# Ring Data Structure

```
public interface RingInterface<T> {
   void advance();
   T getCurrent();
   void add(T

);
   T remove();
}


public class Ring<T> implements RingInterface<T> {
   private T[] items; // Ring ထဲက items များကို သိုလှောင်မယ့် Array
   private int current = 0; // လက်ရှိနေရာကို ပြထားမယ့် Index
   private int size; // Ring ထဲမှာ လက်ရှိပါရှိတဲ့ Item အရေအတွက်

   // Constructor
   public Ring(int capacity) {
      items = (T[]) new Object[capacity]; // Array ကို သတ်မှတ်ထားတဲ့ capacity နဲ့ ဖန်တီးမယ်
      size = 0;
   }

   @Override
   public void advance() {
      current = (current + 1) % items.length; // နောက်တစ်ခုသို့ ရွှေ့သွားမယ်
   }
```

```
    @Override
    public T getCurrent() {
        return items[current]; // လက်ရှိ Item ကို ပြန်ထုတ်မယ်
    }

    @Override
    public void add(T newEntry) {
        items[current] = newEntry; // လက်ရှိနေရာမှာ Item အသစ်ထည့်မယ်
        if (size < items.length) size++;
    }
    @Override
    public T remove() {
        T removedItem = items[current]; // လက်ရှိ Item ကို ဖယ်ရှားမယ်
        items[current] = null; // နေရာကို null ဖြင့် ဖြည့်သွားမယ်
        advance(); // နောက်တစ်ခုသို့ ရွှေ့မယ်
        return removedItem;
    }
}

public class Main {
    public static void main(String[] args) {
        // Create a Ring of String type with a capacity of 5
        Ring<String> ring = new Ring<>(5);

        // Add items to the Ring
        ring.add("Apple");
        ring.add("Banana");
        ring.add("Cherry");
        ring.add("Date");
        ring.add("Elderberry");

        // Display the current item in the ring
        System.out.println("Current item: " + ring.getCurrent());  // Outputs: Apple

        // Advance the pointer and display the new current item
        ring.advance();
        System.out.println("Current item after advancing: " + ring.getCurrent());  // Outputs: Banana

        // Add a new item, which should overwrite the 'Apple' if size is limited to 5
        ring.add("Fig");
        System.out.println("Item after adding Fig and overwriting Apple: " + ring.getCurrent());  // Outputs: Fig
        // Remove the current item (Fig) and check the next item
        String removedItem = ring.remove();
        System.out.println("Removed item: " + removedItem);  // Outputs: Fig
        System.out.println("New current item after removing Fig: " + ring.getCurrent());  // Outputs: Banana

        // Showcasing how the ring wraps around
        ring.advance();  // Moves to Cherry
        ring.advance();  // Moves to Date
        ring.advance();  // Moves to Elderberry
        ring.advance();  // Should wrap around to Banana
        System.out.println("Wrapped around to: " + ring.getCurrent());  // Outputs: Banana
    }
}
```