**CIS 255 - Java Programming**
**Homework #11**
**Lists & Maps**

**Name:** _____

**Date:** _____

1. **Given the declared and initialized LinkedList object below, display all the values in the list after each of the statement(s) is/are executed. If the list is empty, then write "empty". If the statement causes an error, then write "error". (Each statement builds on the one before)**

**List<String> list = new LinkedList<String>();**

**a.)** `list.add("Mercedes");`

**b.)** `list.add("BMW");`

**c.)** `list.add("Audi");`

**d.)** `list.remove(1);`

**e.)** `list.add("Mercedes");`

**f.)** `list.clear();`

**g.)** `list.add("Yugo");`

**h.)** `list.add("Honda");`

**i.)** `list.set(1, "Chevy");`

**j.)**
```
List<String> fix =
    new LinkedList<String>();

fix.add("Yugo");
list.removeAll(fix);
```

**k.)** `list.remove(0);`

**2. Given the declared and initialized LinkedHashMap object below, display all the keys and values in the map after each of the statement(s) is/are executed. If the map is empty, then write "empty". If the statement causes an error, then write "error". (Each statement builds on the one before)**

**Map<String, String> map = new LinkedHashMap<String, String>();**

**a.)** `map.put("M", "Mercedes");`

**b.)** `map.put("B", "BMW");`

**c.)** `map.put("A", "Audi");`

**d.)** `map.remove("b");`

**e.)** `map.clear();`

**f.)** `map.put("M", "Mitsubishi");`

**g.)** `map.put("F", "Ford");`

**h.)** `map.remove("M");`

**i.)**
```
Map<String, String> fix =
  new LinkedHashMap<String, String>();

fix.put("Y", "Yugo");
fix.put("C", "Chevy");
map.putAll(fix);
```

**j.)** `map.put("M", "Mercedes");`