**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.

***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)**

After reading the assigned topics this week, Topic 32: Configuration stuck out to me. I decided to focus on this topic since we have discussed the importance of understanding how and what a program needs to do when executing, and this felt like an extension of that discussion.

When a program launches, it is expected to change based on elements like which user is utilizing the problem and for what purpose. The environment will differentiate based on the user, displaying only relevant information to that user. Elements and values within the app will change based on users and their environments. Because of this, we must properly configure the program to keep those values outside of the app (Thomas & Hunt, 2024/2020, p. 123). When configuring a program externally, individual settings are not universally defined but tailored to the user’s preferences and information (Thomas & Hunt, 2024/2020, p. 123). For example, if someone wants to change their mood on a program from light to dark mode, the default is changed for all if this information is not stored separately per user. According to Thomas and Hunt (2024), within configuration data like “credentials for external services, logging levels and destinations, port, IP address, machine, and cluster names the app uses, and environment specific validation parameters” are just a few examples that are included within configuration (p. 123). These preferences can be stored in a few different ways, ranging from static configuration to service (Thomas & Hunt, 2024/2020).

When researching configuration, I stumbled upon an article from the GeeksforGeeks website, which discusses externalized configuration through Spring Boot. Spring Boot allows programmers to change configuration settings “without changing or redeploying the application” (GeeksforGeeks, 2024). This allows for more versatility and increased flexibility when going through all the product development stages.

**References**

GeeksforGeeks. (2024, August 29). *Spring Boot Externalized Configuration*. GeeksforGeeks. https://www.geeksforgeeks.org/spring-boot-externalized-configuration/

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