Special Note about Destroy Lifecycle and Prototype Scope

Here is a subtle point you need to be aware of with "prototype" scoped beans.

For "prototype" scoped beans, Spring does not call the @PreDestroy method. Gasp!

I didn't know this either until I dug through the Spring reference manual researching a student's question.

Here is the answer from the Spring reference manual. Section 1.5.2

https://docs.spring.io/spring/docs/current/spring-framework-reference/core.html#beans-factory-scopes-prototype

---

In contrast to the other scopes, Spring does not manage the complete lifecycle of a
prototype bean*: the container instantiates, configures, and otherwise assembles a*
prototype object, and hands it to the client, with no further record of that prototype
instance.

Thus, although initialization lifecycle callback methods are called on all objects regardless of scope, ***in the case of prototypes, configured destruction lifecycle callbacks are not called****. The client code must clean up prototype-scoped objects and release expensive resources that the prototype bean(s) are holding.*

To get the Spring container to release resources held by prototype-scoped beans, try using a custom bean post-processor, which holds a reference to beans that need to be cleaned up.

This also applies to XML configuration.