CS0455 – ALGORITHMS AND INFORMATION STRUCTURES

AList Implementation

Task

AList class is an implementation of a list using an array. Pseudocode methods below are written without checks on exceptions and without checks on the need to double and without checks on a maximum size for the queue.
Find and correct errors and complete the last two methods.

Instance Variables (private)

```
T[] list       // Array of list entries; ignore list[0]
int numberOfEntries     // Store entries from list[1] to list[numberOfEntries]
```

Constants (private static final)

```
int DEFAULT_CAPACITY = 25
```

Constructors

```
public AList() {
      this(DEFAULT_CAPACITY)
}

public AList(int initialCapacity) {
      list = new T[initialCapacity]
      numberOFEntries = 0
}

public void add(int givenPosition, T newEntry) {
      list[numberOfEntries] = newEntry
      numberOfEntries++
}

public void add(int givenPosition, T newEntry) {
      makeRoom(givenPosition)
      list[givenPosition] = newEntry
      numberOfEntries++
}

public T remove(int givenPosition) {
      temp = list[givenPosition]
      removeGap(givenPosition)
      list[numberOfEntries] = null
      return temp
}

public void clear()  {
      for i = 1 to numberOfEntries
            list[i] = null
      numberOfEntries = 0
}
```

```
public T replace(int givenPosition, T newEntry) {
      temp = list[givenPosition]
      list[givenPosition] = newEntry
      return temp
}
public T getEntry(int givenPosition)  {
      return list[givenPosition]
}
public T[] toArray()  {
     for i = 0 to numberOfEntries - 1
         result[index - 1] = list[index]
     return result;
}
public boolean contains(T anEntry) {
      for i = 1 to numberOfEntries
            if list[i] == anEntry
                  return true
                  break
      return false
}
public int getLength()  {
      return list.length
}
private void ensureCapacity()  {
      if (numberOfEntries == list.length – 1 {
            int newCapacity = 2 * numberOfEntries
            newList = new T[newCapacity]
            for i = 1 to numberOfEntries
                  newList[i] = list[i]
      }
}
private void makeRoom(int givenPosition)  {









}

private void removeGap(int givenPosition)  {









}
```