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

* Benefits of generics.
* Use and definition of generic classes and interfaces.
* Use of generic methods.
* Raw generic types for allowing backward compatibility.

In Java, there are generics that can be utilized when programming. According to GeeksforGeeks (2025), generics in Java are parameterized types that allow objects to be a “parameter to methods, classes, and interfaces.” These entities are called “generic entities” (GeeksforGeeks, 2016). There are two types of Java Generics: generic method and generic classes. A generic method uses a parameter(s) that is “cited by actual type” and returns a value (GeeksforGeeks, 2016). Generic classes are similar to non-generic classes but “contain a type parameter section” (GeeksforGeeks, 2016). An important part to remember is that an instance of a genetic type only accepts reference types (GeeksforGeeks, 2016). There are some disadvantages to using generics, like them being overcomplex for beginners, creating “overhead during runtime,” not supporting primitive types, and having “limited reflection” (GeeksforGeeks, 2016). Even though there are disadvantages, it has many advantages that can make it much more fitting to tackle different tasks in Java. One of the benefits of generics is the ability to reuse the code with any type (GeeksforGeeks, 2016). Generics also create safer code since errors are shown during compile time (GeeksforGeeks, 2016). Generics eliminates the need for type casting since the compiler casts for us (GeeksforGeeks, 2016). Lastly, generics make code easier to read and maintain while offering algorithmic implementation (GeeksforGeeks, 2016). According to Oracle, n.d., the same conventions and techniques for creating a generic class can be applied to creating genetic interfaces (GeeksforGeeks, 2016).

This is an example of using multiple type parameters (Oracle, n.d.):

public interface Pair<K, V> {

public K getKey();

public V getValue();

}

public class OrderedPair<K, V> implements Pair<K, V> {

private K key;

private V value;

public OrderedPair(K key, V value) {

this.key = key;

this.value = value;

}

public K getKey() { return key; }

public V getValue() { return value; }

}

**References**

GeeksforGeeks. (2016, March 1). *Generics in Java*. GeeksforGeeks. https://www.geeksforgeeks.org/generics-in-java/

Oracle. (n.d.). *Generic Types (The JavaTM Tutorials > Learning the Java Language > Generics (Updated))*. Docs.oracle.com. https://docs.oracle.com/javase/tutorial/java/generics/types.html

**Assignment Requirements and Grading:**

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