it sounds like you're reinventing the wheel.](https://app.pluralsight.com/course-player?clipId=42a62500-befc-422c-94fe-023010a35a2e&startTime=10.75) [They might have even said to you that it sounds like](https://app.pluralsight.com/course-player?clipId=42a62500-befc-422c-94fe-023010a35a2e&startTime=13.68) [you're describing a factory or a singleton.](https://app.pluralsight.com/course-player?clipId=42a62500-befc-422c-94fe-023010a35a2e&startTime=15.8) [If this sounds like a situation that you've been in, then this](https://app.pluralsight.com/course-player?clipId=42a62500-befc-422c-94fe-023010a35a2e&startTime=19.14) [course is for you. Our focus in this course is on presenting the](https://app.pluralsight.com/course-player?clipId=42a62500-befc-422c-94fe-023010a35a2e&startTime=21.99) [design patterns described in the Gang of Four in an](https://app.pluralsight.com/course-player?clipId=42a62500-befc-422c-94fe-023010a35a2e&startTime=25.74) [example‑driven way using Java to do so. Hi. I'm Brian Hansen, and welcome to Creational Design Patterns using Java.](https://app.pluralsight.com/course-player?clipId=42a62500-befc-422c-94fe-023010a35a2e&startTime=28.85)

### [Why Learn Patterns?](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db)

[You might ask yourself why design patterns are important.](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=1.94) [I first learned patterns as a means of communicating](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=5.71) [a problem to another developer.](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=8.74) [You may very well already know how to solve a particular problem and it might](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=11.14) [follow the structure of a pattern, but it's better to have a common vocabulary](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=15.25) [that you can explain to someone what that problem is.](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=19.8) [Patterns are an abstract topic, it isn't something that](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=23.64) [you look at the concept and then have it memorized and](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=27.37) [know how to apply it from there.](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=30.04) [Some patterns are more easily applied to particular problems than another,](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=32.17) [so I suggest revisiting pattern material as you develop](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=36.49) [more and more programming experience.](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=40.81) [Whether it be this course or a book or something else,](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=43.24) [it is amazing how your perception of a pattern will change after gaining more](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=46.76) [experience with it. You might come away with a different understanding of it](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=50.84) [after applying it through different coding practices.](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=54.95) [A lot of people have used a singleton and think they have used design patterns, but there is much more than just a singleton out there.](https://app.pluralsight.com/course-player?clipId=908ec497-62ba-43f3-bb63-deb0e69c89db&startTime=58.86)

### [Pattern Classifications](https://app.pluralsight.com/course-player?clipId=17d522ce-1ecc-4631-8177-897d91bfa70e)

[The Gang of Four breaks patterns out into three groups.](https://app.pluralsight.com/course-player?clipId=17d522ce-1ecc-4631-8177-897d91bfa70e&startTime=1.074) [Those groups are creational, structural, and behavioral.](https://app.pluralsight.com/course-player?clipId=17d522ce-1ecc-4631-8177-897d91bfa70e&startTime=6.699) [This course is going to focus on patterns classified under the creational group.](https://app.pluralsight.com/course-player?clipId=17d522ce-1ecc-4631-8177-897d91bfa70e&startTime=11.074) [Creational parts are focused on, as you might guess, how objects are created.](https://app.pluralsight.com/course-player?clipId=17d522ce-1ecc-4631-8177-897d91bfa70e&startTime=17.074) [There's a lot more than just replacing the keyword new,](https://app.pluralsight.com/course-player?clipId=17d522ce-1ecc-4631-8177-897d91bfa70e&startTime=21.074) [though, when talking about creational patterns.](https://app.pluralsight.com/course-player?clipId=17d522ce-1ecc-4631-8177-897d91bfa70e&startTime=26.074) [The two remaining groups, structural and behavioral, are covered in two separate courses much like this one.](https://app.pluralsight.com/course-player?clipId=17d522ce-1ecc-4631-8177-897d91bfa70e&startTime=29.074)

### [Which Patterns?](https://app.pluralsight.com/course-player?clipId=7337f811-2fa2-49e0-a532-59063bafaf3e)

[The creational patterns that we are going to cover discussed by](https://app.pluralsight.com/course-player?clipId=7337f811-2fa2-49e0-a532-59063bafaf3e&startTime=2) [the gang of four are as follows, the Singleton pattern, the](https://app.pluralsight.com/course-player?clipId=7337f811-2fa2-49e0-a532-59063bafaf3e&startTime=5.32) [Builder pattern, a Prototype pattern,](https://app.pluralsight.com/course-player?clipId=7337f811-2fa2-49e0-a532-59063bafaf3e&startTime=10.61) [the Factory, and then finally, the AbstractFactory.](https://app.pluralsight.com/course-player?clipId=7337f811-2fa2-49e0-a532-59063bafaf3e&startTime=14.35) [We will implement all of these patterns and compare and](https://app.pluralsight.com/course-player?clipId=7337f811-2fa2-49e0-a532-59063bafaf3e&startTime=18.56) [contrast them to one another and described when the best case to use each individual one is.](https://app.pluralsight.com/course-player?clipId=7337f811-2fa2-49e0-a532-59063bafaf3e&startTime=22.16)

### [How Do We Learn Them?](https://app.pluralsight.com/course-player?clipId=46883b40-85ec-4805-b1d6-5f32c7a34822)

[Each module on this course is going to be structured to where we](https://app.pluralsight.com/course-player?clipId=46883b40-85ec-4805-b1d6-5f32c7a34822&startTime=2.24) [first give an overview of what the pattern is,](https://app.pluralsight.com/course-player?clipId=46883b40-85ec-4805-b1d6-5f32c7a34822&startTime=5.58) [then the concepts when choosing to use this pattern,](https://app.pluralsight.com/course-player?clipId=46883b40-85ec-4805-b1d6-5f32c7a34822&startTime=9.7) [what you need to consider is part of the design, we will cover a live](https://app.pluralsight.com/course-player?clipId=46883b40-85ec-4805-b1d6-5f32c7a34822&startTime=13.21) [example of how it's used in the Java API, then we're going to go](https://app.pluralsight.com/course-player?clipId=46883b40-85ec-4805-b1d6-5f32c7a34822&startTime=17.74) [through a demo where you end up coding your own design pattern. We'll](https://app.pluralsight.com/course-player?clipId=46883b40-85ec-4805-b1d6-5f32c7a34822&startTime=22.42) [discuss the pitfalls of this pattern,](https://app.pluralsight.com/course-player?clipId=46883b40-85ec-4805-b1d6-5f32c7a34822&startTime=26.62) [contrast it with another pattern, and then give a summary, an overview, of what we covered and how this pattern applies in day‑to‑day use.](https://app.pluralsight.com/course-player?clipId=46883b40-85ec-4805-b1d6-5f32c7a34822&startTime=29.59)

### [Prerequisites](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc)

[The prerequisites for this course are actually quite simple.](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc&startTime=1.64) [I am using Java 7, but Java 7 or 8 will work just fine.](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc&startTime=5.84) [And then I'm going to be using Spring STS, which is just a flavor of Eclipse.](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc&startTime=10.9) [Honestly, though, this code will work with any IDE out there.](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc&startTime=15.74) [If you prefer IntelliJ over Eclipse or Spring STS over Eclipse,](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc&startTime=19.42) [any of them are going to run just fine.](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc&startTime=24.24) [You could even run the code in a simple text editor and compile it](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc&startTime=26.43) [on the command line as we don't have a very complex package](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc&startTime=31.05) [structure and it should work just fine.](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc&startTime=35.129999999999995) [I like Spring STS because it seems to work and be a little bit more](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc&startTime=37.24) [stable than Eclipse, but it's really dealer's choice. You can choose to use whatever you want.](https://app.pluralsight.com/course-player?clipId=43ada65c-bca8-4cbc-8ee8-897fbcde31cc&startTime=40.48)

### [Next](https://app.pluralsight.com/course-player?clipId=6473ab6c-e1dc-4ab1-b859-2e8138d95dbb)

[So that covers the introduction and prerequisites for this course. Let's go ahead and get started by now looking at the singleton pattern.](https://app.pluralsight.com/course-player?clipId=6473ab6c-e1dc-4ab1-b859-2e8138d95dbb&startTime=1.19)

## [Singleton Pattern](https://app.pluralsight.com/course-player?clipId=4b9930d3-8e43-4a44-92c7-592b2a4e6357)

### [Introduction](https://app.pluralsight.com/course-player?clipId=4b9930d3-8e43-4a44-92c7-592b2a4e6357)

[Hi. This is Bryan Hansen, and in this module,](https://app.pluralsight.com/course-player?clipId=4b9930d3-8e43-4a44-92c7-592b2a4e6357&startTime=2.96) [we are going to look at the singleton design pattern.](https://app.pluralsight.com/course-player?clipId=4b9930d3-8e43-4a44-92c7-592b2a4e6357&startTime=5.48) [The singleton pattern is one of,](https://app.pluralsight.com/course-player?clipId=4b9930d3-8e43-4a44-92c7-592b2a4e6357&startTime=8.6) [if not the most, heavily used design patterns because of its simplicity to implement and the type of problem that it solves.](https://app.pluralsight.com/course-player?clipId=4b9930d3-8e43-4a44-92c7-592b2a4e6357&startTime=10.43)

### [Concepts](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5)

[The concepts when choosing a singleton are that it guarantees](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=0.775) [only one instance is going to be created.](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=4.66388888888889) [It also guarantees the control of a resource.](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=7.775) [Since this is a creational design pattern,](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=10.59852941176471) [the instantiation of it is all controlled through](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=13.069117647058832) [the implementation of the pattern.](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=15.95681818181818) [Although it doesn't have to be, it is usually lazily loaded.](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=17.775) [This ties in nicely with it being a creational pattern.](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=21.965476190476174) [Examples of this in the Java API or commonly used](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=25.775) [frameworks are the runtime environment, logger,](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=30.15) [but depending on the implementation this could be factory instead of singleton,](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=32.775) [and we will discuss this in more detail later, Spring Beans,](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=36.10833333333334) [if you've used the Spring framework at all,](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=39.61710526315791) [you will quickly learn that all Spring Beans are by default singletons.](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=42.56447368421057) [And a fourth example are graphics managers.](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=47.54423076923076) [Typically when you're using a Graphics API of any kind,](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=51.03815789473684) [you're going to get an instance of your graphical environment, and we only want one of those instances at a time.](https://app.pluralsight.com/course-player?clipId=dc419ef2-1f65-43f8-a11b-7ff1dff4bdd5&startTime=53.40657894736842)

### [Design Considerations](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d)

[The singleton is responsible for creating itself and managing its lifecycle.](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=1.14) [It is static in nature,](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=7.14) [although it is not implemented using a static class typically.](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=8.806666666666667) [The reason for not using a static class is that it needs to be thread safe,](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=12.14) [and static doesn't necessarily guarantee this for us.](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=16.940000000000012) [There is a private instance of a singleton,](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=19.85428571428571) [hence the minus sign or hyphen in the UML.](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=23.282857142857125) [There is also a private constructor that is marked the same way.](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=27.14) [This is because we want the singleton itself to call](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=32.14) [the constructor and nobody else.](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=36.139999999999986) [There are no parameters,](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=38.14) [and if you require parameters, that is typically a factory pattern and violates the rules of a singleton.](https://app.pluralsight.com/course-player?clipId=87765f9b-443a-4e4d-990f-7bfd8c74df0d&startTime=39.775555555555556)

### [Example: Runtime](https://app.pluralsight.com/course-player?clipId=d993d8f0-e707-4b34-b722-c82bd3c9d41f)

[This is a code snippet of an everyday example using a singleton.](https://app.pluralsight.com/course-player?clipId=d993d8f0-e707-4b34-b722-c82bd3c9d41f&startTime=2.107) [We get an instance of the runtime environment and then print out the object](https://app.pluralsight.com/course-player?clipId=d993d8f0-e707-4b34-b722-c82bd3c9d41f&startTime=6.107) [address for it just so we can visually see what it is.](https://app.pluralsight.com/course-player?clipId=d993d8f0-e707-4b34-b722-c82bd3c9d41f&startTime=9.440333333333328) [We then get another instance of it and print out the](https://app.pluralsight.com/course-player?clipId=d993d8f0-e707-4b34-b722-c82bd3c9d41f&startTime=12.559954545454545) [address of that object as well.](https://app.pluralsight.com/course-player?clipId=d993d8f0-e707-4b34-b722-c82bd3c9d41f&startTime=15.756454545454545) [We can see that both objects are using the same object address](https://app.pluralsight.com/course-player?clipId=d993d8f0-e707-4b34-b722-c82bd3c9d41f&startTime=17.5) [by using the equals operator to compare them, and it will print out that they are the same object.](https://app.pluralsight.com/course-player?clipId=d993d8f0-e707-4b34-b722-c82bd3c9d41f&startTime=22.299999999999983)

### [Demo: Runtime](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a)

[In this example,](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=1.969) [you can see that we're going to go ahead and get an](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=2.7190000000000003) [instance of the runtime environment,](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=5.719) [and then I went ahead and called garbage collect on here just so you](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=6.969) [could see it was the real runtime environment.](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=10.150818181818178) [Then we print out the object address using the System.out.println.](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=11.969) [Let me go ahead and grab another instance of it,](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=17.969) [print that address out, and you can see that they're the same object address.](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=19.92552173913042) [Then we'll use the equals operator just to again verify that they are](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=22.969) [the same object and print out the System.out.println.](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=26.925521739130442) [So if we go ahead and run this,](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=29.969) [you can see down below here that the object addresses are in fact the same](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=32.40378260869565) [and that it prints out that they are the same instance.](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=36.969) [So this is guaranteeing that we are in fact a singleton. This Runtime.getRuntime method is a singleton.](https://app.pluralsight.com/course-player?clipId=d522d5d2-a1b4-477a-b934-4d9f20a6799a&startTime=39.96900000000001)

### [Exercise - Create Singleton](https://app.pluralsight.com/course-player?clipId=505e4aaa-d7c4-4cd9-ba13-a3dab01ea9b3)

[So let's go ahead and run an exercise to create our own singleton.](https://app.pluralsight.com/course-player?clipId=505e4aaa-d7c4-4cd9-ba13-a3dab01ea9b3&startTime=0.139) [In this demo, we're going to create a singleton,](https://app.pluralsight.com/course-player?clipId=505e4aaa-d7c4-4cd9-ba13-a3dab01ea9b3&startTime=6.139) [and then we're going to demonstrate that only one](https://app.pluralsight.com/course-player?clipId=505e4aaa-d7c4-4cd9-ba13-a3dab01ea9b3&startTime=8.389) [instance of our object is created.](https://app.pluralsight.com/course-player?clipId=505e4aaa-d7c4-4cd9-ba13-a3dab01ea9b3&startTime=10.639) [We're going to then go through and convert it to being lazily loaded and finally make it a thread‑safe singleton.](https://app.pluralsight.com/course-player?clipId=505e4aaa-d7c4-4cd9-ba13-a3dab01ea9b3&startTime=12.538999999999998)

### [Demo: Static Singleton](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e)

[For our singleton exercise, we're going to go ahead and create a](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=1.3) [singleton to manage access to who can create connections to a database](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=4.11) [and where they can create them at. Let's right‑click on our package,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=8.82) [com.pluralsight.singleton, and say New, Class. I'm going to name this](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=13.3) [class DbSingleton, and click Finish.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=19.72) [And since we're going to build upon this example,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=26.44) [I want to start off by not making this lazily loaded or thread safe.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=29.3) [So we'll create an instance of our class by saying private](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=33.55) [static DbSingleton instance = new DbSingleton.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=37.2) [And this will create an instance that our application can hold on to,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=49.44) [and we'll manage it so that it only returns one instance of our application.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=54.41) [To do so, we're going to create a private constructor so that people](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=59.68) [can't create new instances of this class on their own,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=63.81) [and we'll just say private DbSingleton,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=68.14) [and this will manage how that gets created so that](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=75.04) [people can't new up those new instances.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=78.68) [Lastly,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=82.4) [to flush out our singleton and control how people will grab that instance back,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=82.89) [we're going to do the customary method with a singleton of saying](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=88.08) [public static DbSingleton, so it's going to return an instance of](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=91.56) [DbSingleton with a getInstance method.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=96.86) [Now this is just a standard naming convention, it](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=100.34) [doesn't have to be called this,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=102.44) [but it's customary with a singleton to have a getInstance method,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=103.68) [and this will return that instance that we created up above.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=108.84) [So there is everything we need to create our singleton.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=114.84) [We have our static private instance that will hold on to, we have a private](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=118.58) [constructor so that people can't use the keyword new to create their own](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=124.45) [instance of it, and then we'll return that instance to the calling class](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=127.95) [when they call the getInstance method.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=133.24) [And again,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=136.14) [it's not required that it be called getInstance, it's just a standard](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=137.01) [convention when implementing a singleton. To test this,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=140.85) [let's go ahead and create a class called](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=145.57) [DbSingletonDemo that will execute this code.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=149.18) [So just type in DbSingletonDemo here,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=153.29) [and I'm going to check the public static void main create and click](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=157.44) [Finish. And inside of here I'm going to replace this TODO with a](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=162.94) [DbSingleton instance, and remember that was a static method, so we'll](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=168.09) [call DbSingleton.getInstance to return an instance of our singleton for](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=175.22) [us. And to verify that this is running, I'm just going to do a](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=183.3) [System.out.println and return the object address that is created from](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=186.43) [that instance.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=192.66) [And I can run this by right‑clicking on our main method](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=194.64) [and say Run As, Java Application.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=197.98) [When this runs, you'll see that it prints out the object address in our console.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=201.01) [We have for this time that I ran it a 56e88e24.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=205.2) [Each time I run,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=210.39) [it will return a different instance for me. But](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=211.23) [there's a couple of things to note here.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=213.91) [Okay, we have a singleton.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=215.94) [How do we verify that?](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=217.04) [Well,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=218.44) [if we were trying to create a new instance of this](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=219.52) [since we did a private constructor, it won't allow us.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=221.96) [So if I type in here DbSingleton, and we'll say testConst = new DbSingleton,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=225.2) [it's not going to allow me to do this.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=236.14) [See, in fact, it tries to suggest that I put the demo in here.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=237.95) [If I take and close that off to the constructor, it tells me it won't](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=240.87) [allow me to do that. So it won't work that way, but let's test that we're](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=246.62) [only getting one instance of that object back.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=250.63) [I can do that by recreating that line of the singleton again,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=253.54) [saying Singleton anotherInstance = DbSingleton.getInstance.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=258.94) [Now when I run this and I print out that object address again,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=268.94) [it should print the same object address for me.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=273.24) [So I'm going to copy and paste that down there and](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=277.29) [replace that with another instance.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=280.02) [Save that, and right‑click and Run As again, Java Application,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=283.19) [and you'll see that the object addresses are the same down there.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=289.47) [So, it's not newing up another instance,](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=292.81) [it's returning that same object regardless of how many times I](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=295.88) [asked for it through that .getInstance method.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=298.71) [So we've now created our first singleton.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=301.7) [Let's move on to making this lazily loaded, and then we're going to follow that up with being thread safe.](https://app.pluralsight.com/course-player?clipId=8781a834-7314-4bc1-831b-cb8350c93e6e&startTime=305.24)

### [Demo: Lazy Loading](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c)

[To convert our singleton instance from being an eagerly loaded instance to](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=1.34) [being a lazy loaded instance is actually quite simple.](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=6.45) [Currently,](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=9.94) [it's eagerly loaded because on line 5 we're creating a new instance whether](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=10.59) [or not we use this. So all we need to do is grab this code, I'm going to](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=15.89) [copy it and change this to null, and then inside of our getInstance method,](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=22.69) [I'm going to do a simple null check and say if(instance) == null, then](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=28.849999999999998) [create a new instance of it.](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=40.38) [It's a very small change,](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=42.24) [and I will tell you right now that this isn't thread safe.](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=43.96) [We're going to talk about that in the next demo.](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=46.66) [But just changing this to being lazily loaded can be a substantial](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=48.78) [performance improvement for your application.](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=53.92) [I'm going to save this, switch back to our demo, and run this, and](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=56.75) [you'll see that it runs and executes exactly how we thought it would,](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=62.99) [and returns back our object addresses being the same across our](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=66.19) [original instance and our other instance.](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=71.76) [The difference here is just simply that we check to see that it's null and](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=74.9) [spool up that new instance. I've seen in large applications where we have a](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=79.65) [very slow startup because we're eagerly fetching all of those instances at](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=85.07) [the start up of our application.](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=89.96) [This will help it to where your app comes up quicker,](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=91.95) [and it's not such a memory hog when you're starting to spool up the application,](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=94.84) [it only uses what it needs.](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=100.05) [So, this can be a substantial improvement to your code](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=101.56) [and the performance of your code,](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=104.52) [which seems counterproductive where you're lazily loading it. Let's now look at making this thread safe.](https://app.pluralsight.com/course-player?clipId=12159418-4f21-4d9c-8bf3-51710d0eb60c&startTime=106.13)

### [Demo: Threadsafe](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72)

[At the time of the original recording of this course,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=1.04) [the keyword volatile was actually new of that version of Java,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=4.27) [so we didn't record it originally with that concept in mind.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=9.05) [So one of the things we'll do to make this thread safe is we'll go up](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=13.89) [to line 5 here and say private static volatile, and this will help us](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=18.45) [ensure that that instance will remain a singleton through any of the](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=25.51) [changes inside of the JVM.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=30.38) [The other thing we're going to do is to ensure that](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=33.24) [nobody uses reflection on our code.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=36.65) [So we're going to go to this private singleton that we've made and say if](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=39.77) [instance does not equal null, then we are going to throw a new](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=45.86) [RuntimeException, and inside of here we'll just say Use getInstance](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=55.77) [method to create. And we'll close this. Now, that will create our](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=68.09) [instance to where it's volatile and can't be reinstantiated through](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=77.47) [different things going on in the JVM.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=81.78) [This will also protect us from having a Reflection class go](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=83.62) [ahead and create an instance of this.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=87.86) [Now the next thing we're going to do is we're going to](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=89.94) [come down to our getInstance method.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=91.72) [Now, we could do this a couple of ways.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=95.14) [We're going to implement a double‑checked locking](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=97.51) [mechanism and a synchronized check.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=101.31) [Some people would originally on line 13 just make this whole method](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=104.94) [synchronized, and the problem with doing that is it's a performance hit.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=109.36) [If we make this whole method synchronized every](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=113.74) [time we ask for an instance of it,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=116.51) [we're going to actually synchronize that class and slow it completely down.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=119.04) [Rather than doing that,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=124.94) [let's go ahead and look and see if our instance is equal to null, and if it is,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=126.39) [we're going to synchronize inside of here, so we'll say synchronized,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=132.06) [and we're going to do it on the singleton,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=140.54) [the DbSingleton class,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=143.26) [and then we're actually going to check for null one more time.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=147.7) [The idea behind this is that it's only going to actually](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=155.42) [happen if we're creating this one time, so it's a little bit of extra code,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=158.21) [but it should only run if we're actually creating this for the very first time.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=162.15) [So we'll do the exact same code again and we'll say](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=166.95) [if instance is equal to null,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=169.94) [then create our instance of our DbSingleton. From there,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=178.54) [we'll return back out our instance.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=186.44) [So, we added our volatile to our instance, we protected this](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=189.55) [from being instantiated through reflection.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=195.07) [Rather than synchronize on the whole method,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=197.84) [we checked to see if the instance is equal to null, then we synchronized on it.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=199.83) [And the reason for doing this is it may be null, but if](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=203.71) [two threads are trying to go at it, once we've](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=207.72) [synchronized and checked for null again,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=210.38) [if another class has a lock on that,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=213.34) [it will then block our code and create the instance and return that](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=216.07) [synchronized lock to where our code would now go back in and say,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=220.96) [if this instance is null it would go, oh,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=223.6) [no, I'm already created and returned back out of this. You can see how](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=226.37) [that double checked with volatile instance inside of there is handling](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=229.14499999999998) [that functionality inside of our class.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=232.89) [Now, let's double check this by going back to our demo and running it again,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=234.81) [and you'll see that it works correctly.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=239.04) [You're not going to notice a real performance hit inside of](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=241.09) [our application now because we're only doing this with two](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=244.01) [instances across our main method.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=247.77) [But, it does add a little bit of overhead with that](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=250.04) [synchronization inside of there.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=252.36) [But at the safety of our application not having multiple threads accessing this](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=255.49) [and our singleton really not truthfully being a singleton,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=259.85) [we're now lazily loaded, thread safe,](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=263.64) [and we're using the latest version of volatile inside the JVM to make this a fully thread safe instance of a singleton.](https://app.pluralsight.com/course-player?clipId=ddd36dfc-35d9-4770-b904-9423fc2ace72&startTime=267.34)

### [Demo: Add Database](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f)

[To make this an even more realistic example,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=1.84) [I've gone ahead and added the Derby database JARs into our](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=4.74) [application to show you what this would look like to tie to a real](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=7.9) [database. Derby, if you're not familiar with it,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=11.52) [is a very lightweight database that you can use for in‑memory](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=15.13) [applications, and a lot of times it's utilized for testing,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=18.23) [and I've even used it in a few production settings as well.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=21.8) [So the first thing we're going to do is register a driver,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=25.44) [and rather than have you watch me type all of this in, I'm](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=28.1) [going to just paste in these snippets,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=31.34) [and I've replaced our constructor with this code that goes through and](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=34.34) [creates an instance of the driver manager, and this will go through and](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=44.59) [register our database driver using that driver manager and check to see](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=50) [if our connection is not equal to null for a reflection like we had done](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=55.1) [in the previous demo.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=59.21) [So, you'll notice this connection statement is erring out.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=60.64) [We're going to take and create an instance of that connection in here.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=63.67) [Let me grab that snippet,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=69.25) [and we'll copy this and paste it in there. We're going to do the](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=74.08) [same thing as we had done before by creating a volatile instance](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=78.68) [of this, and let me import Connection.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=81.69) [And then the next thing that we need to do is create our](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=88.24) [getConnection method inside of here.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=91.47) [And this looks very similar to what we did with our getInstance, but there is](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=93.95) [one key difference here. Our getConnection is not static,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=98.33) [and this is by design.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=104.42) [And we could set this up one of a couple of ways, but I chose](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=105.71) [to do it this way because it's very similar to how we would do](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=108.57) [this in production applications.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=111.56) [I've gone ahead and made this to where we have to have an instance of our](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=114.04) [singleton and then use that instance to get our connection.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=118.99) [And so we're going to utilize the Singleton to call an](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=125.37) [instance of this class first and then get the connection](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=129.6) [and return that. To test this,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=134.08) [let's switch over to our DbSingleton demo, and inside of here,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=137.28) [we're going to clean this up a little bit.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=144.82) [We've got our code left over from the previous demo.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=147.14) [Let's go ahead and cut all of this out except for that DbSingleton.getInstance,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=149.81) [and we'll start by just grabbing a connection, and](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=157.54) [you'll see I've got some other code here.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=163.57) [We're actually going to run that in a second. And grab that connection,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=164.88) [and then we can go through and utilize that instance.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=169.29) [So, what we have here is us getting an instance of our](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=174.05) [singleton and then just grabbing that connection.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=179.09) [Then let's grab our statement, and I have a statement already built down here,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=183.84) [and we'll execute this code inside of our demo instance as well.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=190.14) [And this code goes through and creates an instance of a table using a](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=195.24) [prepared statement to do so and handles the exceptions.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=200.67) [I'm going to import all of those here, and we have a complete working example.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=206.02) [So let's run this how it is, and we're going to add](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=212.19) [some performance metrics to this.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=214.88) [Let's right‑click and say Run As, Java Application.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=217.37) [You'll see that it goes through and it says that we've created our table.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=221.84) [It just dumps out that System.out.println on line](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=225.21) [20 and prints out Table created.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=227.63) [That's great.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=230.74) [Let's do some more sophisticated things with this code,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=231.54) [though. So, I'm going to switch back over here to my notes,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=235.35) [and I've got some longs that we're using to grab the timeBefore and](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=238.65) [the timeAfter, and we're going to time our connection creation gap to](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=245.38) [see how long it takes us to create this, and then we'll call it again](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=252.08) [when we're done with this and see how quickly it will create that](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=255.52) [connection for us.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=258.58) [So, I grab that chunk of code and replace this connection here with that,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=259.91) [and you'll see that we're just grabbing the timeBefore and](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=266.91) [the timeAfter using a System.currentTimeMillis. I'm going](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=270.22) [to print that difference out.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=273.515) [But before we execute that,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=275.44) [let's do that same thing after. I'm going to go down here and grab another](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=277.38) [timeBefore and another timeAfter and print that out once again. And you'll](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=284.42) [see here that it will execute and require very little to no time to run](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=292.96) [that instance.getConnection again.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=300.25) [So let's save this, and now when we run this,](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=302.84) [you'll notice that the first time it runs it took 503 ms to create it, it](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=310.24) [says that our table is created, and the next time that it runs it's been](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=318.09) [optimized to where it took 0 ms because it is returning us that](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=322.04) [connection that we already have created.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=327.73) [So we have our singleton of our instance, and then our](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=329.54) [connections inside of that singleton that's returning that in 0](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=332.62) [ms, so it's optimized that code to where we only have one](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=337.1) [connection that we're sharing across this.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=340.67) [The creation of that connection has been optimized to where it returns in 0 ms.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=342.99) [It's just returning us back the connection we have. You can see some real performance benefits for us doing our singleton this way.](https://app.pluralsight.com/course-player?clipId=26aa7a97-2943-439e-8b69-d4ac0542fe5f&startTime=349.02)

### [Pitfalls](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289)

[Some of the pitfalls of a singleton are that they are often overused.](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=0.44) [Once people discover the power and simplicity of this pattern,](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=5.06) [they have a tendency to make everything a singleton when](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=8.34) [it doesn't necessarily need to be.](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=10.75) [Although there aren't generally performance problems with singletons,](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=13.14) [if you make everything a singleton, it will slow your application down.](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=16.48) [Since singletons don't expose an interface and have private](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=21.03) [constructors, as well as private member variables,](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=24.38) [they can often be difficult to unit test.](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=27.78) [If you aren't careful when implementing it, they're not thread safe.](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=30.69) [Oftentimes,](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=34.66) [people start off with a singleton that's static, like we demonstrated,](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=35.63) [and it ends up morphing into something else and can](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=39.12) [oftentimes be confused for a factory.](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=42.79) [They start making the getInstance method take parameters.](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=45.48) [A rule of thumb is that as soon as it needs an argument in](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=49.45) [that method, it is not a singleton anymore,](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=52.415000000000006) [but rather a factory.](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=54.58) [Although not a pitfall, the java.util.Calendar is not a singleton, it is](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=56.51) [actually more of a prototype pattern because you are getting a new unique](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=62.82) [instance every time you call the getInstance method.](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=67.49) [People often confuse this, though,](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=70.84) [because it uses that identifier of getInstance, which is typically associated with a singleton.](https://app.pluralsight.com/course-player?clipId=9b0b29a6-d05d-4ffd-999d-772d3f77a289&startTime=72.64)

### [Contrast to Other Patterns](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802)

[To contrast the singleton with another design pattern that it's](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=2.34) [commonly confused with, the factory, let's go through the two side](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=7.02) [by side so you can see the differences.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=10.96) [A singleton will return the same instance every time.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=13.44) [There is a one constructor method with no arguments, and](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=18.71) [notice how I label that as constructor method.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=23.68) [The constructor is always private so you can't get access to it,](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=25.63) [so we have to access that through a construction method. And then](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=29.17) [there is typically no interface.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=32.77) [Since this is a private constructor and a private instance inside of it,](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=34.63) [we don't expose an interface to help us adapt to](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=39.64) [different types of objects return. A factory,](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=43.02) [on the other hand, returns various instances,](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=46.57) [and, as the name implies, it returns multiple objects of various types.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=49.99) [It also has multiple constructors because we're asking for those various types.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=56.94) [There's different construction methods for us to get those instances back.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=61.58) [It is usually interface driven.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=66.84) [It's the opposite.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=69.17) [We want to abstract out the back end and some of those things that we're](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=70.16) [returning so we usually do expose an interface with a factory,](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=73.93) [so it's a lot easier to unit testing and a lot easier to work with.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=77.49) [It also has the ability to adapt to environments more](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=81.54) [easily than the singleton does.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=86.49) [So, when you're looking at a singleton and it's not quite fitting,](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=88.37) [think about the factory and what the factory brings to the table, and](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=92.68) [some of these comparisons may help you in choosing a factory over a singleton, or a singleton over a factory.](https://app.pluralsight.com/course-player?clipId=2f19ea44-7b5a-4610-b039-5f447a54a802&startTime=96.24)

### [Summary](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6)

[To summarize the singleton pattern, we use this pattern when](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=1.04) [we want to guarantee that there's only one instance of an](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=4.68) [object inside of our application.](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=7.42) [It's very easy to implement,](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=9.81) [and it's even easy to make thread safe if you just spend a few](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=11.47) [minutes wrapping the construction of that object. It solves a very](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=14.9) [well defined problem where we only want to have one instance of that](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=19.24) [object inside of our application.](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=24.33) [But it can be abused quite easily.](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=26.94) [It's usually the most abused design pattern that we](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=29.61) [see out there in applications.](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=32.88) [Not everything needs to be a singleton, not everything needs](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=35.64) [to be guaranteed that there's one instance.](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=38.65) [So use it wisely and think about what you're needing to solve with this, and don't confuse it with a factory pattern as well.](https://app.pluralsight.com/course-player?clipId=db74d4d5-a38a-4d2e-89aa-40726e387bc6&startTime=40.22)

## [Builder Pattern](https://app.pluralsight.com/course-player?clipId=b6c097e1-e038-4377-aaf5-c36cbc169661)

### [Introduction](https://app.pluralsight.com/course-player?clipId=b6c097e1-e038-4377-aaf5-c36cbc169661)

[Hi. This is Bryan Hansen, and in this module,](https://app.pluralsight.com/course-player?clipId=b6c097e1-e038-4377-aaf5-c36cbc169661&startTime=2.12) [we're going to look at the builder pattern.](https://app.pluralsight.com/course-player?clipId=b6c097e1-e038-4377-aaf5-c36cbc169661&startTime=4.36) [The builder pattern is a pattern that people often](https://app.pluralsight.com/course-player?clipId=b6c097e1-e038-4377-aaf5-c36cbc169661&startTime=6.97) [use, but rarely create their own.](https://app.pluralsight.com/course-player?clipId=b6c097e1-e038-4377-aaf5-c36cbc169661&startTime=9.33) [It is a great pattern for handling the construction of objects that](https://app.pluralsight.com/course-player?clipId=b6c097e1-e038-4377-aaf5-c36cbc169661&startTime=12.04) [may contain a lot of parameters, and we want to make the object immutable once we're done constructing it.](https://app.pluralsight.com/course-player?clipId=b6c097e1-e038-4377-aaf5-c36cbc169661&startTime=15.13)

### [Concepts](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a)

[When considering a builder,](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=1.175) [you want to focus on whether or not the construction of an object is complex.](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=2.4381578947368423) [By complex,](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=7.175) [we are specifically talking about lots of arguments for](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=7.763235294117647) [a constructor, or lots of setters,](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=10.410294117647064) [and then guaranteeing a contract of how that object gets built.](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=12.175) [Another key concept that is often overlooked is that you can force](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=16.175) [immutability on an object once the construction is finished,](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=20.174999999999986) [which you can't necessarily do with just a bunch of setters.](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=23.246428571428574) [This just helps further enforce our contract.](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=27.175) [Examples of this in the Java API are the StringBuilder,](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=29.64558823529411) [probably one of the most common ones out there,](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=33.175) [which is honestly one of the best examples of a](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=35.87499999999997) [builder pattern; the DocumentBuilder,](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=38.87499999999994) [which I feel like illustrates how this pattern can be used on a](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=40.00833333333333) [complex object creation; and then a Locale.Builder,](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=43.619444444444454) [which is just another good example of this pattern and how it takes various parameters as part of its construction.](https://app.pluralsight.com/course-player?clipId=f1891b14-49ae-4e03-b39d-faf3dec2031a&startTime=46.06973684210527)

### [Design Considerations](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead)

[The builder pattern solves a very common problem in object oriented](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=2.14) [programming and that is determining what constructor to use.](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=5.79) [Oftentimes, people create multiple constructors,](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=9.74) [and it can become difficult to manage.](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=12.31) [The creation of multiple constructors with each parameter](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=14.77) [variation is called a telescoping constructor.](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=17.33) [The builder pattern helped us overcome this by handling that](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=21.24) [construction with an object and rather than by parameters. The builder](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=24.26) [is typically written with a static inner class.](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=29.38) [The reason for this is that it returns an instance of](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=33.24) [the object that it is building.](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=35.55) [It doesn't negate the need for constructors and rather](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=38.34) [works in unison with those to call the appropriate](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=42.21) [constructor based off of its state.](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=45.91) [The builder pattern can negate the need for the job of being anti‑pattern of](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=48.84) [exposing setters for every parameter that we could pass in.](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=53.46) [Since Java 1.5,](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=58.31) [we can take advantage of generics where it warrants to utilize a](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=60.17) [builder to expose various types of objects. But this is often not a necessary feature.](https://app.pluralsight.com/course-player?clipId=84fdc370-9a88-4f5c-9571-b80e2fb62ead&startTime=63.85)

### [Example: StringBuilder](https://app.pluralsight.com/course-player?clipId=2a4b4707-d5d7-4ed6-8056-867bb9f483a4)

[The StringBuilder class was introduced in Java 1.5.](https://app.pluralsight.com/course-player?clipId=2a4b4707-d5d7-4ed6-8056-867bb9f483a4&startTime=1.24) [It fixed a lot of wrong doings that people were using just](https://app.pluralsight.com/course-player?clipId=2a4b4707-d5d7-4ed6-8056-867bb9f483a4&startTime=5.24) [the string class for, or trying to do things better by](https://app.pluralsight.com/course-player?clipId=2a4b4707-d5d7-4ed6-8056-867bb9f483a4&startTime=8.24) [using the string buffer class.](https://app.pluralsight.com/course-player?clipId=2a4b4707-d5d7-4ed6-8056-867bb9f483a4&startTime=10.76) [The StringBuilder is a great example of a builder class,](https://app.pluralsight.com/course-player?clipId=2a4b4707-d5d7-4ed6-8056-867bb9f483a4&startTime=12.92) [although oftentimes they are implemented as an internal static](https://app.pluralsight.com/course-player?clipId=2a4b4707-d5d7-4ed6-8056-867bb9f483a4&startTime=15.91) [class, but it still shows the usefulness of this pattern by building it retroactively for the string object.](https://app.pluralsight.com/course-player?clipId=2a4b4707-d5d7-4ed6-8056-867bb9f483a4&startTime=18.98)

### [Demo: StringBuilder](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5)

[Here is a simple demo of the StringBuilder object.](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=null) [You can see that we're going ahead and getting a new](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=3.5055) [instance of the StringBuilder and then appending these](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=7.79) [individual strings to it as we go,](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=10.45666666666667) [and eventually we come down here and append an integer to it.](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=12.790000000000008) [There is just about any type of object that you want available](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=16.79) [through the corresponding append method to the builder,](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=20.79) [which we finally end up calling the builder.toString,](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=24.554705882352927) [which for this particular builder, since we are building a string,](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=28.79) [they just took advantage of the toString method to make that be our builder.](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=31.845555555555563) [There isn't a .build method, we call builder.toString like this right here.](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=36.70666666666665) [When we run this, you can see that it will go ahead and generate our string.](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=40.694761904761926) [This is an example of the builder pattern,](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=44.737368421052636) [and it has methods to append almost anything we want](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=47.263684210526336) [to a string and the number 42.](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=50.485652173913046) [The good thing about this is it is really performant and it gives us a](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=52.92043478260871) [nicer way to build strings rather than using the plus sign or the](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=58.19) [concat operator inside the string object.](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=63.38999999999998) [It is also a lot more performant than the StringBuffer object.](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=65.79) [The StringBuffer object does some locking much like the old](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=70.31941176470583) [vector object did compared to an array list,](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=74.18999999999998) [and this will result in faster performance for our application,](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=76.98999999999994) [as well as just simple ease of use, and it really is a great example of the builder over a buffer.](https://app.pluralsight.com/course-player?clipId=2abdd9c0-4a17-4710-8df5-0be7a81a7dd5&startTime=81.79)

### [Exercise - Create Builder](https://app.pluralsight.com/course-player?clipId=29100a22-dad4-4611-941f-5e427e921819)

[Now that we've seen an example of the string builder in action,](https://app.pluralsight.com/course-player?clipId=29100a22-dad4-4611-941f-5e427e921819&startTime=0.64) [let's go ahead and build our own implementation of the builder pattern.](https://app.pluralsight.com/course-player?clipId=29100a22-dad4-4611-941f-5e427e921819&startTime=3.83) [First we're gonna look at demonstrating exposed setters out of a bean.](https://app.pluralsight.com/course-player?clipId=29100a22-dad4-4611-941f-5e427e921819&startTime=8.14) [Then we'll show some of the drawbacks of telescoping](https://app.pluralsight.com/course-player?clipId=29100a22-dad4-4611-941f-5e427e921819&startTime=12.92) [constructors. Then we'll finally create our own builder and finish building out the rest of the example.](https://app.pluralsight.com/course-player?clipId=29100a22-dad4-4611-941f-5e427e921819&startTime=15.54)

### [Demo: JavaBean Setters](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95)

[For the first part of our builder example, we're just](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=1.84) [looking at a straight bean here, and this is to create a lunch order.](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=4.99) [This was actually a problem I ran into for a company that I was](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=8.73) [doing some work for while building out this class.](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=12.36) [They had orders for lunch for a simple application that they were](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=15.71) [doing and they wanted to be able to build out each individual](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=20.14) [order, and the first stab that they took at this was a simple](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=23.47) [lunch order bean like this,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=27.93) [and I've simplified this example down, but you can see we have bread,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=29.53) [some condiments,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=33.51) [whatever dressing we want on there, and then finally a](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=35.03) [meat that may or may not be there.](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=38.71) [The problem with this example is first,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=40.78) [we're using the default no‑args constructor, if we were to build this out,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=42.91) [you could see that we just have a public LunchOrderBean here.](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=46.79) [So we're using this default no‑arg constructor here, and](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=55.34) [then we have a getter and setter exposed for every](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=58.85) [individual property that we have there.](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=63.07) [The problem with this is for one, it's not immutable,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=65.86) [meaning that after we create it,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=71.14) [we can go through and change it because we have all these setters, and it's](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=73.26) [really unclear what the contract is of what they must have or must not have to](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=77.19) [signify what their lunch order is. Now to run this, I've got a little simple](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=82.79) [demo here, we just create an instance of the LunchOrderBean, and then just set](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=87.32) [each individual property that you want.](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=92.33) [We set the bread, set the condiments,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=93.86) [set the dressing, and then set whatever meat we want; and that works](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=96.84) [fine, and it will facilitate what we're trying to do.](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=100.51) [If we run this example,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=104.17) [we can see that it has our wheat, lettuce, mustard, and ham,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=106.54) [which is exactly what we set up here, but there are](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=110.42) [kind of some bigger problems with it.](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=114.58) [Like I said,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=116.21) [it's not immutable, it has no contract as to what actually](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=116.84) [signifies being a valid lunch order.](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=121.37) [We could go ahead and comment out all of these, and it would still](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=123.75) [run, and we don't have a real kind of idea of what our order should](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=129.41) [be. So, it'll work, it'll get the job done,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=136.69) [but there's quite a few problems with it. In the next step,](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=139.56) [we're going to go ahead and look at telescoping constructors to see if it won't solve this problem.](https://app.pluralsight.com/course-player?clipId=1e0c3b86-7b50-4325-9f27-d7c53c60cc95&startTime=143.14)

### [Demo: Telescoping](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea)

[Now this class is very similar to the LunchOrder bean,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=1.23) [except it's using telescoping constructors.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=4.72) [So if we look at the constructors in this class,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=7.64) [I've got a LunchOrderTele here, and if I set these two side by side,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=10.89) [you can see that they're very similar in nature to one](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=14.95) [another, except that we have a bean model,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=19.72) [which is a no arguments constructor with setters and getters versus a](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=24.8) [constructor model, which is building upon the constructor for each](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=28.36) [individual case that we want to work with.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=32.77) [Now, let's move this back over here and look at this](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=35.11) [in a little bit greater detail.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=38.39) [You can see that our constructors build upon one another,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=40.14) [so we have one that takes the bread,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=43.769999999999996) [one that takes the bread and condiments, one that takes bread,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=45.76) [condiments and dressing, then one that takes bread,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=48.78) [condiments,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=51.52) [dressing, and meat; and I've telescoped one way with this,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=52.05) [I've gone up, but you could go down.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=55.45) [So what I mean by that is we have this bread,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=58.04) [condiments, and dressing, which is calling this constructor,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=61.02) [which in turn calls this constructor, which in turn calls this constructor.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=64.24) [So they loop or recurse through each one.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=69.33) [The bad part of this is well, we could also go the opposite way,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=73.02) [which can be a little confusing.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=77.93) [I could call this and pass in null to the other ones,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=79.93) [or some default value, and work our way down all four of these constructors.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=82.9) [But what happens if I want to build a sandwich that I don't](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=87.56) [want bread, or I don't want condiments?](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=92.48) [I just want bread and meat.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=96.95) [Or maybe I'm on the Paleo diet or Atkins diet and I want to do](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=99.19) [condiments, I just want lettuce wrapped around meat and I don't want](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=104.62) [any dressing, I don't want anything else.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=107.87) [Well,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=109.82) [this example won't do this, so I have to add another constructor just](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=110.07) [for that. So I could do a constructor that is bread, or condiments and](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=113.94) [meat, or just condiments, or condiments,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=118.97) [meat, or straight meat,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=120.79) [which it doesn't lend itself very well to that. Now the bean](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=121.86) [demo does do that a little bit better,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=125.31) [but then we have getters and setters for everything](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=128.03) [and it's immutable after the case.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=130.49) [So this example is immutable, so I could have it one way or the other.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=131.7) [I can have an immutable example using the telescopic constructors,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=136.62) [or I can have one that will work for all the configurations that I have,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=140.84) [which is the bean model.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=145) [So you can see the drawback to both of them, and just to show you how this runs,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=146.62) [I've got a constructor here that takes wheat, lettuce, mustard,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=150.54) [and ham, if I go ahead and run this example, you'll see that it](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=153.73) [prints out what we're expecting here.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=157.34) [But I've left in these setters to show you what we would have to](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=159.64) [do to get the configurability we want from the bean example, and](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=162.75) [we can't do that if I uncomment this,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=166.9) [it's going to tell us that that's not available](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=168.59) [because that setter doesn't exist.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=170.68) [So this is one of the problems with the telescoping](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=173.79) [constructor method versus the bean.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=175.84) [Now the next example is going to be using the builder,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=179.26) [which is the best of both worlds in this case,](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=181.51) [and you can see how it's a little bit more configurable doing these types of things.](https://app.pluralsight.com/course-player?clipId=b135b6f5-87d5-4c37-b8b7-0651059968ea&startTime=184.05)

### [Demo: Builder](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6)

[I've got a basic class here for our LunchOrder that is very similar to](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=2.34) [what we had for our LunchOrderBean and our LunchOrderTelescoping](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=7.08) [constructor model, except there's a few little changes in here. To](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=11.55) [start with, I've got this String bread, condiments,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=15.45) [dressing, and meat.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=18.71) [And then we've got the skeleton of a public static](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=20.74) [class called Builder inside of here.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=25.32) [Now inside of this,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=28.82) [we're going to actually implement this builder and show you how this works.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=29.54) [It's a little different, so I didn't want to use any copy and paste code in here.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=33.8) [First we want to go through and this seems a little redundant,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=37.9) [but we're going to create the same fields that we have inside of](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=41.33) [our LunchOrder class. So we're going to do private String bread.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=45.09) [We're going to do private String condiments. And then we're going](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=47.94) [to do dressing and meat.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=53.815) [The purpose for this is that the builder is its own container until](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=57.78) [we tell it to finally make what our LunchOrder is.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=64.65) [So now that we have our bread, condiments,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=68.68) [dressing, and meat in here,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=72.27) [we start off first with a basic no args constructor because](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=73.27) [we don't have any qualifications about what we might want to](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=79.94) [have inside of our sandwich.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=83.16) [Now the beauty of this is that we can mandate that this](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=85.25) [constructor take arguments for things that are required.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=89.3) [So if we required everybody to have a bread and a meat associated, we could make](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=92.29) [this constructor force that. But we don't care about that.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=97.39) [So from here,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=102.41) [let's go ahead and we're going to create a bunch of methods that are](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=104.14) [going to look like constructors, but they're not.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=107.64) [And this is the key to the builder pattern.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=109.98) [Let's go ahead and make this full‑screen while we're doing the](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=112.47) [rest of this example. So we're going to do public Builder. It's](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=114.94) [going to return an instance of itself.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=118.5) [And this one's going to be for bread. I'm going to say String bread.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=121.04) [Now the beauty of this is that we could also use](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=126.29) [typesafe enums for this as well.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=129.24) [So I'll say this.bread is equal to bread.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=131.74) [Now the catch here is that we return an instance of this. And by this,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=136.38) [we mean the builder object, this object that we're](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=141.49) [creating. Now that seems a little weird.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=144.59) [It's so you can do a little approach or technique here that](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=148.25) [I'm going to demonstrate in a minute about using the return](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=152.51) [to build out your object.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=155.8) [So let's finish this out for the others.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=157.7) [We're going to do the same for our condiments.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=159.74) [And rather than type everything out,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=165.77) [I'll just copy this and paste it in so you can see it.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=167.44) [So we're going to paste in our condiments, dressing, and](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=175.95) [meat. So now all of these are the same.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=178.79) [We have an instance of a method for bread that returns an](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=181.07) [instance of Builder, an instance of itself.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=185.16) [And then we're going to do one last key part,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=187.52) [and that's implement the actual builder.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=190.12) [Now to do that,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=192.78) [we're going to do a method called, I'm going to create a method](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=194.64) [called public LunchOrder that returns, excuse me public LunchOrder,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=199.09) [and it has a method in here called Build.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=205.99) [Now this is a little different than the others because this is](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=210.28) [going to return a new LunchOrder using this.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=213.48) [Okay,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=224.84) [so this has a‑‑‑ Inside of our LunchOrder object, we have](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=225.75) [a constructor that takes a builder, and then that copies](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=229.65) [over the bread, condiments, dressing, or meat that we have created.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=233.83) [And we can enforce the contract inside of our builder. But also](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=238.52) [notice that there's only getters down here.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=242.34) [I've minimized them so it doesn't take up as much space.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=244.53) [There's only getters,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=247.04) [so there's no setters. Now the beauty of this is we have the flexibility of the](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=248.01) [bean approach with the contract nature of the constructors.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=252.5) [So if we look at our BuildLunchOrder demo, our](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=259.06) [BuilderLunchOrderDemo, you can see that what we did is created](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=263.61) [our new instance using the static inner class here.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=266.39) [So we say LunchOrder. Builder returns an instance of](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=269.21) [LunchOrder.Builder, and we have our builder object.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=272.04) [Now from here, we say builder.bread Wheat .condiments.dressing.meat.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=275.32) [That's why we return an instance of ourselves so that we can just tack](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=280.39) [on these methods as we go and build out our object.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=283.89) [Now why I like this and why this solved the particular problem that I](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=287.1) [was looking at here with this LunchOrder problem is that I was parsing](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=290.58) [these values out of a CSV file, out of a comma‑separated value file as I](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=295.39) [was going. So I, instead of trying to gather all those at once and all](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=300.67) [the different conditions, I could grab it.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=304.49) [And if that value existed, I could then append it on.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=306.34) [And if it didn't, I'd go to the next one.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=310.03) [So if somebody had bread, no condiments, and mayo and turkey, great.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=311.54) [If somebody just had bread and meat, great. I could keep going through.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=316.6) [I don't have to have this in here. So I can cut one of these out,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=320.71) [and it will work just fine.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=324.83) [So if we run this,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=327.74) [you can then see that the next thing we do after we have all of](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=329.44) [our builder condiments and everything else added on there, we](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=332.64) [call build. If you wanted to, you could name this method](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=336.73) [MakeMeASandwich. We call build, and it returns an instance of our LunchOrder.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=339.95) [But we also have that immutability that nobody can edit it.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=345.06) [So now if we run this, we can see there is our wheat, lettuce, mayo, and turkey.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=348.87) [But we can also chop out one of these and it'll work just fine, and we](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=357.24) [don't have to have an edge case constructor for it.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=361.39) [We don't have to worry about the bean method, immutability, or whether](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=365.02) [or not. Now notice this value is null. We could also set a default type](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=369.01) [in there, the empty string or whatever we want to do. So we get a lot](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=372.86) [more flexibility about the creation of our object and how to handle](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=376.46) [these types of things.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=379.94) [And like I said just to reiterate,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=381.14) [if you did want to force them to have certain things in there,](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=382.67) [we could make it to where they have to use one instance of the builder](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=386.03) [with whatever those values are. So we can get that benefit of the](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=390.12) [telescoping constructors without the limitations or the problems of maintaining those individual constructors.](https://app.pluralsight.com/course-player?clipId=64e54a95-fda4-4a7a-8b18-4f2fb3b48be6&startTime=393.23)

### [Pitfalls](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57)

[The builder pattern really doesn't have a lot of negative things about](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=1.09) [it, so there really aren't that big of pitfalls,](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=3.99) [but maybe just some things to consider when choosing to implement it.](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=6.96) [Objects created with the builder are typically designed to be immutable.](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=10.94) [The pattern itself is also typically implemented](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=14.54) [with a static inner class, again,](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=17.22) [not a big issue, and as we demonstrated with the StringBuilder API,](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=19.82) [there are ways around that.](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=23.33) [Unlike the prototype pattern,](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=25.24) [it isn't something that is usually refactored in after the fact, it does](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=27.15) [add a little bit more complexity to our implementing class over what](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=32.51) [could have been done with just a constructor,](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=35.98) [but without some of the nice features of the builder patterns.](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=38.1) [Adding to the complexity is that people are typically not used to an object returning itself for each subsequent call.](https://app.pluralsight.com/course-player?clipId=1812af28-7820-4596-89e9-6093fb215f57&startTime=41.48)

### [Contrast to Other Patterns](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020)

[To contrast the builder pattern, let's compare it with the prototype pattern.](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=2.64) [The builder pattern is designed to handle complex constructors.](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=8.31) [There's typically no need for an interface,](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=13.84) [but you can implement the builder with one if you want. It](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=16.12) [can be implemented with a separate class,](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=20.27) [even though it is typically implemented in the class that it is building.](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=22.59) [And since it is implemented in a separate class,](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=26.29) [it can easily integrate with legacy code without needing change it.](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=29.63) [The prototype, on the other hand,](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=33.99) [is implemented around a clone method and trying to avoid the](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=35.97) [need to call a complex or costly constructor.](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=39.99) [Since the clone method is focused around member variables and constructors,](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=45.4) [it is implemented inside the class that it is trying to clone.](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=50.82) [This can make it difficult to implement in legacy code.](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=54.5) [Both of these patterns, though, are focused on](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=57.69) [complex constructors within one class.](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=60.74) [Their approach, though, is quite a bit different to solving that problem.](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=63.74) [So the builder tries to work with complex constructors where the prototype tries to avoid having to call them again.](https://app.pluralsight.com/course-player?clipId=fb6b363c-c49c-41f5-9e23-d287a9e6f020&startTime=67.84)

### [Summary](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9)

[Let's just recap what we covered with the builder pattern.](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=2.44) [It is a creative way to deal with complexity surrounding](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=6.99) [constructors in the creation of objects.](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=11.59) [It is fairly easy to implement,](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=15.24) [I would say almost as easy as the singleton to implement if done correctly.](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=17.26) [There are very few drawbacks to it, in fact,](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=22.6) [you kind of have to go out of your way to find some](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=25.21) [drawbacks with the builder pattern.](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=27.69) [And lastly, you can refactor it in with a separate class.](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=30.34) [Although it's typically implemented within itself,](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=33.86) [there's no reason that static inner class has to be a static inner class.](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=36.62) [It can be an external class that we go ahead and create what we](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=40.93) [want with our builder and then just call the constructor or](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=45.19) [whatever we want regarding setters or whatnot on the class that we are the builder for.](https://app.pluralsight.com/course-player?clipId=cabb82ec-a51a-487f-893b-3dceab3055e9&startTime=48.4)

## [Prototype Pattern](https://app.pluralsight.com/course-player?clipId=99469eb0-d8f0-494a-9cdc-b7cf21799f3f)

### [Introduction](https://app.pluralsight.com/course-player?clipId=99469eb0-d8f0-494a-9cdc-b7cf21799f3f)

[Hi. This is Bryan Hansen, and in this module,](https://app.pluralsight.com/course-player?clipId=99469eb0-d8f0-494a-9cdc-b7cf21799f3f&startTime=2.25) [we are going to look at the prototype design pattern.](https://app.pluralsight.com/course-player?clipId=99469eb0-d8f0-494a-9cdc-b7cf21799f3f&startTime=4.61) [The prototype pattern is used when the type of object to create](https://app.pluralsight.com/course-player?clipId=99469eb0-d8f0-494a-9cdc-b7cf21799f3f&startTime=7.93) [is determined by a prototypical instance,](https://app.pluralsight.com/course-player?clipId=99469eb0-d8f0-494a-9cdc-b7cf21799f3f&startTime=11.26) [which is cloned to produce a new instance.](https://app.pluralsight.com/course-player?clipId=99469eb0-d8f0-494a-9cdc-b7cf21799f3f&startTime=13.96) [Oftentimes, the prototype pattern is used to get a unique instance of the same object.](https://app.pluralsight.com/course-player?clipId=99469eb0-d8f0-494a-9cdc-b7cf21799f3f&startTime=17.24)

### [Concepts](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444)

[The concepts when choosing a prototype are when you](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=2.44) [are trying to avoid costly creation.](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=5.19) [Choosing a prototype is sometimes not as cut and dry as other patterns.](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=8.24) [I personally feel that it is often a refractory pattern and not a](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=12.45) [pattern that people usually think of upfront.](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=16.87) [Unlike the singleton,](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=19.84) [where you start off knowing that you want only one instance,](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=21.05) [you usually don't think of it in terms of something being expensive to create.](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=24.49) [Prototypes also avoid subclassing. Usually you create an instance of the](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=29.21) [actual prototype that you are trying to work with.](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=34.69) [The next concept can be a little confusing,](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=38.34) [too, but they typically don't use the keyword new.](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=40.39) [The first instance created might use the keyword new,](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=44.47) [but after that they are cloned. Although it can be implemented without it,](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=47.92) [there are good reasons to create an interface for your prototype instance.](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=53.31) [Prototypes are also usually implemented with some sort of registry.](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=58.99) [The original object is created and then kept in our registry.](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=64.09) [When another object is needed,](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=68.64) [we create a clone of that object from the registry. An](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=70.18) [example of the prototype pattern from the Java API is the](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=74.25) [java.lang.Object clone method.](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=78) [Although it is not all of a prototype pattern,](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=81.92) [it is definitely the basis for what we will design our prototype patterns around.](https://app.pluralsight.com/course-player?clipId=9cf2d984-c893-4b70-8578-9477b931e444&startTime=84.93)

### [Design Considerations](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2)

[The prototype pattern is an interesting pattern in that it](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=2.24) [is just changing the way that we call the keyword new. If an](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=4.97) [object is expensive to create,](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=8.43) [but we can get what we need by copying the member variables, then](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=10.35) [the prototype is a great fit. The prototype typically implements the](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=13.65) [Clone/Cloneable method and interface.](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=17.84) [This enables us to avoid using the keyword new. Typically, if our creation is](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=20.83) [expensive, it is going to be when we call the keyword new.](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=25.91) [Although we are using the clone method and essentially just making a copy,](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=29.34) [each instance is still unique. Different from patterns like the](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=33.96) [builder, costly construction is not handled by the client.](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=38.78) [In fact,](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=42.07) [I would say that the builder is the opposite of the prototype](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=42.61) [pattern. Different from the singleton, you can utilize](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=45.68) [parameters in the clone if you need to, but typically you don't.](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=49.09) [As the architect,](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=52.94) [you can choose whether you want to do a shallow versus deep copy. A](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=54.1) [shallow copy just copies the immediate properties, whereas a deep copy will copy any of its object references as well.](https://app.pluralsight.com/course-player?clipId=6264b4a0-dd7d-4736-abb4-68aa4aacb4e2&startTime=58.99)

### [Example: Statement](https://app.pluralsight.com/course-player?clipId=824de5c5-f670-4d9e-a33c-07f870a99222)

[Here is some code of an everyday example that you might run into.](https://app.pluralsight.com/course-player?clipId=824de5c5-f670-4d9e-a33c-07f870a99222&startTime=2.04) [Creating database statements can be expensive,](https://app.pluralsight.com/course-player?clipId=824de5c5-f670-4d9e-a33c-07f870a99222&startTime=6.42) [especially if we just want to swap out the parameters that we are](https://app.pluralsight.com/course-player?clipId=824de5c5-f670-4d9e-a33c-07f870a99222&startTime=8.61) [passing in so that we can run the query again.](https://app.pluralsight.com/course-player?clipId=824de5c5-f670-4d9e-a33c-07f870a99222&startTime=11.65) [In this example,](https://app.pluralsight.com/course-player?clipId=824de5c5-f670-4d9e-a33c-07f870a99222&startTime=14.2) [you can see that the constructor takes a lot of information, but the clone](https://app.pluralsight.com/course-player?clipId=824de5c5-f670-4d9e-a33c-07f870a99222&startTime=15.22) [method itself is very simple and takes advantage of the values passed in](https://app.pluralsight.com/course-player?clipId=824de5c5-f670-4d9e-a33c-07f870a99222&startTime=19.05) [previously to get another instance. Let's run this sample code in a demo and dive into it a little deeper.](https://app.pluralsight.com/course-player?clipId=824de5c5-f670-4d9e-a33c-07f870a99222&startTime=23.89)

### [Demo: Statement](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c)

[For this everyday example, we have created a statement object.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=2.5) [The more functionality we tie into this object,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=6.43) [the more expensive it would become to create it.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=8.74) [For our example,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=11.19) [you can see that we have a constructor that takes a string, an array](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=12.03) [list, and a custom object that we have created called Record. Not to](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=15.89) [stray very far from this demo, but if you are familiar with prepared](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=21.51) [statements in the Java API,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=25.04) [you will know that they are compiled against the database that we're](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=26.42) [using. This example is to help illustrate this same functionality](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=28.88) [without duplicating the prepared statement object.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=32.85) [Notice that this object also implements the cloneable interface,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=35.74) [which forces us to then implement the clone method.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=40.62) [From here, you can see that we just call the parent clone method, super.clone.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=46.01) [And this is what returns our instance that we then cast to our statement object.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=51.34) [The record object itself is just an empty object, but I have](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=57.57) [included it here to show what happens with object references](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=61.48) [that are included in our clone.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=64.36) [So we have a clone object inside of here and we](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=67.14) [implement the cloneable, it then, in turn,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=69.29) [creates an instance of this record object.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=71.88) [So let's see what happens.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=74.15) [I've created a demo over here where we build a simple string with](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=75.86) [some SQL in it and then pass in a list of parameters. In our](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=80.09) [parameters we've added a movie of Star Wars,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=85.52) [and then we have a record object, and right now our record object is empty.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=89.24) [And since it's passed by reference,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=93.96) [we could shove some values into this and then return](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=95.57) [that object to see what happens.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=98.8) [So let's go ahead and we create a statement and then run](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=101.41) [this statement and see what happens.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=104.63) [First,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=106.41) [we're going to go through and see what returns for our](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=106.77) [SQL, our parameters, and our record.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=109.055) [And then we're going to call clone on that, and our second](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=111.43) [statement is going to return some values that will also look](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=114.93) [like the same thing, our SQL, our parameters.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=118.86) [But I want you to pay particular attention to what](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=121.94) [happens with our record object here.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=123.94) [So let's run this, just right‑click and say Run As, Java Application.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=126.72) [And you'll see that it dumps out our SQL, so it says our select \* from](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=132.24) [movies where title = whatever, and that's what are are our SQL is that's](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=136.7) [injected there, and then returned here is well,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=141.87) [and then it returns our Star Wars instance or basically our](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=145.37) [array list that has those parameters added.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=149.09) [But it also returns the same object, our same record object.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=151.43) [So even though it went through and it did our clone inside of statement,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=157.14) [it just returned the references to the array lists that were passed in and](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=161.94) [the references to the record objects that were passed in.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=166.57) [This is an example of a shallow copy.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=170.04) [Now, if we were doing a deep copy,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=173.81) [it would return a new array list with those parameters passed into it,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=176.38) [and a new record object with whatever values the](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=180.98) [record object had stored in here, which currently is nothing.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=183.83) [But this is an example of a shallow clone because those objects are just](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=187.18) [getting returned the same as what we had passed in.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=191.27) [So it's not necessarily a true copy.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=194.45) [In fact,](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=197.24) [it's a little bit of a dangerous copy because we could go through](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=197.64) [and change the parameters in this array list, and it would reflect in both objects.](https://app.pluralsight.com/course-player?clipId=2392e92b-e2b0-4d80-8085-c8b57c8fa79c&startTime=201.68)

### [Exercise - Create Prototype](https://app.pluralsight.com/course-player?clipId=5adeb5a6-ce97-474e-af66-f73db91e7ab0)

[Now that we've seen an example of one,](https://app.pluralsight.com/course-player?clipId=5adeb5a6-ce97-474e-af66-f73db91e7ab0&startTime=1.63) [let's go ahead and create our own prototype in this exercise.](https://app.pluralsight.com/course-player?clipId=5adeb5a6-ce97-474e-af66-f73db91e7ab0&startTime=3.21) [We're going to create a prototype pattern, demonstrate the shallow copy of it, and then create an object with a Registry.](https://app.pluralsight.com/course-player?clipId=5adeb5a6-ce97-474e-af66-f73db91e7ab0&startTime=7.34)

### [Demo: Prototype](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de)

[For this example, I've set up a couple of base](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=2.19) [classes, and objects, just to implement,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=4.83) [to prove out our prototype pattern.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=8.58) [Let's start out with this Item class, and the Item class has](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=11.04) [just some very basic things in here, a string,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=14.91) [a price, that's a title, a double that's a price, and a](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=17.86) [string that's a URL; and what we're going to implement here](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=20.759999999999998) [is just some basic object look ups.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=23.59) [Think of it in terms of a company like Amazon, that if they were just](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=26.13) [to display an object for every item on their page,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=29.36) [it would become very expensive to create all those objects, especially](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=32.8) [if we're filling all of this type of information in. The sub classes of](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=36.38) [Item are Book, which just extends Item, and then has a number of pages](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=41.22) [just to have an additional filled in here to offset it from a basic](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=47.33) [item, and then a Movie, which we have that has a runtime associated with it.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=52.16) [The two other classes that we have associated with this are also a Registry.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=58.44) [We're going to go through and create this registry and](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=62.83) [fill it the rest of the way out.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=66.14) [Now this is probably a lot of the meat of the prototype and](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=67.74) [shows actually how the prototype works, and I've purposely](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=70.48) [left this createItem method empty.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=73.24) [There is a loadItems method down below that just shoves some basic](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=76.46) [information in there, but this is really where the heart and soul of](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=80.5) [the prototype is going to take place.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=84.12) [Lastly,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=87.25) [there's a demo class that we're going to run, and you're going to see how](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=87.7) [this executes what we're doing inside of our object.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=90.34) [So, let's go ahead and get started.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=93.3) [If we open up our Item object, this Item.java,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=94.84) [you can see that it's an abstract class,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=98.11) [meaning that we're going to want to have the Book and Movie](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=100.65) [objects implement the final functionality of this or determine](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=103.96) [what that final functionality is.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=107.57) [We want to go ahead and implement the clone method in here.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=109.34) [So we're going to say that this implements Cloneable, and the Cloneable method,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=112.56) [the Cloneable interface forces us to implement this method,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=121.32499999999999) [which is the clone method,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=125.74) [and you'll see it throws a CloneNotSupportedException, that's in case](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=129.14) [we have a subclass that doesn't do what we want it to do, and then it](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=133.02) [will also return this object of clone.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=137.13) [So let's go ahead and just save this right now, this object](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=141.31) [from the super object that we're cloning.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=144.63) [Now, if we have nothing unique in our sub objects,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=147.7) [this will go ahead and do the clone for us, and that's actually](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=151.7) [going to work for what we're doing right now.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=154.86) [So our basic clone method inside of our Item class gets](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=156.98) [extended by implementing the Cloneable interface.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=163.01) [I don't have to do anything to Book, and I don't have to do anything to Movie.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=166.34) [Now,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=170.64) [let's go ahead and go over to our registry now, and](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=170.87) [look at our createItem method.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=175.25) [So this is just a method that we've created, and it's our own basic registry.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=178.13) [There's many types of registry that you could implement out there,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=181.71) [but we're going to go ahead and just put a little try catch here, and inside](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=184.58) [of our catch, we're going to catch that CloneNotSupportedException, and just](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=191.09) [for sake of debugging, we'll do an e.printStackTrace, where you'd want to do](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=202.15) [something a little bit more official inside of our class if we were going to](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=207.62) [use it for production code.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=212.3) [Now let's go ahead and say item =, and we'll want to cast this because](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=214.8) [it returns an object, (items.get(type)).clone(). Now you'll notice that](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=220.76) [we have to cast this interface right here,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=235.97) [or cast our item being returned to our item, and that's](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=239.46) [because of our interface inside of here that implements](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=243.97) [Cloneable, forces it to return an object.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=246.885) [Unfortunately, the Cloneable interface was created with Java 1.0,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=250.45) [which means it doesn't have any knowledge of generic, so we](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=254.95) [can't tell it that we're going to pass in this object and](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=258.6) [return this type of object on our clone.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=260.85) [I actually think this is one problem with using just the basic](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=263.24) [Cloneable interface from the Java API, this is why I recommend](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=267.3) [oftentimes implementing your own interface and creating your own](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=272.44) [clone functionality that way. Not a big deal,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=276.91) [but one of the shortcomings, people will get a little leery when](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=279.58) [they start having to typecast things like this.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=282.4) [So now that we have our createItem method in our registry,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=285.52) [when we run this,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=289.64) [what it's going to do is it's going to go through, and the first time we run](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=290.88) [this, our registry has a loadItems method in there that goes through and](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=293.36) [creates some basic items and puts them in our registry.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=299.17) [So, we're going to create a movie, put that in our registry,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=301.96) [using this items.put, and then we're going to create a book, also put](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=305.73) [that in our registry using this items.put down here.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=309.02) [Now,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=312.14) [what this does is when we go through and want to get an instance of it, we](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=312.41) [say, give us back an instance of a movie, and give us back an instance of a](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=316.26) [book, and then we can change those values from there, and we get a unique](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=319.85) [instance every time. Now, one last thing to know is that our object types are](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=324.27) [being used by a key to look this up,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=332.34) [they're not necessarily a bad thing, but, I would maybe consider also](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=334.48) [implementing this with an enum. So let's look at our run method inside of our](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=337.53) [PrototypeDemo, I've got a basic main method set up where I create an instance of](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=341.97) [the registry, then I go ahead and create a movie,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=346.63) [and this is basically executing our clone method for us, and then I have my](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=351.96) [instance and can do whatever I want with it, override whatever values I](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=357.48) [want to, and manipulate whatever fields I want.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=360.72) [So it gives us a way to set up some defaults that we want for each](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=363.56) [object that's going to be returned without having that heavyweight](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=368.49) [expensive create an object every time.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=371.39) [Think if you had 10,000 objects on the page,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=373.32) [which, believe it or not,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=375.58) [is a common problem if you're implementing like a hibernate, where it](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=376.75) [creates a lot of objects and returns those, it can be very object or](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=379.85) [labor intensive to return that information.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=383.88) [So now let's go ahead and run this, it's going to go through,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=386.97) [create the registry,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=389.37) [create an instance of a movie, print it out, just so we can see what the](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=390.61) [address is, and all the various information with it, then it's going to](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=393.55) [create another one, and show us that it's also a unique instance, and go](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=397.24) [ahead and give us our values back.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=402.03) [So let's right‑click and say,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=405.2) [Run As Java Application, and you'll see down below here that we have](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=407.34) [returned an object of type Movie, with this address, and it says what its](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=413.57) [runtime is, and we didn't override that runtime, that's why it's still](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=419.06) [stuck at two hours, and then we have the Creational Patterns in Java, and](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=421.68) [then we created another instance which is moving down here with its own](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=427.37) [unique entrance, and its unique object address, and then it has a runtime](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=429.81) [of two hours as well, and it's looking at the Gang of Four.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=436.55) [URL we didn't set in either of them, so it's being returned as](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=440.21) [null in both instances. So you can see how it gives us a great](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=443.45) [way to set some default information up, which was all defined right here,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=446.72) [and then override those without having to call new each time.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=453.44) [So the only time we ever call new inside this application is right](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=456.99) [here on this line and right here on this line,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=460.77) [every time we want to get an instance of it,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=462.66) [we get an instance back of a movie,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=464.65) [and we don't have to use the keyword new anymore.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=467.64) [So it's a lot lighter weight object instantiation, and a lot faster,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=470.84) [but we're getting a unique instance each time,](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=474.89) [and that's the definition of that prototype, is getting a unique instance every time we ask for this object back.](https://app.pluralsight.com/course-player?clipId=0cb4d0a9-a451-4442-a269-d67110da24de&startTime=477.23)

### [Pitfalls](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae)

[What are some of the pitfalls of a prototype?](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=1.34) [Well, prototypes are often not used.](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=4.24) [This is a difference from the singleton pattern,](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=7.44) [where people usually overuse it.](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=9.82) [Another pitfall would be that you typically have to use it with another pattern.](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=12.84) [A loose definition of a framework versus a pattern is that](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=16.82) [if a pattern contains other patterns, it is a framework.](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=21.04) [This isn't always true,](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=25.5) [but it makes you sometimes question the use of a prototype because we](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=27.18) [typically have to implement that with some sort of registry.](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=30.63) [Lastly,](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=35.12) [a lot of times you want a deep copy and the clone](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=36.04) [interface only does a shallow copy.](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=38.7) [You can, of course, implement the functionality of a deep copy yourself,](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=41.08) [but that requires more coding yourself, and people start](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=44.79) [to second guess the validity of the pattern and whether it's solving anything for them.](https://app.pluralsight.com/course-player?clipId=8617f7b3-a2a6-4ab7-a7f8-29cea8169eae&startTime=47.5)

### [Contrast to Other Patterns](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5)

[Since it's easy to always just pick on the singleton pattern, let's](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=1.04) [compare the factory with the prototype pattern.](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=4.47) [A prototype is focused on lightweight construction,](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=7.7) [either through a copy constructor or using the clone method,](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=10.93) [like we demonstrated in our two examples. You can choose to](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=14.07) [do a shallow versus deep copy, but really in the end,](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=17.93) [you're looking at just creating a copy of yourself,](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=21.83) [even though in our one example we did an abstract class with a sub item,](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=24.85) [we're looking at creating a copy of whatever item instance we are.](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=29.76) [So we were a movie and we got copies of a movie, we weren't looking for](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=33.87) [different objects based off of what we were trying to do.](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=37.88) [The factory, on the other hand,](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=41.54) [is focused on dealing with flexible objects based upon your request. you can](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=43.3) [utilize multiple constructors instead of just the clone method,](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=48.95) [and it also utilizes creating a concrete instance of an object, and it](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=52.64) [is a fresh instance since we're utilizing the keyword new, so there](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=57.3) [aren't any programmatic defaults by nature.](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=61.11) [You remember back in our demo,](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=64.27) [we had set a default URL on a price and a run time, things like that; typically, that's not a feature that we program into a factory pattern.](https://app.pluralsight.com/course-player?clipId=c1c02ff3-7fc2-485f-9271-36fa5049eaf5&startTime=66.15)

### [Summary](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57)

[To summarize the prototype pattern, we are using this pattern to](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=2.44) [guarantee a unique instance every time we ask for it.](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=6.37) [A drawback or a side note on it is that it's often](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=11.05) [something that gets refactored in later,](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=14.2) [and that's because we're usually looking for it to help us with](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=16.48) [some performance issues inside of our application.](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=19.73) [So if we have an application,](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=22.8) [it's creating a lot of objects, we want to go ahead and implement this](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=23.93) [pattern so that it can help us obtain these objects faster without the](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=27.57) [heavyweight or overbearing nature of creating then using the keyword](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=31.81) [new every time we want an object.](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=36.36) [And lastly, I would note, don't always just jump to a factory.](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=38.58) [Look at a prototype to see if it will solve your problem with your](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=42.5) [current situation because a factory can often lead to other things that a prototype is nicely suited for.](https://app.pluralsight.com/course-player?clipId=a43cc954-277e-4b53-a845-2a7a4adf6e57&startTime=45.77)

## [Factory Method Pattern](https://app.pluralsight.com/course-player?clipId=2beac0ee-a616-43c8-88a9-434b62a5a6e2)

### [Introduction](https://app.pluralsight.com/course-player?clipId=2beac0ee-a616-43c8-88a9-434b62a5a6e2)

[Hi.](https://app.pluralsight.com/course-player?clipId=2beac0ee-a616-43c8-88a9-434b62a5a6e2&startTime=1.44) [This is Bryan Hansen, and in this module,](https://app.pluralsight.com/course-player?clipId=2beac0ee-a616-43c8-88a9-434b62a5a6e2&startTime=1.92) [we are going to look at the Factory Method designed pattern.](https://app.pluralsight.com/course-player?clipId=2beac0ee-a616-43c8-88a9-434b62a5a6e2&startTime=4.29) [The factory method pattern is in some ways the opposite of](https://app.pluralsight.com/course-player?clipId=2beac0ee-a616-43c8-88a9-434b62a5a6e2&startTime=7.48) [the singleton pattern and it probably is the second most used creational design pattern.](https://app.pluralsight.com/course-player?clipId=2beac0ee-a616-43c8-88a9-434b62a5a6e2&startTime=10.77)

### [Concepts](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134)

[The concepts when choosing a factory are that it](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=2.26) [doesn't expose instantiation logic.](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=5.721538461538463) [The client knows next to nothing about even the type](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=7.26) [of object that is being created.](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=10.385) [It is able to do this by deferring the instantiation or](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=12.26) [a creation logic to the subclass.](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=16.385) [All the client typically knows about is a common](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=18.26) [interface that the factory exposes.](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=22.117142857142845) [Factories are oftentimes implemented by an architecture or a](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=24.26) [framework and implemented by the user of that framework.](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=28.259999999999987) [This establishes a contract for how things will be](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=32.26) [implemented within the framework,](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=35.635) [but allowing flexibility for the end user to define how it can be implemented.](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=37.135) [Examples of this in the Java API are the Calendar,](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=42.974285714285735) [ResourceBundles, and a NumberFormat.](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=47.26) [Oftentimes people think that Calendar is a singleton because it](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=49.26) [has no arguments or a no arguments constructor,](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=54.26) [and factory methods can have arguments.](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=57.759999999999984) [The difference is that a calendar can return different](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=60.26) [subclasses of the calendar and the client is unaware,](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=63.26000000000002) [whereas with a singleton you are just getting a single instance of that implementing class.](https://app.pluralsight.com/course-player?clipId=7415d5c2-17f9-4bae-9e0f-7a2a7ed94134&startTime=66.26)

### [Design Considerations](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463)

[I view the factory as almost being the opposite of the singleton.](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=null) [The factory is responsible for creating instances and managing the lifecycle,](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=5.352) [at least the creation part of the lifecycle.](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=9.983578947368416) [Objects created are referenced through a common interface.](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=13.352) [Factories will also reference multiple concrete classes or implementations,](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=17.352) [but the client is unaware since they are referenced](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=21.6645) [through the common interface.](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=25.602) [The other key design principle is that the method to](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=27.352) [request an object is typically parameterized.](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=31.102) [These parameters are what are used to determine the concrete type.](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=33.352) [The UML diagram on the left shows the Factory,](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=38.352) [which is an implementation of the factory pattern.](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=41.52847058823528) [It has a factoryMethod,](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=44.352) [which will call and then return an interface to whatever](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=46.12977777777777) [object type we are attempting to build.](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=50.5742222222222) [The Factory itself refers to a concrete implementation that does](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=53.55199999999999) [the actual instantiation of our ConcreteObject.](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=57.55199999999998) [So we have our Factory class or our Factory abstract class that has a](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=60.12977777777777) [static factoryMethod that we are going to call,](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=66.352) [and based off of those parameters that we pass in, we're going to call a concrete instance that's going to return the object type for us.](https://app.pluralsight.com/course-player?clipId=8bb7483f-d772-4ffd-b1fe-8b7321a40463&startTime=69.15115789473683)

### [Example: Calendar](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074)

[Here's an everyday example we use in Java all the time,](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=1.54) [and that is the actual Calendar class.](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=4.36) [So in this snippet of code, we go ahead and do Calendar.getInstance,](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=6.77) [which returns an instance of the Calendar class.](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=10.33) [And then on the next line I call System.out.println. Now this is to show](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=13.64) [you the actual object type that's going to be returned.](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=17.75) [This particular class will show you the implementing class it's underneath.](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=20.9) [So the concrete implementation rather the in the](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=24.54) [abstract base class that we're using.](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=27.31) [And then finally,](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=29.12) [just to show an example of how the code is being ran, I used the](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=30.12) [calendar.get and pass in a parameter of Calendar.DAY\_OF\_MONTH just to show that the object is working, how we think it should be.](https://app.pluralsight.com/course-player?clipId=6ffab1bb-ad09-4e98-9c01-cbacb5ee6074&startTime=33.99)

### [Demo: Calendar](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c)

[Here is that same example code that we were just looking at.](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=1) [I want to point out a couple of things.](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=4.382857142857143) [First when we run our code,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=7.253333333333333) [it's going to go ahead and call our Calendar.getInstance,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=8.92) [but we're not going to know exactly what type of](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=12.25333333333334) [calendar's going to be returned,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=15.420000000000003) [and that's why I use this System.out.println here.](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=16.920000000000005) [So if we go ahead and run that,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=19.92) [you'll see that the java.util.GregorianCalendar down here gets](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=23.11999999999999) [dumped out from that System.out.println,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=27.119999999999976) [so the Gregorian calendar is the actual concrete implementation](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=30.1075) [that's being called underneath this factory instance.](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=34.045) [Now another thing that I wanted to point out is that we](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=36.498947368421064) [can use different types of calendars,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=40.28842105263161) [and that's what makes this different than some of the other](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=42.096470588235306) [design patterns because it is parameterized.](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=45.33176470588239) [So we can do Calendar cal = Calendar.getInstance,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=48.02526315789474) [and you can see that there are different time zones available,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=52.762105263157906) [locales,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=58.42) [different things like that that we can call to get various types of calendars.](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=58.92) [So we could use a time zone from the East Coast,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=65.92) [the West Coast, Pacific/Mountain, whatever,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=69.97263157894733) [and get our calendar with that.](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=71.8147368421052) [So instead of like the singleton where I'm going to just get](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=73.82) [whatever object it says is available for me,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=77.41999999999996) [I can get different types of calendars based just](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=80.02526315789473) [simply off of time zone or locale,](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=83.3410526315789) [so that's a nicer feature of the Calendar API, and really that's a](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=85.92) [feature of the factory method over singletons or some of the other creational design patterns that are out there.](https://app.pluralsight.com/course-player?clipId=1985af20-576c-4fa2-b2a2-8655d6749a8c&startTime=91.39368421052629)

### [Exercise - Create Factory](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686)

[Now that we've seen an everyday example,](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=1.94) [let's look at what it would be like to write our own factory and go](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=4.1) [ahead and implement an exercise doing so. First,](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=7.64) [we're going to start off by creating some basic pages.](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=10.93) [So this example is going to be a canned tool for generating websites,](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=13.74) [much like if you went to wordpress.org or a site like that](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=18.46) [that will generate websites for you.](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=23.19) [The next piece that we're going to do is actually](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=25.44) [create the website from the pages, and this is just going to be a templated tool.](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=27.29) [We're not going to actually implement a website.](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=30.92) [The next step will be to create the concrete classes that actually](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=33.9) [implement those features of creating the website.](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=38.11) [And then finally, we're going to create our factory.](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=41.48) [We're going to polish that last bit of our factory off by creating an enum that switches the logic in the factory based off of that enum.](https://app.pluralsight.com/course-player?clipId=06f88ce6-e2ba-4a5a-9fdb-acb307b4e686&startTime=44.5)

### [Demo: Factory](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f)

[One of the drawbacks to the factory pattern is that it takes a lot of moving](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=1.74) [pieces to really demonstrate what we're trying to do with it.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=6.43) [I've gone ahead and set up some scaffolding,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=9.95) [some skeleton pages to help facilitate this, and so](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=12.69) [we're not just watching me type forever.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=16.39) [We're going to create an application that's the](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=18.84) [equivalent of a website generator.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=21.42) [If you went to something like wordpress.org and requested a new website,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=23.11) [you're going to get a bunch of canned pages to begin with.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=26.93) [Now, we're not going to create a website,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=29.31) [but you'll get the idea of what these canned pages are and how this works.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=31.02) [So I've gone ahead and created a page abstract class, it's just a base class,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=34.4) [and it's just to facilitate showing what you would do in an actual](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=39.24) [application that you're trying to implement.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=42.58) [So let's go ahead and add one more page in here.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=45.04) [Just so you can see how I'm creating these. I'm saying New,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=47.29) [Class, and I'm going to do a CartPage and this will be for a](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=50.39) [shopping cart in our basic application here.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=53.96) [We're going to try and we're going to create two types of](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=58.26) [website, a blog versus a shopping or a, you know,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=60.87) [Amazon style e‑commerce type application.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=67.16) [So we have our CartPage. Again, there's our](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=70.21) [AboutPage, our ContactPage our PostPage.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=72.63499999999999) [Nothing too extravagant going on there.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=74.76) [Now,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=77.85) [the next thing I'm going to do is I'm going to go through](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=78.02) [and create the website abstract class.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=80.86) [Now, this is going to be the start of our factory pattern.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=84.65) [So I want to go out here and right‑click and say New,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=87.33) [Class, and we're going to say that this is a website.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=90.64) [It is abstract.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=95.94) [And let's go ahead and just click Finish on this. Now inside of here,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=98.34) [we're going to add a few things. We're going to add all of](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=102.84) [our pages that were going to create, so we're going to say protected ListPage,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=105.15) [pages, = new ArrayList, and save this.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=114.54) [So now what we've got is just some holders for the pages are going to create.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=124.26) [Now,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=128.45) [this point is really critical, and I didn't want to just copy and](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=128.92) [paste this in because this is kind of some of the key points of](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=131.81) [what's going on with the factory pattern here.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=134.77) [The next thing we're going to do is we're going to just put a default](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=137.14) [no‑args constructor in here where we say public Website and we call](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=139.34) [this.createWebsite. Now we haven't created this method yet,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=146.8) [so it's going to complain that this doesn't exist yet.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=152.32) [Now, while we're at it,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=155.64) [we might as well generate a getter for these, Source,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=156.65) [Generate Getters and Setters, and we just want a getter](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=162.995) [for the get pages. So we say OK.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=166.2125) [And now we've got our basic structure there.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=168.94) [Now the last thing we want to do is create our public](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=170.89) [abstract void createWebsite method.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=173.25) [Now, this is the crux of the the factory method pattern here.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=182.5) [So all of the base classes, excuse me, the concrete classes,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=188) [that implement this base class, are going to go through and](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=192.04) [override this method to generate those.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=197.17) [So now that we've got our website, let's go through and start off by making](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=199.9) [our first type of website which is going to be a blog.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=203.59) [So we'll say New, Class, and it is a blog, and it's super class](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=206.68) [is going to be Website, and click Save that.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=213.56) [Now,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=217.09) [you'll see it automatically overrides that method for us](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=217.25) [because we haven't declared it as abstract,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=219.88) [which is what we want.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=221.61) [So we're going to say pages.add, and we're going to add a new PostPage and](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=223.02) [we'll add a pages.add, and let's do a new AboutPage. And you can see how we're](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=233.35) [putting pages in here that are specific to the blog.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=244.09) [So say new CommentPage, and just for good measure,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=248.54) [let's add one more for the ContactPage.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=255.57) [So we'll say pages.add and will do new ContactPage.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=258.23) [So you can see how the base class doesn't have anything to do with the creation,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=263.94) [but rather the concrete implementation does.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=270.15) [So the blog class, which is implementing that factory method,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=272.99) [is what's concerned about creating the implementation that we're looking at.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=277.15) [Now that we have our blog, let's go ahead and do the same thing and create our](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=282.01) [shop website. I'm going to go over here and say New, Class,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=286.05) [and we want to call this Shop, and it's super class is](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=290.86) [going to be Website, click Finish.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=294.27) [And we're going to do basically the same thing. For the shop, we want](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=297.61) [to do pages.ad, and we're going to do a new CartPage, and then we](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=301.18) [want to do pages. add, and you see how the content here is different](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=313.53) [for the website, for the shop website versus the blog website. And](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=318.93) [the concrete implementation is what's concerned with how that gets](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=325.86) [implemented.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=329.68) [But we're still going to return the website interface or](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=330.92) [contracts. We are using an abstract class here, which is](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=335.5) [giving us the same thing as an interface.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=338.1337500000001) [We could implement a website interface and achieve the same thing.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=340.49) [By interface, we're really talking about the contract and not the](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=345.44) [keyword of interface inside of our application.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=349.1) [So just for good measure here we'll throw a SearchPage and save that.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=352.04) [So now we have our shop and our blog.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=358.51) [Now we can move on to our actual factory.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=363.94) [Now,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=367.6) [I have created a class out here called WebsiteFactory with nothing in it yet,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=369.34) [and just to show you that typically when we have a class,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=374.42) [the factory for it is just, it's more of a rule of thumb, but is usually](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=378.47) [named whatever that class is with the factory name.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=383.51) [So website, we have a WebsiteFactory, pretty common sense there.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=385.93) [So we're going to say public static Website getWebsite,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=389.85) [and this is where the factory method really steps into play here](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=399.3) [because we have those concrete implementations.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=406.79) [We want those to be concerned with how they're created,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=409.53) [not this base class, but we don't want people to be able to](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=412.38) [access those concrete classes on their own. So we're going to](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=415.94) [put a sightType inside of here,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=419.48) [and we can do a basic switch statement here, and as of Java 7,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=423.94) [we can do a switch on a string.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=427.67) [So we can say sightType and pass into here some basic cases.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=428.89) [So we'll say case and we'll do blog and we can say return new blog.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=435.1) [And now let's shift this in here, and we'll say case shop and do the same thing](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=452.24) [here, pass in for our arguments, not really our arguments,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=461.88) [but our meat of our method here, as a new Shop.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=471.28) [And then we can always,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=478.24) [which is a nice feature of this, add a default case where we](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=479.8) [just return null. So now we have our basic factory method and](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=484.76) [where we're getting our method from,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=492.75) [we have our factory that calls based off these types, but our true method is](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=494.23) [occurring, our factory method pattern, is occurring in this createWebsite](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=498.74) [method that's overridden from our website itself.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=502.41) [And that's why I was saying this is a little bit more complicated of a](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=505.74) [pattern because there are a couple of moving pieces to it.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=508.66) [So our factory is going to call the new no args constructor for a blog,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=512.32) [and it could have arguments too, it doesn't have to be a no arguments,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=518.481875) [which is then overridden from Website that calls our constructor at](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=521.4649999999999) [this.createWebsites. And this is where our factory method is coming the](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=526.9) [case, it's called a factory method, a lot of people will shorten that to](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=531.39) [just factory.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=534.8) [It is a factory method.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=535.41) [Our method is what's concerned about our instantiation of a](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=536.63) [createWebsite, which in our blog or our shop, then goes through and](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=540.0799999999999) [actually builds what our application is going to do here.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=544.42) [So now to demo this, we can go over to our Factory Demo and we can say](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=547.69) [WebFactory.get, and I have this basic code already set up for us to where I](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=551.7478125) [just call our WebsiteFactory.getWebsite and we can pass in a blog and get that](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=559.85) [string back and call system.println with us,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=564.86) [and it will show what pages we have built in that. And then we](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=567.9) [call WebsiteFactory.getWebsite for our shop,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=570.89) [and it will also print out what pages we have available to us there.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=573.54) [So let's go ahead now and run this.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=576.81) [Save that, and you'll see that the pages are different down](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=581.74) [here based off of what type of site we have.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=585.96) [So for our blog,](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=589.37) [we have a PostPage, an AboutPage, ContactPage, CommentPages, those](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=591.37) [types of things. Where for our shop, we have our ItemPage and our SearchPage, and those types of things, so.](https://app.pluralsight.com/course-player?clipId=e7350162-da53-4e10-8cc2-b67674d3c69f&startTime=596.32)

### [Demo: Enum](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5)

[One thing we can do to make our factory a little bit](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=1.64) [better is to get rid of these string liberals in here and](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=4.23) [convert this over to enum. Now,](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=8.33) [this is definitely not a requirement of the factory pattern,](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=10.36) [but it really is good coding practices.](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=13.44) [And since patterns are an architectural type principle,](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=15.62) [really, there is a better way to do this and we want to show you that.](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=19.39) [So what we're going to do is we're going to go up here](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=22.44) [and right‑click and say New, Enum,](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=24.47) [and we're going to call this the WebsiteType enum. And inside of here,](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=28.04) [we're going to create two of them. We're going to create](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=34.13) [BLOG and SHOP, and just store those two.](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=35.715) [Now, if we come over to our factory,](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=42.4) [we can get rid of this string here by saying this is a WebsiteType.](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=46.04) [And the beauty of this is when we switch on an enum,](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=52.15) [we don't even have to preface it with whatever that type is.](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=57.69) [I can just say BLOG and SHOP, and it already knows what I'm trying to do.](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=60.46) [So I don't even have to say WebsiteType.BLOG and](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=68.09) [WebsiteType.SHOP, it already knows since we're switching on](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=71.92) [an enum to just go ahead and do that.](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=73.78) [I really like this style of coding with enums. As we go over here to our](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=75.96) [demo and now pass in WebsiteType.BLOG and WebsiteType.SHOP,](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=81.38) [and we've gotten rid of all of those string literals now. If we go](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=97.74) [ahead and run our code again, just to make sure that's working](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=101.23) [correctly, you see it works just fine and we've gotten rid of those](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=103.98) [strings which could break and become a little bit more brittle in our code down the road.](https://app.pluralsight.com/course-player?clipId=7db08666-e61e-458a-ad50-d595f0c49bc5&startTime=106.6)

### [Pitfalls](https://app.pluralsight.com/course-player?clipId=716d3cd0-a421-41be-80a0-03e4f9daeeca)

[The pitfalls of a factory method pattern are namely complexity.](https://app.pluralsight.com/course-player?clipId=716d3cd0-a421-41be-80a0-03e4f9daeeca&startTime=0) [You will notice compared to the other creational design patterns the factory](https://app.pluralsight.com/course-player?clipId=716d3cd0-a421-41be-80a0-03e4f9daeeca&startTime=5.5) [pattern is almost double the amount of code to demonstrate it.](https://app.pluralsight.com/course-player?clipId=716d3cd0-a421-41be-80a0-03e4f9daeeca&startTime=8.941176470588239) [The part that most people often get wrong when implementing the factory is](https://app.pluralsight.com/course-player?clipId=716d3cd0-a421-41be-80a0-03e4f9daeeca&startTime=11.953947368421055) [that creation doesn't take place in the factory itself,](https://app.pluralsight.com/course-player?clipId=716d3cd0-a421-41be-80a0-03e4f9daeeca&startTime=15.375) [but rather in the subclasses of the type of factory method we are creating.](https://app.pluralsight.com/course-player?clipId=716d3cd0-a421-41be-80a0-03e4f9daeeca&startTime=18.217105263157883) [The factory method pattern is also the pattern](https://app.pluralsight.com/course-player?clipId=716d3cd0-a421-41be-80a0-03e4f9daeeca&startTime=23.022058823529406) [that's generally not refactored into.](https://app.pluralsight.com/course-player?clipId=716d3cd0-a421-41be-80a0-03e4f9daeeca&startTime=26.316176470588218) [You need to design from the beginning that it's going to be a factory and then plan accordingly.](https://app.pluralsight.com/course-player?clipId=716d3cd0-a421-41be-80a0-03e4f9daeeca&startTime=28.375)

### [Contrast to Other Patterns](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36)

[To contrast the factory pattern with another pattern,](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=null) [we're actually going to do the same comparison we did with the singleton.](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=3.524888888888889) [The singleton and the factory are almost the exact opposite of one another.](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=9.056) [The singleton returns the same instance.](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=13.931) [It has one constructor method with no arguments and no](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=16.754529411764707) [interface and no subclasses typically,](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=21.460411764705867) [whereas the factory returns various instances and has multiple constructors](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=24.06433333333334) [or arguments that we can add to a constructor type method.](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=29.39766666666669) [It is very interface driven, and when I say interface driven,](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=34.59766666666667) [I don't always mean that we're implementing an interface,](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=38.73099999999998) [but it can be an abstract class or a contract, it's very contract driven.](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=41.430999999999955) [There are always subclasses involved.](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=47.431) [You can't have a factory pattern without having some](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=49.931) [fashion of a subclass involved.](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=53.145285714285684) [And it's easily adaptable to your environment,](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=54.931) [so a lot of frameworks are written using the factory pattern where you implement them per environment as you want to use them.](https://app.pluralsight.com/course-player?clipId=5b37a640-959a-4d6d-9d6c-7313bde38d36&startTime=58.0421111111111)

### [Summary](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2)

[Just to recap the factory pattern and how it differs](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=null) [from some of the other patterns, it is parameter driven, in fact,](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=3.5095000000000005) [this is one of the few creational patterns that is parameter driven.](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=7.876142857142858) [It solves complex creation in a different fashion than all the other patterns.](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=13.019) [So we had the builder that was involved with enforcing a contract](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=18.019) [with our constructor and multiple parameters,](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=22.254294117647042) [but it didn't support parameter driven construction meaning that](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=24.46344444444444) [if we wanted to choose a type at run time,](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=28.463444444444427) [the factory is really the only one that deals with that.](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=32.382636363636365) [One of the drawbacks of the factory is that it can be a little bit](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=36.019) [complex and it really is the opposite of a singleton.](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=40.49900000000002) [So if you're looking at a singleton,](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=43.019) [and it doesn't seem like it's the right fit for it,](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=44.609909090909085) [you probably need to be looking at a factory.](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=47.10990909090908) [Most of these patterns stand on their own,](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=49.769000000000005) [so unlike traditional courses we do here where we try to talk](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=52.56900000000002) [about the next module that's going to come up,](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=56.382636363636365) [all of these patterns will stand on their own two feet,](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=58.837181818181826) [except for the factory has kind of a cousin,](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=62.069) [which is the AbstractFactory,](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=65.219) [and that is the next module we're going to cover in this series is the AbstractFactory and how that applies to the factory method pattern.](https://app.pluralsight.com/course-player?clipId=e9fead45-db5c-4bfd-81b7-c6ad226553c2&startTime=66.61899999999997)

## [AbstractFactory Pattern](https://app.pluralsight.com/course-player?clipId=ee671b49-7adc-4375-98ca-ac896cd4b129)

### [Introduction](https://app.pluralsight.com/course-player?clipId=ee671b49-7adc-4375-98ca-ac896cd4b129)

[Hi. This is Bryan Hansen, and in this module,](https://app.pluralsight.com/course-player?clipId=ee671b49-7adc-4375-98ca-ac896cd4b129&startTime=1.94) [we're going to look at the abstract factory design pattern.](https://app.pluralsight.com/course-player?clipId=ee671b49-7adc-4375-98ca-ac896cd4b129&startTime=4.51) [The abstract factory is very similar to the factory method pattern, and in fact, is typically implemented as a factory of factory patterns.](https://app.pluralsight.com/course-player?clipId=ee671b49-7adc-4375-98ca-ac896cd4b129&startTime=8.43)

### [Concepts](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01)

[The concepts when choosing an AbstractFactory are](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=1.867) [that it is a factory of factories.](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=4.867) [Although it can be implemented without using the factory method pattern,](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=7.867) [more often than not it is.](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=12.396411764705885) [It is typically summarized as a factory of related objects.](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=14.867) [It also takes the concept of a common interface a step further.](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=18.55121052631579) [The common interface is implemented throughout the](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=23.367) [AbstractFactory and its underlying factories,](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=26.867) [and just like the factory method pattern is deferring the](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=29.242) [instantiation or creation logic to subclasses as well.](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=32.992) [Examples in the Java API are the DocumentBuilder from the XML APIs.](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=36.46699999999999) [There aren't a lot of other examples because it is often implemented in frameworks and not just in the standard Java API.](https://app.pluralsight.com/course-player?clipId=530b53d3-7cc8-4e17-9ecd-155fda468a01&startTime=42.867)

### [Design Considerations](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45)

[The design principles when implementing an AbstractFactory are that you](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=1.532) [want to group a collection of factories together.](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=5.976444444444446) [The factory is still responsible for the lifecycle itself,](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=9.532) [and it has a common interface that is carried throughout the AbstractFactory,](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=13.032000000000004) [as identified on the UML on the left here,](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=18.2195) [down through the ConcreteFactory, and finally to the implementing class below.](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=23.282) [Just like the factory,](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=28.41435294117646) [there are concrete classes that are finally](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=30.061411764705866) [returned from the underlying factory.](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=33.032) [The AbstractFactory also has parameterized create methods just](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=35.532) [like the factory method pattern does as well.](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=39.532) [One key point is that the AbstractFactory is typically built using composition](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=42.64311111111108) [where that is not the case with the factory method pattern,](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=47.47936842105263) [so one very key distinction there.](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=52.68989473684209) [The AbstractFactory itself implements a very good use of objects for development through composition.](https://app.pluralsight.com/course-player?clipId=64198e24-18f0-4ad7-ad7b-6302c793cf45&startTime=55.532)

### [Example: DocumentBuilderFactory](https://app.pluralsight.com/course-player?clipId=e272a2bd-e1f6-4e1a-8dc9-b0fde0842330)

[One of the few examples in the Java API is the](https://app.pluralsight.com/course-player?clipId=e272a2bd-e1f6-4e1a-8dc9-b0fde0842330&startTime=1.191) [DocumentBuilderFactory from the Java XML APIs.](https://app.pluralsight.com/course-player?clipId=e272a2bd-e1f6-4e1a-8dc9-b0fde0842330&startTime=7.0145294117647055) [The DocumentBuilderFactory itself is an](https://app.pluralsight.com/course-player?clipId=e272a2bd-e1f6-4e1a-8dc9-b0fde0842330&startTime=10.191) [AbstractFactory, the DocumentBuilder is a factory,](https://app.pluralsight.com/course-player?clipId=e272a2bd-e1f6-4e1a-8dc9-b0fde0842330&startTime=13.619571428571428) [and then the document itself is the concrete class](https://app.pluralsight.com/course-player?clipId=e272a2bd-e1f6-4e1a-8dc9-b0fde0842330&startTime=16.476714285714284) [that is created from those factories.](https://app.pluralsight.com/course-player?clipId=e272a2bd-e1f6-4e1a-8dc9-b0fde0842330&startTime=21.191) [The document is actually an interface,](https://app.pluralsight.com/course-player?clipId=e272a2bd-e1f6-4e1a-8dc9-b0fde0842330&startTime=24.191) [and the factory chooses an implementation of that concrete](https://app.pluralsight.com/course-player?clipId=e272a2bd-e1f6-4e1a-8dc9-b0fde0842330&startTime=27.014529411764716) [class to return to the end user or the client. Let's look at a live example of this code.](https://app.pluralsight.com/course-player?clipId=e272a2bd-e1f6-4e1a-8dc9-b0fde0842330&startTime=31.249823529411792)

### [Demo: DocumentBuilderFactory](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717)

[Just a quick walkthrough of this code.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=1.593) [We start off by creating a simple XML document that we've stored into a string,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=4.043) [and then to utilize that in the DocumentBuilderFactory and](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=9.181235294117648) [the DocumentBuilder, we have to convert that into a](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=11.828294117647067) [ByteArrayInputStream like we've done here.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=15.2805) [The AbstractFactory in this example is the DocumentBuilderFactory.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=18.093) [Now the reason it's an AbstractFactory is because we don't know](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=22.593) [what the underlying implementation of this is,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=27.122411764705866) [nor do we know what the underlying implementation of](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=29.828294117647058) [the factory is when we get it.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=31.94594117647058) [All we know is that we get our document and can](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=33.593) [eventually run and use that document.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=38.12241176470591) [Once we have our document,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=40.593) [we can get an element and work with it and see what we have there.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=42.09299999999999) [I've added a couple of System.out.printlns at the bottom of this just for](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=46.593) [some rudimentary debug purposes to show you what the actual implementation](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=51.32984210526319) [of the AbstractFactory is and the factory is,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=54.692999999999984) [so you can see we call a getClass on them down below here.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=57.09299999999996) [Now if we go ahead and run this,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=62.41908695652175) [you'll see that we have a Root element of document that gets displayed there,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=66.2451739130435) [but we also get our actual implementation class of that](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=73.29888235294119) [DocumentBuilderFactory and that DocumentBuilder.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=78.593) [You can see that our DocumentBuilderFactory is](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=80.77481818181819) [com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderFactoryImpl,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=84.593) [and then we have that same thing for the DocumentBuilder itself,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=90.71064705882354) [so we're using the xerces implementation.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=98.593) [Now if you don't know much about this factory,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=101.593) [and it's not really an important part of the demo,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=104.45663636363632) [but you can swap out this implementation by switching up a command line](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=107.63845454545445) [parameter that tells it to use a different library.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=113.32984210526315) [Most people are happy with what the default implementation of it is,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=117.593) [which is the apache xerces implementation,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=121.12241176470592) [but that's why this is an AbstractFactory.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=122.9459411764706) [We know nothing about what's going on from a client](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=125.41652941176476) [perspective of which implementation is chosen,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=129.093) [and we don't care what factory's returning as well.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=132.093) [All we care is that we get our document and it's usable.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=136.593) [So all of those hideous details of creating that are hidden from us,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=140.02157142857132) [but we can swap them out if we need to.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=143.81039130434786) [Now, as I've mentioned a couple of times now,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=146.85386956521754) [the xerces and XML APIs are more of a framework for parsing XML documents,](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=149.593) [and that's why this is dumping out that code as part of a](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=155.59299999999982) [framework and it's chosen for us under the hood.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=161.39300000000003) [So, you can see the variables are named appropriately for an AbstractFactory](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=164.593) [versus factory in this example just to help illustrate which pieces of this are in fact the AbstractFactory versus the factory.](https://app.pluralsight.com/course-player?clipId=a186f1bf-ca91-404f-83b1-3ac2970ed717&startTime=169.53417647058816)

### [Exercise - Create AbstractFactory](https://app.pluralsight.com/course-player?clipId=61161182-b152-452d-97ef-aacc55c6a35a)

[Now that we've seen a demo of an AbstractFactory,](https://app.pluralsight.com/course-player?clipId=61161182-b152-452d-97ef-aacc55c6a35a&startTime=0) [let's go ahead and build our own.](https://app.pluralsight.com/course-player?clipId=61161182-b152-452d-97ef-aacc55c6a35a&startTime=2.6774999999999993) [We will first walk through some code that I've set us as a](https://app.pluralsight.com/course-player?clipId=61161182-b152-452d-97ef-aacc55c6a35a&startTime=4.76) [template because it is quite a lengthy example.](https://app.pluralsight.com/course-player?clipId=61161182-b152-452d-97ef-aacc55c6a35a&startTime=8.474285714285713) [Then we're going to build the actual AbstractFactory itself.](https://app.pluralsight.com/course-player?clipId=61161182-b152-452d-97ef-aacc55c6a35a&startTime=10.76) [Since it calls a factory, we will build out that piece,](https://app.pluralsight.com/course-player?clipId=61161182-b152-452d-97ef-aacc55c6a35a&startTime=14.260000000000003) [and then once we're all finished with that we'll finally produce a final product and run that demo.](https://app.pluralsight.com/course-player?clipId=61161182-b152-452d-97ef-aacc55c6a35a&startTime=19.76)

### [Demo: AbstractFactory](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c)

[To not make this demo take so long,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=3.057) [I've gone ahead and set up some templated scaffold code that](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=5.723666666666666) [we're going to walk through really quick.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=9.390333333333333) [This implements a factory for creating credit cards,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=12.001444444444447) [so basically think of it as you're submitting an application for a credit card.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=15.112555555555561) [This application is going to choose for you and create the right](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=20.056999999999988) [credit card based off of the criteria you put in.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=23.87518181818182) [It dabbles a little bit on some strategy or some behavioral patterns too,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=27.96609090909092) [and I'm not going to go into too much detail there,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=32.914142857142856) [but this is a great example and one that I've had to do inside of](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=36.05699999999999) [applications before of how you can use this in your day‑to‑day code.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=40.60245454545456) [The AbstractFactory, in this case, is the CreditCardFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=45.057) [and we're going to go ahead and implement this out,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=47.8805294117647) [but we'll walk through the other code first.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=51.44588888888889) [There are concrete factories in here, or factory methods,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=54.556999999999974) [such as this VisaFactory.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=58.057) [Now it's given me some red errors here because I've](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=59.390333333333345) [deleted some of the AbstractFactory code,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=62.7236666666667) [so it's saying hey, you're not overriding anything in this.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=64.67238461538462) [You can see that as we go onto this factory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=68.057) [and if you've gone through the Factory Method Design Pattern module, that](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=71.05699999999997) [it's coming in here and switching on a type and saying oh,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=74.68857894736841) [you want a gold card?](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=78.47805263157888) [Well I'm going to return the VisaGoldCreditCard and defer to the](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=80.057) [factory below to handle creation, and I'm just using the default](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=85.557) [no‑args constructor in the factory below because I really don't care](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=90.39033333333332) [about that implementation of this detail.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=94.66811111111106) [I'm more concerned about the AbstractFactory.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=97.78427272727274) [Now there is some validator code in here for validating the](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=101.057) [credit card information you're putting in,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=105.58641176470583) [and we're just going to show how you can plug those pieces in.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=108.057) [I'm not going to take the example much deeper than that.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=110.76533333333327) [So there are a bunch of different things in here.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=113.057) [There are some CardTypes, so we have GOLD, and PLATINUM.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=115.1622631578948) [We have a basic CreditCard class that goes through and](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=117.057) [has a cardNumberLength and a cscNumber.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=121.432) [It also has some Validator code, so I put in a PlatinumCreditCard validator,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=124.057) [a PlatinumCreditCard and a PlatinumValidator, and a](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=130.93935294117648) [GoldCreditCard and a GoldValidator,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=134.257) [which are just stubs for you actually putting your own code into.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=135.85700000000003) [The real catch here is that we have our AbstractFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=141.057) [which is our CreditCardFactory, and then our concrete factories,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=146.057) [which are an AmexFactory here, which is pretty empty,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=150.27922222222227) [and then a VisaFactory here, which this one is implemented more complete.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=154.2792222222224) [Well let's go ahead and get started.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=159.25700000000003) [So we're going to open up the CreditCardFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=162.057) [and since this is our AbstractFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=164.72366666666673) [this comes in here and determines, based off of what we're doing,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=166.7236666666668) [which card the person should get.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=170.91414285714288) [So we're going to say if(creditScore is greater than 650,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=173.057) [then we're going to give them an American Express.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=179.3727894736841) [This doesn't mean you have to have a great credit score to](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=185.057) [have an American Express or if you have a bad credit score](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=187.78427272727262) [you're going to get a Visa.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=190.6452352941176) [It's just a simple example based off of conditions.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=192.40994117647048) [So we're going to say AmexFactory.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=195.057) [Now we're going to save this, else we'll return them a Visa,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=199.39033333333336) [return new VisaFactory.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=208.057) [Now the way an AbstractFactory works is there are also some interfaces](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=210.70405882352938) [that we're going to pass through to our concrete classes and concrete](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=221.29229411764695) [factories down below. So we're going to say okay,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=227.50144444444456) [we want a public abstract CreditCard meaning that every factory](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=231.707) [that we return is going to implement this method.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=238.20700000000005) [So we're going to get the credit card type that we have,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=244.057) [so we say getCreditCard,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=247.32972727272735) [and this is going to take in an enum that we've created called CardType,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=248.42063636363648) [and I showed that earlier, it was the CardType of platinum versus gold,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=252.51853846153855) [so we'll say cardType, and then we're also going to just,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=258.26752631578955) [because a factory is a group of similar factories,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=264.057) [we're going to put a validator in here as well.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=267.84647368421065) [We'll say public abstract, and we'll do Validator getValidator,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=272.057) [and we'll also pass in a CardType here as well.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=279.6359473684213) [And basically the logic behind this is that you're going to apply for a card,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=288.057) [we're going to give you back the card you qualify for,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=293.147909090909) [and then we're going to validate that you can actually have](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=296.96176190476206) [the card that you say you have access to.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=301.15223809523854) [So we have our Validator and our CreditCard.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=304.3903333333333) [Now let's go ahead and open up our ConcreteFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=307.057) [which is our AmexFactory that we've created right here.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=309.55699999999996) [So the AbstractFactory just says I'll return you whichever factory](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=312.057) [you need to fulfill these two methods down here.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=316.2675263157896) [So we're going to open up our AmexFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=320.057) [and inside of here we want to go ahead and build a](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=323.3297272727275) [switch based off the card type.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=328.2388181818187) [Now this is one of the things people don't like about the AbstractFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=329.85699999999997) [so we're going to say switch, I'm going to do cardType, let's](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=332.4569999999998) [actually implement out our switch statement first,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=338.4570000000002) [and we'll say cardType.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=342.6570000000004) [Now I've said this in past examples,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=345.057) [but this is another case that I just get to bring this up.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=347.1570000000001) [I really like using switches for this because I can,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=351.057) [as I'm passing in an enum in this switch,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=354.69336363636376) [I have the values here already available as my cases,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=357.9660909090911) [so I can say case GOLD, and I can return,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=365.35699999999986) [return new AmexGoldCreditCard.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=374.35699999999963) [And if we really wanted to get into the finite details](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=377.057) [of the factory method pattern here,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=380.7236666666665) [we don't have to rely on just the default no‑args constructor, we could](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=382.72366666666636) [do something more with the template method pattern.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=389.35111764705863) [I'll say PLATINUM, and we'll do AmexPlatinumCreditCard and save that.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=394.057) [So now we have our factory in place that's going to go ahead and defer](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=402.057) [to this instance to create what is being returned.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=408.93199999999973) [So whatever the credit card is, whatever the creation of that credit card is,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=413.057) [our parent class has no knowledge of what's going on](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=417.4780526315791) [here or how those were created though.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=419.89033333333344) [So our AbstractFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=421.057) [it barely even knows which factory we've chosen let alone what](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=422.22366666666676) [type of credit card's going to get returned, it doesn't know if](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=426.5014444444448) [it's a platinum versus a gold card or whatever other card we](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=430.49178260869553) [have implemented there.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=434.1439565217388) [It also doesn't know how those objects were created down there.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=435.057) [So we have our AmexFactory.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=438.057) [The other thing we need to do is pass in our validator code.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=439.4458888888889) [Now I do have that copied here just to save on typing out this information.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=443.057) [I'm going to go ahead and paste it in and save this.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=450.057) [So now we have our AbstractFactory, which is our CreditCardFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=456.9141428571427) [that determines which factory should be chosen.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=462.682) [Then we defer to our AmexFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=467.057) [and we are calling the no‑args constructor here.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=469.199857142857) [If we wanted to do the template method pattern,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=472.057) [we could do a template method,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=474.7570000000001) [a factory template method that would go ahead and call](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=476.5570000000002) [the appropriate instance down below.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=479.80700000000013) [We can also choose some things based off a constructor.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=481.55700000000024) [Go ahead and look at the factory pattern though.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=485.057) [We don't need to belabor it in this example.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=487.55699999999996) [And then from here, we can go down and do some more](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=490.057) [sophisticated things in our credit cards down below.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=493.65700000000015) [Right now I just have the default empty class with some fields](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=496.057) [that are provided to us by way of our CreditCard.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=500.96609090909124) [I have a protected int cardNumberLength and a cscNumber,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=505.057) [the security code number that just goes ahead and](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=508.55700000000024) [returns that information for us.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=512.057) [Now we can go ahead and open up our demo,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=513.8991052631578) [so we have our AbstractFactoryDemo,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=517.5833157894734) [and inside of here you can see a couple of things.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=519.3903333333334) [I have gone ahead and created an instance of the CreditCardFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=523.057) [and like our previous example,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=527.182) [this is named abstract just to show you which that factory is.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=529.057) [The only thing it knows and passes into it is saying hey,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=532.329727272727) [I want to get a CreditCardFactory, and my credit score is 775.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=535.9141428571429) [That's all we know is we're just going to pass in that my credit score is 775,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=541.057) [and this is great.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=543.4855714285709) [I'll get you the right one based off of your credit score.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=544.057) [Then it says okay, now get me the credit card,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=548.4206363636365) [and I think I want a platinum credit card.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=552.057) [So it will come back and say great,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=554.0135217391309) [I'm going to go ahead and give you a platinum credit card based](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=555.7526521739138) [off that factory. So no knowledge of how that was created or what](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=559.0570000000004) [was done is passed back to the client.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=562.7712857142867) [The same thing,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=564.7236666666663) [I go ahead and get another instance of the factory down here](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=565.4379523809519) [based off of 600 and get the alternate type of credit card](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=568.457) [of a gold and pass that back.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=573.2569999999997) [So let's go ahead and run this,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=576.057) [and you'll see that we get, for our first example,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=579.3727894736845) [an AmexPlatinumCreditCard back,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=584.1096315789481) [and for our second example, the VisaGoldCreditCard back.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=585.557) [So our American Express Platinum gets returned for this set of code here,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=589.557) [and the Visa Gold gets returned for this set of code here,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=595.2388181818183) [which is exactly what we'd expect to do because as we](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=599.6024545454547) [go through our AbstractFactory, which is our CreditCardFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=604.3236666666669) [it just determines which factory to use.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=608.057) [Again,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=609.7040588235295) [it's not telling us how to create the card, it's not telling us what](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=609.9393529411766) [should be returned based off that, it just says hey,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=613.2474761904759) [this is the factory you need based off the parameters you gave me.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=615.6284285714278) [And there are no variables stored here, there are no values stored here,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=620.0570000000001) [it's just a collection of multiple factories here.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=625.1998571428574) [Notice our AmexFactory knows nothing of our VisaFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=628.057) [but they both tie in the common interface of our CreditCardFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=630.723666666667) [and then we have a CreditCard and a Validator.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=634.457) [Again,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=638.057) [that common interface that's going to get passed back](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=638.3236666666667) [throughout all of our factories.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=640.7236666666665) [Our CreditCard is actually an abstract class,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=642.057) [not an interface,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=644.8569999999999) [but just to mix it up I created a Validator interface](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=646.0569999999998) [that is, in fact, an interface.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=650.478052631579) [So to show you how both of those tie together from our CreditCardFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=653.0043684210527) [we have our common interface of the CreditCardFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=658.432) [as well as our common interface of our CreditCard and](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=661.432) [our Validator that gets returned.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=665.292294117647) [Then we go into our ConcreteFactory,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=667.3511176470586) [and from the ConcreteFactory, we actually go ahead and create](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=669.8217058823525) [our instance and handle our validation.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=674.2877692307692) [Now I've left the implementation of the validators and](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=677.057) [even the credit cards pretty simple,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=679.5570000000005) [but it's because it has nothing really to do with the relevancy of the pattern.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=681.2236666666674) [This is what we're trying to do with the pattern,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=684.9141428571422) [showing that the creation logic is not known by the client, in fact,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=687.4736666666666) [it's not even known by the AbstractFactory, it just knows to defer that to](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=692.7938421052631) [the right factory and then it handles it from there.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=697.951736842105) [So our individual factories know about their creation logic,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=701.682) [and they've got grouping of similar features,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=705.057) [so they both have validators, they both have credit cards.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=707.5275882352943) [Another great example where I saw this used and where I've used it](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=711.057) [in the past is for different database types.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=714.1522380952374) [Most people nowadays are using Hibernate or some](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=716.057) [ORM mapping tool like Hibernate,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=719.0043684210525) [but I've done the same thing where I've had a database and I know I've had](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=720.8464736842103) [the same set of queries I want to run against tables,](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=725.9141428571434) [but I didn't know if I was running it against Oracle or SQL Server or MySQL.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=729.057) [Same exact example.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=733.057) [I'm going to have the same queries return.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=734.4206363636365) [I just need to know what type of database I'm going against.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=738.057) [I'm going to have the same type of credit card returned, I](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=740.6656956521745) [just need you to get the right factory back for me and give](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=743.3201578947369) [me those queries from there.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=746.7412105263163) [So this is a little example of how we want to use that factory and how it ties everything together.](https://app.pluralsight.com/course-player?clipId=dab76e6f-3570-4023-a85e-cb01ccd5ea7c&startTime=748.057)

### [Pitfalls](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745)

[Some of the pitfalls with the AbstractFactory are its complexity.](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=1.365) [The AbstractFactory is the most complex of the creational design](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=7.365) [patterns and is definitely more difficult to implement.](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=11.253888888888893) [At some point in development, there is a runtime switch.](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=14.365) [The client has some influence what we do with the switch,](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=17.365000000000006) [and this makes some people afraid that the client](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=20.69833333333333) [knows too much about what is going on.](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=23.69833333333332) [It isn't a big issue, it can just be a hang up for some people architecturally.](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=26.365) [Although other patterns sometimes do this,](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=31.365) [it is definitely a pattern that contains other patterns.](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=34.165000000000006) [It's also very problem specific.](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=38.365) [Other patterns solve a broader problem,](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=40.42382352941178) [say for instance like the singleton, it just limits it to a single instance](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=42.89441176470591) [where the AbstractFactory is a grouping of factories.](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=48.865) [Another pitfall is that it usually starts off as a factory and](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=52.365) [then is refactored to an AbstractFactory. Most people don't often visualize using the AbstractFactory to begin with.](https://app.pluralsight.com/course-player?clipId=b6e43176-77ba-4073-b454-d7235e092745&startTime=57.03166666666663)

### [Contrast to Other Patterns](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7)

[Rather than contrast the AbstractFactory with](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=null) [another type of creational pattern,](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=2.722125) [I'm going to compare it with the factory pattern.](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=4.9905625) [A factory returns various instances and allows multiple constructors.](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=10.028230769230769) [It is interface‑driven and it is adaptable to each environment more easily.](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=16.259) [All of these hold true for the AbstractFactory as well.](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=23.259) [It is implemented with a factory, it hides the underlying ConcreteFactory,](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=27.99584210526315) [and the AbstractFactory adds one more layer of abstraction to our environment.](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=33.40185714285714) [Another key difference between these two is that the](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=40.259) [AbstractFactory is typically built through composition.](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=45.05899999999998) [So, all of these features of the factory apply to the AbstractFactory with](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=48.259) [these additional, nice added bonuses of the AbstractFactory.](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=54.03677777777776) [It hides what factory we're using.](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=57.84723529411765) [It also abstracts our environment built with composition.](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=61.02370588235296) [I should note, though,](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=65.259) [it doesn't have to be implemented with the factory, it just usually is 99% of the time.](https://app.pluralsight.com/course-player?clipId=5feb28dd-c821-4ce5-a046-83b273950aa7&startTime=69.259)

### [Summary](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e)

[To recap, the AbstractFactory is a group of similar factories.](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=null) [It is quite complex a lot more so than the other creational](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=6.007999999999999) [design patterns that we've gone through.](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=13.016) [It is heavily abstracted.](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=16.016) [We utilize interfaces, subclasses, composition,](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=18.016) [and just general software contracts to develop this pattern](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=21.23028571428571) [and achieve various levels of abstraction.](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=27.016) [We typically don't do this with other design patterns.](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=30.216000000000008) [And it's typically written as a framework pattern meaning that](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=35.016) [this is part of a larger framework. Where the other patterns may](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=38.54541176470587) [solve a particular problem and do it in a generic way that we can](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=42.68266666666668) [use it anywhere in our code,](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=47.34933333333338) [the AbstractFactory is typically built as part of a](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=50.12126315789474) [larger framework inside of our code.](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=54.8581052631579) [The AbstractFactory is the last of the creational patterns that we](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=58.016) [have gone through in this Creational Design Pattern Series if](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=62.545411764705904) [you've been following these in order.](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=66.65236363636363) [I urge you to look at this in relation to the factory, even](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=69.10690909090907) [though it can be implemented with something else, this is usually written with the factory pattern in mind.](https://app.pluralsight.com/course-player?clipId=e01aa8cb-443a-49e0-be9e-74cb61c2d53e&startTime=75.016)

## [What Next?](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028)

### [What Next?](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028)

[Well thank you for completing this design patterns course](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=2.3200000000000003) [covering creational design patterns.](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=5.68) [If you're wondering what to look at next,](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=8.4) [I might recommend that you keep an eye out for the Design](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=10.81) [Patterns in Java covering structural design patterns, as well as](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=14.59) [the course covering behavioral patterns.](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=19.51) [I'm working on both of those at the time of this being](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=21.94) [published and it'll be out shortly.](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=26.15) [There are also all of my other courses on Pluralsight, Maven Fundamentals,](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=28.64) [Spring Fundamentals, Spring MVC, Spring MVC4, Spring JPA](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=33.53) [Hibernate, and Spring Security just to mention a few.](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=38.71) [Please explore those other courses if you've enjoyed this, and](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=43.65) [keep an eye out for those two design patterns courses coming within the next couple of months. Thank you.](https://app.pluralsight.com/course-player?clipId=812b5e05-c90a-4252-81d0-087c122ee028&startTime=46.78)