**Discussion Topics: Pragmatic Programming**

In this module's discussion board assignment, answer the following questions:

1. Using the ***Pragmatic Programmer*** reading assignment, select one (1) topic and complete the following:
   * Why did you select this topic?
   * Summarize the main points (in your own words) of that topic in three or four sentences.
   * Find at least one additional resource (video, book, article, website, etc.) that supports your summary. Include a link to that resource.

I decided to focus on topic 41: Test to Code for this module. My reasoning for picking this topic is that sometimes things can appear to be running smoothly, but appearances can be deceiving. Just because something looks like it is working does not mean that it is or that it will continue to work. This is a lesson I have learned many times in my life.

In topic 41, Thomas and Hunt describe the importance of implementing test code throughout the programming practice. Of course, testing code helps to detect errors within a program, but it goes beyond that. Thomas & Hunt emphasize the importance of thinking while you are creating test code instead of honing in on discovering errors (2024/2020, p. 154). Tests are used to fix mistakes in coding (Thomas & Hunt, 2024/2020). As crucial as testing can be, avoid unnecessary test code. Test code needs to be designed to " get you closer to a solution” (Thomas & Hunt, 2024/2020, p. 156). Tests can help avoid user issues, but releasing code to users can help produce errors that are not always discovered in internal testing practices (Thomas & Hunt, 2024/2020). Thomas says that after many years of coding, testing may not be as necessary since programs should be created to be testable by then (Thomas & Hunt, 2024/2020).

While researching more about writing tests in code, I found an article on Medium by Sail Costa. Costa goes over why testing is vital, even if it may feel like a hassle. There are different ways to write tests, and people do it at various stages, but the important part is that we remember to test the code, which can provide insight to other developers (Costa, 2016).

**References**

Costa, S. (2016, April 9). *Why You Should Write Tests - Saul Costa - Medium*. Medium. https://saulcosta.com/why-you-should-write-tests-b581d8a6b6d2

Thomas, D., & Hunt, A. (2020). *The Pragmatic Programmer: your journey to mastery*. Addison-Wesley. (Original work published 2024)

***Before you submit your thread, put your name in the subject line.***

**Assignment Requirements and Grading:**

1. An initial post of approximately 250 words is due by **Thursday, 11:59 p.m., CT**.
2. For the initial post to be considered substantive, it should be at least 250 words in length and fully cover the topics being presented. Single-sentence definitions or responses will not be awarded points.
3. Submit your post by clicking on the **Assignment Link** above, then **Create Thread**. You must create a thread in order to view your peers' posts. Tip: Create your post in a Word document and then copy and paste your work into the thread.
4. A minimum of three (3) responses, **to the original threads of other students**, of 100-200 words each are due by **Sunday, 11:59 p.m., CT**.
5. To view the rubric grading criteria, click on the following link: [Discussion Board Grading Rubric](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf).

**(50 points)**

Jessica, I really enjoyed reading your post for this module! You did a very nice job explaining what topic 43 meant. I definitely am a fan of incorporating safeguards within code because you can never be too safe. Not preparing for these circumstances can create a bad reputation with clients and other users. If we allowed personal information in output, people’s private info would be easily compromised by others. As technology changes and continues to advance, we must stay up with it. This also helps against potential hackers. I enjoyed reading through the article you attached from GoLinuxCloud! It really does have a similar message to the Stay Safe Out There topic.

Jacob, you did an excellent job summarizing your selected topic for this module! I completely agree with you that strong teamwork is vital in cybersecurity. It was especially intriguing reading about your personal experience with this topic. Working with teams that are not good at communicating can be frustrating. It is nice to have a team where members can execute similar tasks. I have felt the absence of certain coworkers throughout different jobs since not everyone can jump between various roles. Sometimes, I have also been a part of teams where others are simply unwilling to do a task when someone is out, even if they are trained to do so.

Scott, you accurately summarized topic 44: Naming Things from our textbook. I like how you used the word underestimated when discussing the importance of naming things. I think this is really important! Names should be thoughtful and have a purpose. Code is usually shared with different developers, and file names that are unclear to others cause confusion. A fun file name may seem fitting at the moment, but it can easily throw off others who are not in on it. Developers having to reach out to each other just to gain clarification on a file name is a waste of time.