**Why is this week’s topic important for teamwork?**

It is important to understand when and when you don’t need teamwork (it’s more often that you *do* than *don’t*). If you expect to be handling projects as a lone wolf all the time, you will find your employment to be very short. Projects are almost always too broad in scope for a single individual to handle alone and would take that individual far too long time to accomplish. The purpose of teams is to delegate tasks to specialized individuals so that it gets done significantly quicker.

**How do you plan on contributing to the team, besides completing your tasks?**

Contributing to the team is much more than just completing your tasks. Because teams are collaborations, it’s important to share knowledge and assistance. Coordinate between members of the team to ensure that your work will be able to integrate well with others, and potential solutions if not (or bring it to the team leader if the project needs more drafting).

**Tell me about what you taught someone? And what did they learn, that they didn’t know before.**

Programming Lingo: I taught my boss how to use singleton instances of classes to make them function as both static *and* instance objects. It was complicated, but revolutionary because we can now use entire classes as identifiers as opposed to strings or enums (though they are technically classes but you know what I mean).

Non-Programming Lingo: When someone asks who you are, imagine if you could telepathically send your entire brain instead of just telling them your name. I taught my boss how to do that, except in computer programming and not humans.

**If you were to write your experience as a STAR story, how would you phrase it?**

We had a situation where we needed a way to identify an object during both compile time and runtime, but that identifier needed a lot of information and had to be accessible by reference. My job was to come up with and architect the scheme which would allow this, and finally implement it throughout the project. I first came up with a way which would allow an identifier to exist at all times and still be referenced as though it’s specific to that object. That was relatively easy to figure out, but architecting it such that it would be scalable and flexible was challenging as I had to rework the existing system to fit with it. I slowly adapted the systems to work together and once they had been tied together, I began cleaning them up so that all the work which the objects were doing individually was instead handled in the backend by shared routines through the identifier. In the end, it brought all the objects together into a strict schema which enforced necessary procedure and performed most of the work under the hood. This was beneficial because, as a programmer, we need maintenance to be as low as possible to maximize productivity and minimize errors. It also increased performance substantially.

**If this was a religion class, how would you relate this week’s topic to the gospel?**

There is teamwork between you and God. He has placed you here to further your experiences as a mortal, to learn and to overcome, and to grow as a spiritual being. This was an agreement made between you and Him in the premortal existence. Therefore, while you are here fulfilling your end of the agreement (learn of God and serve him) , communicate with Him in exchange that He will also fulfill his end of the agreement (blessings).

**Score: 100/100**

I chose 5 20-point assignments and answered them thoughtfully, deeply, and completely.