

Collections - Lists and Stacks

Programming Projects:

1) A NameManager class

- Start Eclipse and create a new Java project called 'lab5'
- Create class called **NameManager**
- Add an attribute called 'names' that stores a list of String type elements, as follows -


```
List<String> names = new ArrayList<String>();
```
- Implement a method called **addName(String name)** that adds the given name to the 'names' list.
- Implement a method called **printNames()** that iterates over the list and prints each name to the console.
- Implement a method called **removeLongNames()** that iterates over the list and removes any names which are more than 15 characters in length. *Hint:* Use an explicit **Iterator** and its **remove()** method.
- Now add a **Driver** class which creates an instance of the **NameManager** class and calls the methods for testing, e.g.

```
NameManager manager = new NameManager();

manager.addName("M.Mickleson");
manager.addName ("Johnua Taylor Biggs");
manager.addName ("P.Smith");
manager.addName ("Peter Jonathan Smythton");
manager.addName ("P.Thompson");

manager.printNames();    // should print all names

manager.removeLongNames();

manager.printNames();    // should now have longer names removed
```

Think you're finished? Check you have commented your code and added Javadoc to the methods and classes.

2) A NumberSorter class

This exercise demonstrates how two **Stack** type collections can be used to implement a simple (but inefficient) sorting algorithm.

- Create class called **NumberSorter**
- Add a method with the following signature -
void sort(int [] numbers)
- Now implement this method using the flow chart shown on the next page, start with this code -

```
void sort(int[] numbers) {  
    // The source stack, from which the numbers are popped  
    Stack<Integer> srcStack = new Stack<Integer>();  
  
    // The dest stack, to which the numbers are pushed  
    Stack<Integer> destStack = new Stack<Integer>();  
  
    // Add the initial array of numbers to the source stack.  
    for (int number : numbers)  
        srcStack.push(number);  
  
    // bulk of code goes here (hint: use while loops rather than if's)  
}
```

- Now add a **Driver** class which creates an instance of the **NumberSorter** class and calls the method for testing, e.g.

```
NumberSorter sorter = new NumberSorter();  
  
sorter.sort(new int [] {1,9,2,3,10,8,12,1, 99, 2, 43, 68, 109, 0});  
sorter.sort(new int [] {1,2,3,4});  
sorter.sort(new int [] {4,3,2,1});
```

- Change the sort method so it takes an extra parameter, which determines whether the sort should be ascending or descending, e.g.

```
void sort(int [] numbers, boolean ascending)
```

- Update the implementation so that it uses this flag correctly, then add some test code within the driver class to ensure the method is working properly.

Think you're finished? Check you have commented your code and added Javadoc to the methods and classes.

Stack Sorter - Flow Chart

