

**Username:** Pralay Patoria **Book:** Under the Hood of .NET Memory Management. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC 107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

---

## Summary

From this chapter, I want you to understand the importance of object lifetime. Where and when you create an object is critical to the efficiency of your application. Adding references from one object to another, whether it's to collections, events or delegates, will likely keep your object alive for far longer than you realize. Those references will get them promoted to Gen 2 before you know it, and Gen 2 collections make your app work harder.

Don't assume that, simply because something is a local variable, it will just go out of scope. If you add a reference to it, then it could stick around for a long time, and if you do that a lot, then you could quickly end up with a memory leak or, at the very least, an application with a larger memory footprint than necessary.

I'm going to go into all of this in even more detail in the [next chapter](#), as we build on what we've learned so far.