```
::::::::::::::
ListArrayBased.java
::::::::::::::
// ********************************************************
// Array-based implementation of the ADT list.
// *********************************************************
public class ListArrayBased implements ListInterface
{

    protected int MAX_LIST = 3; //fixes programming style
    protected Object []items;  // an array of list items
    protected int numItems;  // number of items in list

    public ListArrayBased()
    {
        items = new Object[MAX_LIST];
        numItems = 0;
    }  // end default constructor

    public boolean isEmpty()
    {
        return (numItems == 0);
    } // end isEmpty

    public int size()
    {
        return numItems;
    }  // end size

    public void removeAll()
    {
        // Creates a new array; marks old array for
        // garbage collection.
        items = new Object[MAX_LIST];
        numItems = 0;
    } // end removeAll

    public void add(int index, Object item)
    throws  ListIndexOutOfBoundsException
    {
        /***
        if (index >= MAX_LIST)
        {
            throw new ListException("ListException on add");
        }  // end if

        ***/
        if (index >= 0 && index <= numItems && items[index] != null) //fixes imple
mentation error
        {
            // make room for new element by shifting all items at
            // positions >= index toward the end of the
            // list (no shift if index == numItems+1)
            for (int pos = numItems-1; pos >= index; pos--)  //textbook code modif
ied to eliminate logic error causing ArrayIndexOutOfBoundsException
            {
                items[pos+1] = items[pos];
            } // end for
            // insert new item
            items[index] = item;
            numItems++;
        }
```

```
        else if(index >= 0 && index < MAX_LIST && items[index] == null) //fixes im
plementation error
        {
            items[index] = item;
            numItems++;
        }

        else
        {
            // index out of range
            throw new ListIndexOutOfBoundsException(
                "ListIndexOutOfBoundsException on add");
        }  // end if
    } //end add

    public Object get(int index)
    throws ListIndexOutOfBoundsException
    {
        if (index >= 0 && index < numItems)
        {
            return items[index];
        }
        else
        {
            // index out of range
            throw new ListIndexOutOfBoundsException(
                "ListIndexOutOfBoundsException on get");
        }  // end if
    } // end get

    public void remove(int index)
    throws ListIndexOutOfBoundsException
    {
        if (index >= 0 && index < numItems)
        {
            // delete item by shifting all items at
            // positions > index toward the beginning of the list
            // (no shift if index == size)
            for (int pos = index+1; pos < numItems; pos++) //textbook code modifie
d to eliminate logic error causing ArrayIndexOutOfBoundsException

            {
                items[pos-1] = items[pos];
            }  // end for
            numItems--;

            items[numItems] = null; //fixes memory leak
        }
        else
        {
            // index out of range
            throw new ListIndexOutOfBoundsException(
                "ListIndexOutOfBoundsException on remove");
        }  // end if
    } //end remove
}
::::::::::::::
ListArrayBasedPlus.java
::::::::::::::

public class ListArrayBasedPlus extends ListArrayBased {
    public ListArrayBasedPlus()
    {
        super();
    }  // end default constructor

    public void addResize(int index, Object item)
    {
        if(index > MAX_LIST || numItems+1 > MAX_LIST)
        {
            MAX_LIST += 1;
            resize();
            add(index, item);
        }
        else
        {
            add(index, item);
        }
    }

    public void resize()
    {
        Object []bufferList = makeBuffer();
        items = new Object[MAX_LIST];
        for(int index = 0; index < numItems; index++)
        {
            items[index] = bufferList[index];
        }
    }

    public void reverse()
    {
        Object []bufferList = makeBuffer();
        items = new Object[MAX_LIST];
        for(int index = 0, lastIndex = numItems-1; index < numItems; index++, last
Index--)
        {
            items[index] = bufferList[lastIndex];
        }
    }

    public Object[] makeBuffer()
    {
        Object []bufferList = new Object[numItems];
        for(int index = 0; index < numItems; index++)
        {
            bufferList[index] = items[index];
        }
        return bufferList;
    }


    public String toString()
    {
        StringBuilder info = new StringBuilder();
        for(int index = 0; index < numItems; index++)
        {
            info.append(get(index));
            if(index < numItems - 1)
            {
                info.append(", ");
            }
            else
            {
                info.append(".");
            }
        }
        return info.toString();
    }
}
::::::::::::::
ListArrayListBased.java
::::::::::::::
import java.util.ArrayList;

public class ListArrayListBased implements ListInterface {

    ArrayList<Object> items;

    public ListArrayListBased(int size)
    {
        items = new ArrayList<Object>(size);
    }

    public boolean isEmpty() {
        if(items.size() == 0)
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public int size() {
        return items.size();
    }

    public void add(int index, Object item) throws ListIndexOutOfBoundsException {
        if(index >= items.size() && index < -1)
        {
            if(items.get(index) != null)
            {
                for(int pos = items.size() +1; pos >= index; pos--)
                {
                    items.add(pos, items.get(pos-1));
                }
                items.add(index, item);
                //numItems ++;
            }
        }
        else
        {
            throw  new ListIndexOutOfBoundsException("Index out of bounds.");
        }

    }

    public Object get(int index) throws ListIndexOutOfBoundsException {
        if(index > items.size() && index < -1)
        {
            return items.get(index);
        }
        else
        {
            throw new ListIndexOutOfBoundsException("Index out of bounds");
        }
    }

    public void remove(int index) throws ListIndexOutOfBoundsException {
        if(index > items.size() && index < -1)
        {
            int size = items.size();
            for(int pos = index; pos < size; pos++)
            {
                items.add(pos, items.get(pos+1));
            }
            //numItems--;
        }
        else
        {
            throw new ListIndexOutOfBoundsException("Index out of bounds");
        }

    }

    public void removeAll() {
        items = new ArrayList<Object>();
    }
}
::::::::::::::
ListArrayListBasedPlus.java
::::::::::::::
import java.util.ArrayList;

public class ListArrayListBasedPlus extends ListArrayListBased
{

    public ListArrayListBasedPlus(int size)
    {
        super(size);
    }

    public void reverse()
    {
        ArrayList<Object> bufferList = makeBuffer();
        for(int index = 0, lastIndex = size()-1; index < size(); index++, lastInde
x--)
        {
            items.add(bufferList.get(lastIndex));
        }
    }

    private ArrayList<Object> makeBuffer()
    {
        ArrayList<Object> bufferList = new ArrayList<Object>();
        int size = items.size();
        for(int index = 0; index < size; index++)
        {
            bufferList.add(items.get(index));
        }
        return bufferList;
    }
}
```