Select **one** of the topics below. Then, answer the “what,” “how,” and “why” questions below about your selected topic. Provide a simple code example to further illustrate your thoughts.

* Java Enterprise Edition (JEE)
* Jakarta
* JavaBeans
* Eclipse

**“What,” “How,” and “Why” Questions**

* **What:** What are you writing about? Give the audience a brief overview of the topic by providing them with foundational information (history, background information, etc.).
* **How:** How is the information relevant? Apply personal knowledge (this can be through research or actual practiced knowledge) to build trust with the audience.
* **Why:** Justify your position and/or course of action. The audience needs proof the information you are presenting is creditable and actionable.

For my post, I will be discussing Jakarta. Before Jakarta, Java EE was a platform used to create "mission-critical business applications" (Educative, 2024). Jakarta EE was created to improve Java EE. Since Jakarta EE is under the Eclipse Foundation and not Oracle, its name has changed from Java EE since it is a registered trademark (Educative, 2024). Jakarta EE is a faster version of Java EE, but has not erased Java EE; they are just different platforms (Educative, 2024). Jakarta tackles "integration and architectural challenges faced by Enterprise Applications" (Educative, 2024). Jakarta EE also differs from the Jakarta Project (Educative, 2024). Jakarta EE has different components and technologies that can be utilized to create "cloud-native apps, microservices, and scalable web apps" (Educative, 2024). It can also work for JVM tools and help build applications for cloud and non-cloud-based environments (Educative, 2024). Jakarta EE classifications include "web tier for frontend presentation, backend business logic, and storage and access of data" (Educative, 2024). Jakarta EE capabilities include messaging service, a mail server, a Java API, JSON processing, multithreading, and much more (Educative, 2024).

This information is relevant since it is an acceptable alternative to Java EE. It can be utilized to overcome integration issues and build business applications, similar to Java EE. This is also extremely relevant when learning how to develop a JEE application.

The information provided is credible since it is derived from a resource listed by our professor. How.dev is an encyclopedia created for developers by developers, so experienced people are providing the information.

Since Jakarta is Java EE, I have included the test code I created to ensure my Eclipse download for Enterprise Java and Web Developers worked correctly.

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello World!");

}

}

**Reference**

Educative. (2024). *What is Jakarta EE?* HowDev. https://how.dev/answers/what-is-jakarta-ee

**Assignment Requirements and Grading:**

* An initial post of approximately 250 words is due by **Thursday, 11:59 p.m., CST**.
* 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.
* 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., CST**.
* This discussion board is worth **25 Points**.
* To view the rubric grading criteria, click on the following link: [Discussion Board Grading Rubric.](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf)

Hello, Megan! I think you did a fantastic job on your discussion board. You did a comprehensive job of addressing what Jakarta EE is, how it is relevant, and why we should use it. I also chose to write about the same topic, so it was intriguing reading your explanation. Something that continues to impress with the software development community is the ability to fix existing problems. If a software application or programming language is not working as well as it could, developers can come together to launch a new version. The power of open source resources really is something special.

Hey, Adrian! I really enjoyed reading your post for this module. I think you did a great job tackling the why, how, and why behind JavaBeans. After learning more about JavaBeans, I was thoroughly impressed by its capabilities. A building block on which to base code can be extremely beneficial when working on projects. Especially those that need to take on multiple elements when building a GUI program. I love how JavaBeans have survived as the JDK constantly releases new updates. It proves to be a problem function. I am glad you mentioned that a JavaBean class must have a no-argument constructor. It is a vital part to remember.

Hello there, Jacob! Your post for this module was very insightful. You did a great job covering what, how, and why of Eclipse. The code you included is a good example of how Java can be used in Eclipse. By creating a project and class, you can easily run the string of code you provided. I have had a similar experience to yours when using Eclipse. I also like using Eclipse a lot and prefer it to other IDEs like NetBeans. I find it much easier to work with. The plugins are also a great addition and make it easier to link extensions.