Tamayo, Jericho Kyle M.

**97 things every programmer should know**

▼ Title Of Book and Chapter

Act with Prudence (Chapter 1)

▼ What are Three Things I learned today

1. Before – knowing its definition is important to understand why one wants to study this concept. Knowing the purpose will help to focus the reading and retain the information learned.

After - Prudence is the attribute of being cautious, smart, and discrete when making decisions and taking acts is the definition of prudence.

2. Before - It may also be helpful to have a basic understanding of related concepts such as ethics, decision-making, and personal responsibility.

After - Prudence is a vital life skill that aids people in making wise choices and abstaining from negative behavior. It is a useful skill in both personal and professional situations and is frequently regarded as a sign of sound judgment.

3. Before - It is important to set realistic expectations for what can be learned from the information and to approach the topic with an open mind and a willingness to learn. This will help the reader to better understand and retain the information and to apply it in their own life.

After - Prudence is used in a variety of areas of life, such as financial planning, moral judgment, and interpersonal relationships. It comprises weighing the positives and cons of a situation as well as the long-term effects of one's decisions. It requires decision-making, critical thinking, and the analysis of information.

**▼ Title Of Book and Chapter**

Apply Functional Programming Principles (Chapter 2)

**▼ What are Three Things I learned today**

1. Before – I’ve learned before that it is better to have or to set functions on your code so that if the trouble comes out then you can easily troubleshoot it.

After – I’ve learned that it should be important to apply it.

2. Before - Recognizing the idea that functions, whatever of the context in which they are being executed, provide the same output for the same input.

After – I’ve learned to recognize functions that return other functions as outputs or take them as arguments.

3. Before – As stated on becoming a functional programming paradigm expert, the level of code you produce in different situations can be significantly improved by. Your designs will have a considerably greater level of referential transparency if you fully comprehend and use the functional paradigm.

After - Purposely using only immutable data and pure functions, functional programming is a technique for creating software. This has no adverse effects on a function. This facilitates discovering errors and makes it easier to identify the function's intended use.

**▼ Title of Book and Chapter**

**Ask "What Would the User Do?" (You Are not the User) (Chapter3)**

**▼ What are Three Things I learned today**

1. Before – As for my understanding we programmers, we don’t know what is the perspectives of the user’s so we must not assume that we got them.

After - By posing the question "What Would the User Do?" for me I can gain insight into the user's thoughts and feelings as well as empathy for their requirements and driving forces.

2. Before – Although I felt confident in the webpage I created; I can't guarantee that I took the users' suggestions into consideration.

After - Instead than concentrating solely on technology or my personal tastes, this question might direct me toward developing designs and solutions that are based around the user.

3. Before - Understanding the user's perspective and needs can help me find solutions to their problems, which can result in creating better products and services that meet the user's needs. This can also lead to increased user satisfaction and loyalty.

After - I’ve learned that asking this question can help me better understand the user's needs, wants, and preferences, which can in turn improve communication and collaboration with the user. This can lead to a more effective design process and ultimately, a better product.