1. It does not need a constructor because it is already being incremented by one and helps be initialized to 0 already
2. It does not need a main because it is called from Runner.java and Runner.java has a main to do it for the Counter class
3. We know that ModNCounter inherits from Counter because of the word “extends”
4. The method that is the constructor in ModNCounter class is public ModNCounter (int n)
5. cycleLength is not visible to the parent Counter class because cycleLength is private.
6. When we call value and reset methods for ModNCounter, the reset method would be called from the extends Counter
7. When we call an increment and reset methods for ModNCounter2, it will get it from Counter
8. Decrement does not exist in Counter
9. The behavior for DecrementableCounter does change.
10. toString is being inherited from ModNCounter which helps output spring, summer, fall winter.