Topic: Select at least two (2) of the topics below. In your writing, be sure to explain the what, how, and why of the selected topics. If necessary, provide code examples to further illustrate your thoughts.

* Creating a Class vs. Creating an Object
* Multiple Objects Instances

The relationship between classes and objects is analogous to that between an apple-pie recipe and apple pies: you can make as many apple pies as you want from a single recipe (Liang, 2020). Object instances are unique containers that are created from a class. Using our apple pie example, you may have an apple pie recipe and use it to make 5 different apple pies. Each of those apple pies could vary slightly if you choose to increase/decrease the amount of apple in them. This would be represented by instantiating the apple pie class with different amounts of apples, or using a setting method to adjust. They are all still apple pies, but each one may store different variables and have different references. Even if you make two identical apple pies, they will still be unique objects, just like you can instantiate two objects using the default constructor then use them separately.

To summarize:

* What – Classes are created to define objects. Objects are created to use in your program.
* How – Classes will have a suite of methods, variables, and constructors that you create to provide them with useful functionality and information. Objects are instantiated to make use of these methods and variables.
* Why – Using classes and objects reduces code reuse and allows for unique ways to manipulate and store data.

Reference:

Liang, Y. D. (2020). Introduction to java programming and Data Structures. Pearson Education, Inc.

Response 1:

Christopher,

Understanding classes and objects is fundamental for object-oriented programming. Blueprints is an excellent analogy to explain the relationship between classes and objects.

Modifying attributes is something you may want to restrict. Another use of Setter methods is to verify inputs so that no bad data is passed to your class. If you use dot operators it is not possible to do that.

Anitha,

In addition to Static, methods can be declared public or private. This defines who can call the methods, which is useful to create methods that are only used internally. Protected also exists, but I do not believe we will be covering that before inheritance.