Antonio Rosado; 1

```
1
```

```
..............
Lab2Status.txt
Problem 1: compiles, runs correctly on all provided input
Lab2Conclusions.txt
I found Lab 2 very informative as the method of creating another class for a full
Array/ArrayList is completely new to me and I found it to be a very useful impleme
ntation that I will use in the future. ::::::::::
ListInterface.java
// ***************
// Interface ListInterface for the ADT list.
// *****************
public interface ListInterface
   boolean isEmpty();
   int size();
   void add(int index, Object item) throws ListIndexOutOfBoundsException;
   Object get(int index) throws ListIndexOutOfBoundsException;
   Object remove(int index) throws ListIndexOutOfBoundsException;
   void removeAll();
   String toString();
} // end ListInterface
/** Source code for ArrayList
 public boolean isEmpty()
 return size == 0;
public int size()
 return size;
public void add(int index, E e)
   checkBoundInclusive(index);
   modCount++;
   if (size == data.length)
    ensureCapacity(size + 1);
   if (index != size)
    System.arraycopy(data, index, data, index + 1, size - index);
   data[index] = e;
   sizett:
public E get(int index)
 checkBoundExclusive(index);
 return data[index];
public E remove (int index)
```

```
checkBoundExclusive(index);
 E r = data[index];
 modCount++;
 if (index != --size)
   System.arraycopy(data, index + 1, data, index, size - index);
 // Aid for garbage collection by releasing this pointer.
 data[size] = null;
 return r;
 */
ListIndexOutOfBoundsException.java
public class ListIndexOutOfBoundsException
   extends IndexOutOfBoundsException
   public ListIndexOutOfBoundsException(String s)
       super(s);
   } // end constructor
ListException.java
* Purpose: Data Structure and Algorithms Review Programming Assignment
 * Status: Complete and thoroughly tested
 * Last update: 1/30/23
 * Submitted: 1/30/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.01.30
public class ListException
   extends RuntimeException
   public ListException(String s)
       super(s);
     // end constructor
} // end ListIndexOutOfBoundsException
......
ListArrayBased.java
..............
// ****************
// Array-based implementation of the ADT list.
// *********************
public class ListArrayBased implements ListInterface
   private static final int MAX_LIST = 3;
   protected static 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) // fixes implementation/programming st
yle errors
    throws ListIndexOutOfBoundsException
        if (numItems == items.length)
            throw new ListException("ListException on add");
        if (index >= 0 && index <= numItems)</pre>
            // 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
            // 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)</pre>
            return items[index];
        else
            // index out of range
            throw new ListIndexOutOfBoundsException(
```

```
"ListIndexOutOfBoundsException on get");
        } // end if
   } // end get
   public Object remove(int index)
   throws ListIndexOutOfBoundsException
       Object result;
        if (index >= 0 && index < numItems)</pre>
            // delete item by shifting all items at
           // positions > index toward the beginning of the list
           // (no shift if index == size)
           result = items[index];
                    if(numItems == items.length)
               throw new ListException("ListException on remove");
           for (int pos = index+1; pos < numItems; pos++) //textbook code modifie</pre>
d to eliminate logic error causing ArrayIndexOutOfBoundsException
               items[pos-1] = items[pos];
           } // end for
           items[--numItems] = null; // fixes memory leak
        else
            // index out of range
           throw new ListIndexOutOfBoundsException(
                "ListIndexOutOfBoundsException on remove");
        } // end if
        return result;
    } //end remove
ListArrayBasedPlus.java
* Purpose: Data Structure and Algorithms Review Programming Assignment
 * Status: Complete and thoroughly tested
 * Last update: 02/02/23
 * Submitted: 02/02/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.02.02
public class ListArrayBasedPlus extends ListArrayBased
    * Constructor.
   public ListArrayBasedPlus() {
        super();
    * Adds items to Array
   public void add(int index, Object item)
        if(items.length == numItems)
```

```
resize(); // if items reaches max size/num of items, resize array
        super.add(index, item); // call superclass
     * Reverses Array.
   public void reverse()
        for (int index = 0; index < numItems / 2; index++)</pre>
            Object temps = items[index];
            items[index] = items[numItems - index - 1];
            items[numItems - index - 1] = temps;
     * Resizes array if size cap is reached.
   public void resize()
        int resize = items.length;
        Object[] new_list = new Object[items.length * 2]; // new Array == oldArray
        for(int index = 0; index < numItems; index++)</pre>
           new list[index] = items[index];
        items = new_list;
     * Returns a string value of item(s) in Array
   public String toString()
        StringBuilder sb = new StringBuilder();
        for (int index = 0; index < numItems; ++index) // pre-increment</pre>
            sb.append(items[index] + " ");
        return sb.toString();
     * Returns length of Array
     * @return
                     int length
   public int length()
        return items.length;
}:::::::::::
Lab2P1Driver.java
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Lab2P1Driver extends ListArrayBasedPlus
```

```
static BufferedReader stdin = new BufferedReader(new InputStreamReader(System.
in));
   public static void main (String[] args) throws IOException
    ListArrayBasedPlus list_plus = new ListArrayBasedPlus();
   boolean exit = false;
         int pos = -1;
          while (!exit) {
            System.out.println("Select from the following menu: \n"
                             + "0. Exit the program \n"
                             + "1. Insert item into the list \n"
                             + "2. Remove item from the list \n"
                             + "3. Get item from the list \n"
                             + "4. Clear the list \n"
                             + "5. Print size and content of the list \n"
                             + "6. Reverse the list \n ");
            System.out.print("Make your menu selection now: ");
            int input = Integer.parseInt(stdin.readLine());
            System.out.println(input);
            // possible cases for initial input
            switch (input) {
            case 0:
                System.out.println("Exiting program... good bye");
                exit = true;
                break;
                System.out.println("You are now inserting an item into the list.")
                System.out.print("Enter item: ");
                Object item = stdin.readLine();
                System.out.println(item);
                            System.out.print("Enter the position to enter the item
in: ");
                pos = Integer.parseInt(stdin.readLine());
                System.out.println(pos);
                if (pos <= list plus.size())</pre>
                 list_plus.add(pos, item);
                  System.out.println("Item " + item + " inserted in position " + p
os + " in the list.");
                else
                  System.out.println("Position specified is out of range!");
                break;
            case 2:
                        System.out.println("You are now removing an item from the
list.");
                        System.out.print("Enter position to remove item from: ");
                pos = Integer.parseInt(stdin.readLine());
                System.out.println(pos);
                 if(pos > list_plus.size() - 1)
```

```
System.out.println("Position specified is out of range!");
                  else
                   list_plus.remove(pos);
                   System.out.println("Item " + list_plus.items[pos] + " removed
from position " + pos + " in the list.");
               break;
           case 3:
               System.out.print("Enter position to retrieve item from: ");
               pos = Integer.parseInt(stdin.readLine());
               System.out.println(pos);
                 if(pos > list_plus.length())
                    System.out.println("Position specified is out of range!");
                  else
                       System.out.println("Item " + list_plus.get(pos) + " retrie
ved from position " + pos + " in the list.");
               break;
           case 4:
                System.out.println("Clearing list...");
               list plus.removeAll();
               System.out.println("List cleared.");
               break;
           case 5:
               if(list_plus.isEmpty())
                  System.out.println("List is empty.");
               else
                   System.out.println("List of size " + list plus.size() + " has
the following items: " + list_plus.toString());
               break;
           case 6:
               System.out.println("Reversing list...");
               list_plus.reverse();
               System.out.println("Reversed list: ");
                for (int index = 0; index < list_plus.numItems; index++)</pre>
                    System.out.println(list_plus.get(index) + "\n");
               break;
       }
Lab2P1Sampleruns.txt
```

```
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List is empty.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: Data
Enter the position to enter the item in: 0
Item Data inserted in position 0 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 1 has the following items: Data
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: Beverly
Enter the position to enter the item in: 0
Item Beverly inserted in position 0 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 2 has the following items: Beverly Data
Select from the following menu:
```

Select from the following menu:

0. Exit the program

4. Clear the list

1. Insert item into the list

2. Remove item from the list

3. Get item from the list

5. Print size and content of the list 6. Reverse the list Make your menu selection now: 1 You are now inserting an item into the list. Enter item: Jean-Luc Enter the position to enter the item in: 4 Position specified is out of range! Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 5 List of size 2 has the following items: Beverly Data Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 1 You are now inserting an item into the list. Enter item: Jean-Luc Enter the position to enter the item in: 2 Item Jean-Luc inserted in position 2 in the list. Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 1 You are now inserting an item into the list. Enter item: Geordi Enter the position to enter the item in: 1 Item Geordi inserted in position 1 in the list. Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 1

```
You are now inserting an item into the list.
Enter item: Worf
Enter the position to enter the item in: 3
Item Worf inserted in position 3 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 5 has the following items: Beverly Geordi Data Worf Jean-Luc
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 6
Reversing list...
Reversed list:
Jean-Luc
Worf
Data
Geordi
Beverly
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 5 has the following items: Jean-Luc Worf Data Geordi Beverly
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 5
Position specified is out of range!
Select from the following menu:
```

0. Exit the program

0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 2 You are now removing an item from the list. Enter position to remove item from: 3 Item Beverly removed from position 3 in the list. Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 5 List of size 4 has the following items: Jean-Luc Worf Data Beverly Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 2 You are now removing an item from the list. Enter position to remove item from: 0 Item Worf removed from position 0 in the list. Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 1 You are now inserting an item into the list. Enter item: Will Enter the position to enter the item in: 1 Item Will inserted in position 1 in the list. Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 5 List of size 4 has the following items: Worf Will Data Beverly

Select from the following menu:

```
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 3
Enter position to retrieve item from: 2
Item Data retrieved from position 2 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 3
Enter position to retrieve item from: 0
Item Worf retrieved from position 0 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 3
Enter position to retrieve item from: 7
Position specified is out of range!
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 4 has the following items: Worf Will Data Beverly
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 6
Reversing list...
Reversed list:
Beverly
Data
```

Antonio Rosado; 1

```
02/02/23
18:55:09
```

\* @version: 2023.02.02

```
Will
Worf
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 4 has the following items: Beverly Data Will Worf
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 4
Clearing list...
List cleared.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List is empty.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 0
Exiting program... good bye
ListArrayListBased.java
......
 * Purpose: Data Structure and Algorithms Review Programming Assignment
 * Status: Complete and thoroughly tested
 * Last update: 02/02/23
 * Submitted: 02/02/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
```

```
import java.util.ArrayList;
public class ListArrayListBased implements ListInterface {
   ArrayList<Object> items;
   public ListArrayListBased()
        items = new ArrayList<Object>();
     * Check if items ArrayList is empty.
     * @return items.isEmpty()
   public boolean isEmpty()
        return items.isEmpty();
     * Return size of ArrayList.
     * @return items.size()
    public int size()
        return items.size();
     * Add item to ArrayList.
     * @param int index
                            index of item
     * @param Object item item Object
    public void add(int index, Object item)
        items.add(index, item);
     * Retrieve item in ArrayList by index.
     * @param int index index of item
     * @return
                            item index
     * @throw
                            ListIndexOutOfBoundsException
    public Object get(int index)
        if(index >= 0 && index < items.size())</pre>
            return items.get(index);
        else
            throw new ListIndexOutOfBoundsException("ListIndexOutOfBoundsException
 on get");
```

```
/**
     * Remove item in ArrayList by index.
     * @param int index
                           index of item
     * @return
                               items.remove(index)
     * @throws
                            ListIndexOutOfBoundsException
    public Object remove(int index) throws ListIndexOutOfBoundsException
        return items.remove(index);
    /**
     * Retrieve all items in ArrayList.
     * @param int index
                           index of item
     * @throws
                           ListIndexOutOfBoundsException
   public void removeAll(int index) throws ListIndexOutOfBoundsException
        items.removeAll(items);
   @Override
   public void removeAll() {
        // TODO Auto-generated method stub
ListArrayListBasedPlus.java
/**
 * Purpose: Data Structure and Algorithms Review Programming Assignment
 * Status: Complete and thoroughly tested
 * Last update: 02/02/23
 * Submitted: 02/02/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.02.02
import java.util.ArrayList;
public class ListArrayListBasedPlus extends ListArrayListBased {
     * Reverses ArrayList if size cap is reached.
   public void reverse()
        ArrayList<Object> reversed = new ArrayList<Object> ();
        for(int index = items.size() -1; index >= 0; index--)
           reversed.add(items.get(index));
        items = reversed;
```

```
* Returns a string value of item(s) in ArravList
   public String toString()
        return items.toString();
Lab2P2Driver.java
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Lab2P2Driver extends ListArrayListBasedPlus{
    static BufferedReader stdin = new BufferedReader(new InputStreamReader(System.
   public static void main (String[] args) throws IOException
        ListArrayListBasedPlus list = new ListArrayListBasedPlus();
        boolean exit = false;
        int pos = -1;
        while (!exit) {
          System.out.println("Select from the following menu: \n"
                             + "0. Exit the program \n"
                             + "1. Insert item into the list \n"
                             + "2. Remove item from the list \n"
                             + "3. Get item from the list \n"
                             + "4. Clear the list \n"
                             + "5. Print size and content of the list \n"
                             + "6. Reverse the list \n ");
           System.out.print("Make your menu selection now: " );
           int input = Integer.parseInt(stdin.readLine());
           System.out.println(input);
           // possible cases for initial input
           switch (input) {
           case 0:
                System.out.println("Exiting program... good bye");
               exit = true;
               break;
           case 1:
               System.out.println("You are now inserting an item into the list.")
;
               System.out.print("Enter item: ");
               Object item = stdin.readLine();
               System.out.println(item);
                           System.out.print("Enter the position to enter the item
in: ");
                pos = Integer.parseInt(stdin.readLine());
                System.out.println(pos);
                if(pos <= list.size())</pre>
                 list.add(pos, item);
                 System.out.println("Item " + list.get(pos) + " inserted in posit
ion " + pos + " in the list.");
                el se
```

```
System.out.println("Position specified is out of range!");
               break;
            case 2:
                System.out.println("You are now removing an item from the list.");
                        System.out.print("Enter position to remove item from: ");
                pos = Integer.parseInt(stdin.readLine());
                System.out.println(pos);
                if(pos > list.size() - 1)
                    System.out.println("Position specified out of range!");
                else
                    list.remove(pos);
                    System.out.println("Item " + list.get(pos) + " removed from th
e list.");
               break;
            case 3:
                System.out.println("Enter position to retrieve item from: ");
                pos = Integer.parseInt(stdin.readLine());
                if(pos > list.size())
                    System.out.println("Position specified is out of range!");
                else
                    System.out.println("Item " + list.get(pos) + " retrieved from
position " + pos + " in the list.");
                    list.get(pos);
               break;
                System.out.println("Clearing list...");
                list.removeAll();
                System.out.println("List cleared.");
               break;
            case 5:
               if(list.isEmpty())
                  System.out.println("List is empty.");
                else
                    System.out.println("List of size " + list.size() + " has the f
ollowing items: " + list.toString());
               break;
            case 6:
                System.out.println("Reversing list...");
                list.reverse();
```

```
System.out.println("Reversed list: ");
                for (int index = 0; index < list.size(); index++)</pre>
                    System.out.println(list.get(index) + "\n");
                break;
::::::::::::::
Lab2P2Sampleruns.txt
......
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List is empty.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: Data
Enter the position to enter the item in: 0
Item Data inserted in position 0 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 1 has the following items: [Data]
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: Beverly
```

Enter the position to enter the item in: 0

```
Item Beverly inserted in position 0 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 2 has the following items: [Beverly, Data]
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: Jean-Luc
Enter the position to enter the item in: 4
Position specified is out of range!
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 2 has the following items: [Beverly, Data]
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: Jean-Luc
Enter the position to enter the item in: 2
Item Jean-Luc inserted in position 2 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 1
```

You are now inserting an item into the list.

```
Enter item: Geordi
Enter the position to enter the item in: 1
Item Geordi inserted in position 1 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 1
You are now inserting an item into the list.
Enter item: Worf
Enter the position to enter the item in: 3
Item Worf inserted in position 3 in the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 5 has the following items: [Beverly, Geordi, Data, Worf, Jean-Luc]
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 6
Reversing list...
Reversed list:
Jean-Luc
Worf
Data
Geordi
Beverly
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 5 has the following items: [Jean-Luc, Worf, Data, Geordi, Beverly]
Select from the following menu:
```

0. Exit the program

1. Insert item into the list

```
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 5
Position specified out of range!
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 3
Item Beverly removed from the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 4 has the following items: [Jean-Luc, Worf, Data, Beverly]
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 2
You are now removing an item from the list.
Enter position to remove item from: 0
Item Worf removed from the list.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
```

5. Print size and content of the list

You are now inserting an item into the list.

Enter the position to enter the item in: 1

Make your menu selection now: 1

6. Reverse the list

Enter item: Will

Item Will inserted in position 1 in the list. Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 5 List of size 4 has the following items: [Worf, Will, Data, Beverly] Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 3 Enter position to retrieve item from: Item Data retrieved from position 2 in the list. Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 3 Enter position to retrieve item from: Item Worf retrieved from position 0 in the list. Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 3 Enter position to retrieve item from: Position specified is out of range! Select from the following menu: 0. Exit the program 1. Insert item into the list 2. Remove item from the list 3. Get item from the list 4. Clear the list 5. Print size and content of the list 6. Reverse the list Make your menu selection now: 5 List of size 4 has the following items: [Worf, Will, Data, Beverly] Select from the following menu:

```
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 6
Reversing list...
Reversed list:
Beverly
Data
Will
Worf
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 4 has the following items: [Beverly, Data, Will, Worf]
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 4
Clearing list...
List cleared.
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 5
List of size 4 has the following items: [Beverly, Data, Will, Worf]
Select from the following menu:
0. Exit the program
1. Insert item into the list
2. Remove item from the list
3. Get item from the list
4. Clear the list
5. Print size and content of the list
6. Reverse the list
Make your menu selection now: 0
Exiting program... good bye
```