

# Epilogue

## Closing remarks

As we reach the end of the course, I'd like to share my experience working with patterns as a veteran of the Silicon Valley. First, you'll not always find patterns implemented as the text-book versions you just read. They'll usually deviate a little here or there, but knowing the text-book version would let you immediately recognize the pattern at hand and improve your understanding of the code and your ability to navigate and make changes to it.

The frequency with which you see patterns applied will depend on what company or product you work for. If you are working in an off-shore small scale wordpress-website-building sweat shop, chances are slim you'll work on any of the patterns. The cost benefits are just not there. If you are working on a website of ten pages and your customer will pay you \$2000 then neither you nor your boss would be inclined to spend the effort and energy in going the extra mile and using nifty patterns in code. However, for a project of \$200,000 you better use design patterns where applicable or your customer will not be happy when change requirements start to cost him a fortune because you need to cut and rip apart the code and start all over again.

If you work for a company that produces a framework or library for consumption by other developers then chances are you'll be working with and applying patterns all over the place. The further up in the food chain you work and the more uncertainty and abstraction you deal with, the more likely you are to use design patterns. However, at the very least, you'll find yourself working with creational design patterns at any company as any decent sized project will have you `new()` up objects.

The foremost benefit you should have from this course is to learn the design pattern vocabulary. Being cognizant of what a *builder pattern* or a

*factory method pattern* refers to in a dev meeting exudes maturity and depth in a software engineer. Furthermore, it makes communication of complex designs amongst developers easy, leveraging the shared vocabulary.

Last but not the least, I would like to caution to not get overboard with design patterns and apply them for the sake of applying design patterns. They are a means to an end, not the end in themselves. The end being maintainable, extensible and flexible code. Keep it simple and apply patterns when and where it makes sense.

Wishing you all the best in your tech future.

**C. H. Afzal.**

**21st Sept, 2018**

*taught with passion and crafted with ♥ in Santa Clara, U.S.A.*

## Credits

Every great product is a result of team-effort and so is this course. Our team members include:

- [Ahsan Khalil](#) (Illustrations and graphic design)
- Educative's Proofreading Ninjas