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

* BigInteger & BigDecimal Classes
* Calendar & GregorianCalendar Classes
* Cloneable Interface
* Rational Class

Some features discussed throughout this module included BigInteger & BigDecimal classes, Calendar & GreogrianCalendar Classes, Cloneable Interface, and Rational class. I will elaborate on the Calendar & Gregorian Calendar Classes, as well as the Cloneable Interface Java features for this discussion board.

In Java, Calendar is an abstract class that contains concrete subclasses (Liang, 2019). The Gregorian Calendar is an extension of the Calendar and allows a specific calendar system to be utilized (Liang, 2019). When called, the GregorianCalendar method receives the time from the default time zone (University of Washington, 2025). To use the calendar class, import java.util.Calendar, and use import java.util.GregorianCalendar for the Gregorian Calendar subclass (Liang, 2019). When using the Gregorian Calendar, it is important to remember that the month starts from 0, so it goes from 0-11 instead of 12. The date starts at 1, and so does the first day of the week, which is Sunday. When determining AM or P, 0 means AM and 1 means PM.

The Cloneable Interface allows an object to be cloned (Liang, 2019). A cloneable interface does not have a method, so it is empty, which is known as a “marker interface” (Pratik T, 2024). After implementing the cloning interface into your code through the class, Pratik (2024) explains that the clone() should be used to override the Object class and then called using the super.clone() method. The following is an example provided by Pratik:

class Person implements Cloneable {  
 String name;  
 int age;  
 Person(String name, int age) {  
 this.name = name;  
 this.age = age;  
 }  
 @Override  
 protected Object clone() throws CloneNotSupportedException {  
 return super.clone();  
 }  
 @Override  
 public String toString() {  
 return "Person{name='" + name + "', age=" + age + "}";  
 }  
 public static void main(String[] args) {  
 try {  
 Person original = new Person("Alice", 30);  
 Person cloned = (Person) original.clone();  
   
 System.out.println("Original: " + original);  
 System.out.println("Cloned: " + cloned);  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
 }  
}

**References**

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

Pratik T. (2024, July 11). *Understanding the Cloneable Interface, Shallow Copy, and Deep Copy in Java*. Medium. https://medium.com/@pratik.941/understanding-the-cloneable-interface-shallow-copy-and-deep-copy-in-java-73c45066ecb1

University of Washington. (2025). *Class java.util.GregorianCalendar*. Washington.edu. https://courses.cs.washington.edu/courses/cse341/98au/java/jdk1.2beta4/docs/api/java/util/GregorianCalendar.html

**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).

Colton, you are spot on when you say that Java has many built-in features that help developers! I am often shocked at how versatile the programming language truly is. You did a great job on your post and explaining the Cloneable interface and the BigInteger and BigDecimal classes. After reviewing our sources for the week, it took me a few times to fully comprehend how creating a copy of an object worked. You perfectly summed up the process and included a fitting example. I am glad you added that mathematical methods must consist of the proper method to be properly handled.

Jacob, I think your discussion post and further thoughts on the Cloneable Interface and the Rational Class were relevant and insightful. I am glad you mentioned how the Cloneable interface must be implemented, or it will run into an error. This is important to remember and will help avoid an easy mistake. I like that you also mentioned how cloning may not be the best method to use. It can become easily complicated and tedious to use the clone method. I looked into the case study you included on the DEV Community website and was very fascinated with it. I will be checking our articles later!

Miles, your discussion board for this week was great and thorough. There are many scenarios where utilizing a cloneable interface is practical when programming with Java. I think how you compared cloning to using a Word document. It perfectly explains how the deep copy works when cloning. It even helped me better understand the concept. Other calendars can be implemented that could be more fitting for what we are coding for. I think it is important to mention that when using the calendar, months start at 0 while days start at 1, and the first day of the week is Sunday.