Technical Practice - Solution

Problem 1 - Going Away Cards

Python

Solution without duplicate checks:

```
CARD_COMMON_TEXT = "Hello there, and Thank You! "
def everyone_sign(names) :
    result = {}
    for name in names :
        signature = ""
        for name_tentative_signature in names:
            if name != name_tentative_signature:
                signature = signature + name_tentative_signature + ", "
        signature = signature[0:-2]
        signature = signature + "."
        result[name] = CARD_COMMON_TEXT + signature
    return result
```

Solution with duplicate checks:

```
CARD_COMMON_TEXT = "Hello there, and Thank You! "
def everyone_sign(names) :
    result = {}
    for name in names :
        list_copy = list(names)
        list_copy.remove(name)
        result[name] = CARD_COMMON_TEXT + ",".join(list_copy)
    return result
```

Another solution

```
# https://codehabitude.com/2013/12/24/python-objects-mutable-vs-immutable/
```

```
CARD_COMMON_TEXT = "Hello there, and Thank You! "
def everyone_sign_more_efficient(names):
    result = {}
    for name in names:
        other_names = []
        for other_name in names:
            if other_name != name:
                other_names.append(other_name)
        result[name] = CARD_COMMON_TEXT + ", ".join(other_names) + "."
    return result
```

And a fourth using list comprehension

```
# http://treyhunner.com/2015/12/python-list-comprehensions-now-in-color/
CARD_COMMON_TEXT = "Hello there, and Thank You! "
def everyone_sign_list_comprehension(names):
    result = {}
    for name in names:
        other_names = [other_name for other_name in names if other_name != name]
        result[name] = CARD_COMMON_TEXT + ", ".join(other_names) + "."
    return result
```

Java

```
public class Cards {

  /**
   * @param names A list of each participant's name.
   * @return A map from each participant's name to the list of every
   *     other participants.
   * @throws IllegalArgumentException If names is null, names contains null
   *     items,  or if names contains duplicate items.
   */
  public static Map<String, List<String>> sign(List<String> names) {
    if (names == null) {
      throw new IllegalArgumentException("names cannot be null");
    }
    for (String name : names) {
      if (name == null) {
        throw new IllegalArgumentException("names cannot contain null");
      }
    }
    // Converting a list to a Set removes duplicates.
```

```
    if (new HashSet(names).size() != names.size()) {
      throw new IllegalArgumentException("names cannot contain duplicate items");
    }

    Map<String, List<String>> signaturesByName =
        new HashMap<String, List<String>>();
    for (String name : names) {
      List<String> signatures = new ArrayList<String>();
      for (String signer : names) {
        if (!signer.equals(name)) {
          signatures.add(signer);
        }
      }
      signaturesByName.put(name, signatures);
    }
    return signaturesByName;
  }
}
```