Select at least **two** of the following topics on Java features for your initial. Provide a code example, where necessary, to elaborate your thoughts.

* Field Encapsulation
* Objects with Methods and Methods Receiving Instances
* A Collection of Object Instances
* The ‘this’ Reference
* Variable Scope

As we have discovered, Java offers many widely beneficial features when working with the programming language. Some of those features range from the variable scope to a collection of object instances, but this discussion post focuses on field encapsulation and the ‘this’ reference. Encapsulation in Java hides private and sensitive data. For encapsulation to work correctly, the class variables must be private. A getter and setter method is utilized to retrieve the data. Because of encapsulation, users will not see the details of the implementation.

The ‘this’ reference refers to the object. According to Narang and Kumar (2016), this instance is used to “call current class methods and fields, pass an instance of the current class as a parameter, and differentiate between the local and instance variables.”

Here is an example provided by the GeeksforGeeks website. (Narang & Kumar, 2016). The class Test are the instance variables, and the ‘this’ keyword in the test constructor assigns parameters to a and b. The object is created and called in the public static void main(String[] args) method. This example demonstrates the functionalities of the ‘this’ reference.

// Java code for using 'this' keyword to

// refer current class instance variables

**class** Test {

**int** a;

**int** b;

    // Parameterized constructor

    Test(**int** a, **int** b)

    {

**this**.a = a;

**this**.b = b;

    }

**void** display()

    {

        // Displaying value of variables a and b

        System.out.println("a = " + a + "  b = " + b);

    }

**public** **static** **void** main(String[] args)

    {

        Test object = **new** Test(10, 20);

        object.display();

    }

}

**References**

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

Narang, M., & Kumar, A. (2016, January 12). *“this” reference in Java*. GeeksforGeeks. https://www.geeksforgeeks.org/this-reference-in-java/

W3 Schools. (n.d.). *Java this Keyword*. Www.w3schools.com. Retrieved February 4, 2025, from <https://www.w3schools.com/java/ref_keyword_this.asp>

**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 on the following link: [Discussion Board Grading Rubric](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf).

Nardos, you did a great job of explaining field encapsulation and variable scope in your discussion post. Field encapsulation is very essential when dealing with users’ sensitive information. Certain information should have exclusive privileges so that private information is not placed in the wrong person’s hands. The example you included perfectly demonstrates field encapsulation in action. I am glad that you broke down the main variable scopes because they are important to remember. If you use a local variable, then it cannot be accessed outside of the method, so it is vital to pay attention to the variables is vital when coding.

Nima, I think you did an excellent job on your post for this module. I think the code example for your explanation of a collection of object instances perfectly fits with your description. I like how our discussion board posts focus on topics that are utilized later in our programming assignments. I have really enjoyed using the ‘this’ keyword thus far. Your example for the ‘this’ reference also perfectly expands on your explanation. You are spot on that using ‘this’ can make code clearer to read and helps with conflicts. This is why I have enjoyed using the ‘this’ reference in coding.

Jacob, after reading your post, I think you did a great job writing your thoughts on variable scopes and a collection of object instances in Java. You are absolutely correct that understanding variable scope is crucial when working with this programming language. I really like the flexibility that variable scopes provide since the variable required changes based on the coding circumstances. I like how you took the time to explain how your examples apply to your points. The example for your collection of object instances, I think, is a really helpful guide for this week’s programming assignment and, I imagine, many upcoming ones.