**97 THINGS EVERY PROGRAMMER SHOULD KNOW- CHAPTER 1: ACT WITH PRUDENCE**

**WHAT ARE THREE THINGS I LEARNED TODAY**

1. **Before**- I unintentionally act without considering other factors and end up having a technical debt without my knowledge

**After**- I should escalate first the situation before acting especially when it involves critical part of the project

1. **Before**- The faster I finished my module in software the better

**After**- The codes in my module should be reviewed thoroughly to avoid committing technical debt in the future.

1. **Before**- When I promised to fix the code later, I end up not fixing it at all

**After**- I should keep track of the unfinished and unstable code that I have develop.

**97 THINGS EVERY PROGRAMMER SHOULD KNOW- CHAPTER 2: APPLY FUNCTIONAL PROGRAMMING PRINCIPLES**

**WHAT ARE THREE THINGS I LEARNED TODAY**

1. **Before**- I didn’t follow any paradigm in coding or development
2. **After**- I will start to practice functional programming paradigm in every development that I will be involved with
3. **Before**- I only create functions when necessary
4. **After**- I should create functions in every process that I will work on
5. **Before**- Creating user-defined functions are time consuming
6. **After**- Creating user-defined functions are time consuming but at the same time it saves a lot of time in the future.

**97 THINGS EVERY PROGRAMMER SHOULD KNOW- CHAPTER 3: ASK "WHAT WOULD THE USER DO?" (YOU ARE NOT THE USER)**

**WHAT ARE THREE THINGS I LEARNED TODAY**

1. **Before**- I hastily decide what users should do
2. **After**- I should involved users especially when developing a feature that is intended for them.
3. **Before**- I consider myself as a user.
4. **After**- Being a developer has a big gap between being a user
5. **Before**- I dictate the user what he/she should do
6. **After**- I should watch him/her how he/she complete different tasks.