CS 1501
Algorithm Implementation
Summer 2017
Programming Project 2

Online: Sunday, June 4, 2017
Due: All assignment materials: 1) Your PHPArray.java file, 2) The Assig2.java main program file 3) Any other files that you have written 4) **[W section only]** OOP paper and 5) Assignment Information Sheet. All materials must be zipped into a single .zip file and submitted **via the course submission site** by **11:59 PM on Saturday, June 17, 2017**.
Late Due Date: 11:59PM on Monday, June 19, 2017

Note: Do not submit a NetBeans or Eclipse (or any other IDE) project file. If you use one of these IDEs make sure you extract and test your Java files WITHOUT the IDE before submitting.

Background:
The PHP language has an interesting array data type. Rather than a simple array like that of C or Java, the PHP array is actually a hybrid of a hash table and a linked list. This allows for a lot of functionality, including hash table access, indexed integer access, and sequential access (via the linked list). For more information on PHP arrays, see: http://php.net/manual/en/language.types.array.php.

Details:
In this project you will approximate a PHP array using Java. Your basic approach should be as follows:

- You will create a new parameterized Java class with the following header:

```
public class PHPArray<V> implements Iterable<V>
```

  This will allow a user to instantiate a PHPArray of an arbitrary Java type and to iterate through its values using the predefined Java **for** loop.
- Your PHPArray class will be a symbol table and will thus store data as (key, value) pairs, where the keys will be Java Strings and the values will be the parameter type V
- Your PHPArray will use a **linear probing hash table** as part of its underlying storage mechanism. This will enable the put(), get() and unset() [i.e. delete] methods to be done in O(1) time. Your hash table should handle delete and should automatically resize by doubling the table size prior to a put() if the table is >= ½ full. For help on this part of your project, see the author's LinearProbingHashST.java class. Note, however, that there will be some differences in your class from this due to the requirement to also keep the data in insertion order via a linked list. For example, I do not recommend keeping the keys and values in separate parallel arrays, like the author does in LinearProbingHashST. Consider rather a linked type (i.e. a node) that contains both key and value fields. These objects could then be stored in your hash table and also accessed sequentially via a linked list.
- Your PHPArray will also use a **linked list of data** to keep track of the items in sequential order (based on the order that they were put() into the PHPArray). This ordering will be that produced / accessed by the Iterator() and some other methods as well. **It is very important that this linked access does not alter the constant time access of the hash table.** Thus you should think carefully about how you will implement this. For example, keeping a completely separate hash table and linked list will not work, since to remove an item from the linked list requires linear time.
- There is a lot of other functionality required of your PHPArray class, based loosely on the functionality of the actual PHP array. **This functionality is demonstrated in the main program Assig2.java. See the extensive comments in this document to determine the methods that you must implement and see the output produced in a2out.txt to see the required output.**

Important Notes:

- If you are making an array of a parameterized type in Java, there are some issues with the array instantiation that may cause you grief. Below is a link that may help. You can also Google this to see other potential solutions.
  http://stackoverflow.com/questions/19478225/array-of-generic-nodes-java
- You must utilize the Java Iterable and Iterator interfaces in this project. For help with these see the Java API and also your CS 0445 Text. I also have some notes and examples on my CS 0445 Web page. See:
  http://www.cs.pitt.edu/~ramirez/cs445/
- **W students:** In addition to your program you must **write a short (4-6 pages, double-spaced) technical paper on Object-Oriented Programming.** Include a brief history, the primary features, some example languages and key

differences between them. **You must use at least two verifiable (i.e. published) references that are NOT Wikipedia.**

- **Extra Credit:** There are a LOT of additional methods / functions available for PHP Arrays. If you implement one or more of these that are non-trivial you can get some extra credit. See: http://php.net/manual/en/ref.array.php