File - C:\Users\niall\IdeaProjects\B00787301_COM498_Assignment2\src\AListArray.java

```java
import java.util.Arrays;

public class AListArray<T> implements ListInterface<T>
{
    private T[] list;
    private int numberOfEntries;
    private int capacity;
    private final int DEFAULT_SIZE = 9;

    private void addCapacity()
    {
        capacity += DEFAULT_SIZE;
        list = Arrays.copyOf(list, capacity + 1);
    }

    public AListArray()
    {
        T[] tempList = (T[])new Object[DEFAULT_SIZE +1];
        numberOfEntries=0;
        list = tempList;
        capacity = DEFAULT_SIZE;
    }

    public void add(T newEntry)
    {
        if (numberOfEntries == capacity)
            {
            addCapacity();
            }
        numberOfEntries++;
        list[numberOfEntries] = newEntry;
    }

    public void add(int newPosition, T newEntry)
    {
        if(newPosition >=1 && newPosition <= numberOfEntries + 1)
            {
            if(numberOfEntries == capacity)
                {
                addCapacity();
                }
            for (int i = numberOfEntries; i >= newPosition; i--) {
                    list[i + 1] = list[i];
                }
                list[newPosition] = newEntry;
                numberOfEntries++;
            }
        else {
            throw new IndexOutOfBoundsException("New entry position is out of bounds");
            }
        }

    public T remove(int position)
    {
        if(position >=1 && position <= numberOfEntries)
            {
            T valueToReturn = list[position];
            for(int i = position; i < numberOfEntries; i++) {
                list[i] = list[i + 1];
            }
            numberOfEntries--;
            return valueToReturn;
            }
        else throw new IndexOutOfBoundsException("Remove position is out of bounds: " + position);
    }

    public void clear()
    {
        numberOfEntries = 0;
    }
    public T replace(int position, T newEntry)
    {
        if(position >= 1 && position <= numberOfEntries) {
            T valueToReturn = list[position];
            list[position] = newEntry;
            return valueToReturn;
```

```
 77         }
 78         else throw new IndexOutOfBoundsException("Replace position is out of bounds");
 79     }
 80     public T getEntry(int position)
 81     {
 82         if(position >=1 && position <= numberOfEntries) {
 83             return list[position];
 84         }
 85         else throw new IndexOutOfBoundsException("Get entry position is out of bounds");
 86     }
 87     public T[] toArray()
 88     {
 89         T[] arr = (T[]) new Object[numberOfEntries];
 90         System.arraycopy(list, 1, arr, 0, numberOfEntries);
 91         return arr;
 92     }
 93     public boolean contains(T anEntry)
 94     {
 95         boolean found = false;
 96         int i = 1;
 97         while (i <= numberOfEntries && !found)
 98             if(list[i++].equals(anEntry))
 99                 found = true;
100         return found;
101     }
102     public int getLength()
103     {
104         return numberOfEntries;
105     }
106     public boolean isEmpty()
107     {
108         return (numberOfEntries == 0);
109     }
110
111     public String toString()
112     {
113         String strResult = "[ ";
114         for(int i = 1; i <= numberOfEntries; i++)
115             strResult += list[i] + ". ";
116         strResult += "]";
117         return strResult;
118     }
119 }
120
```