Discuss **two** of the following topics on Java methods for your initial post. Provide a code example, where necessary, to elaborate your thoughts.

* “Pass-by-value" vs. “Pass-by-reference”
* **Parameter Order**
* **Formal parameters vs. actual parameters**
* Compatible types

Java has many methods that can be invoked to return different results. For this post, I will be focusing on parameter order and formal parameters vs actual parameters. When we define a method, it involves assigning a parameter (Liang, 2019). When dealing with Java methods, it is vital to remember the importance of parameter order. If the parameter order is mixed up in a line of code, it will not perform as desired. This goes beyond arguments needing to be in order; according to Liang (2019), the parameters must also match the “number and compatible type.”

For example, we can add a string that prints a message “n” amount of times. This could look like the following example:

public class dbexample {

public static void main(String[] args) {

nPrintln("This is an example for our module 3 discussion board!", 4);

}

public static void nPrintln(String message, int n) {

for (int i = 0; i < n; i++) {

System.out.println(message);

}

}

}

Output:

This is an example for our module 3 discussion board!

This is an example for our module 3 discussion board!

This is an example for our module 3 discussion board!

This is an example for our module 3 discussion board!

If the four or any number is placed before the message, the program will raise an error when trying to run.

A formal parameter can also be called a parameter, which is “defined in the method signature (Liang, 2019). Liang (2019) describes a parameter as being a “placeholder” for a new value that is passed. An actual parameter is also known as an argument, so once input or data is assigned, it becomes an actual parameter (Liang, 2019). It is interesting to note that parameters are optional since a method does not require them.

**References**

Liang, Y. D. (2019). *Introduction to Java programming and data structures: comprehensive version*. Pearson. https://plus.pearson.com/home?utm\_source=ereader

**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**.
* To view the rubric grading criteria, click the following link: [Discussion Board Grading Rubric](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf).

Jessica, I think you did a fabulous job explaining pass by value vs pass by reference and formal parameters vs actual parameters. Your examples are all accurate and fitting for what we are currently learning. I am glad you explained how dealing with objects can be confusing because I felt the same way at first. Many parts of Java seem to be confusing or very similar until they are further broken down. Taking the time to understand the differences can make coding programs much easier. Your examples for formal and actual parameters also fit very well. I have utilized the references you included and think they provided insightful knowledge.

Jacob, your discussion board post was very nicely said! You are spot on with your explanations of formal parameters, actual parameters, and compatible types. The one website you included, Geeks for Geeks, always has great and helpful information to reference in Java and for other programming languages. I have never used or read from your rollbar source, so I was intrigued when I referenced your link! I like how it includes code examples and explanations to help comprehend how a program works. I also like how, similar to our text, there are incorrect examples to help explain why a solution may seem right but ultimately why it will not work.

Samir, you did a nice job on your post for this module. I also focused on parameter order and formal parameters vs. actual parameters. A method may appear to be perfect, but if the parameter is in the wrong order, it will not function the way we want it to. In addition, it must also be compatible in type and match the number. Your example was a nice way to show how parameter order works in action. When learning about formal and actual parameters, I reviewed the sections a few times to ensure I remembered the differences and did not confuse them.