Homework 4

Due on November 6, 2016

October 20, 2016

Notes:

- Please zip the files into a single file named "Homework4.zip" and upload it into the grading area.
- I will use my own Java main method by importing the package into my Java class method to test your code. Please be sure that all the constructors and methods are working in your own main method i.e. design a wide testing main to be sure that everything is working fine. We will use our own main to do the final testing.

1. Given the interface for Stack, please create two implementations for it:
   (a) Derived from a Chain List
      i. Name "MyDerivedStack"
   (b) From Scratch
      i. Name "MyScratchStack"

   Please use package MyStack for the interface and the two different implementations.

2. Given the interface for Queue, please create an implementation for it:
   (a) From Scratch
      i. Name "MyScratchQueue" using a Chain list (Not extending, but making everything from scratch)
      ii. Use this implementation to implement the Radix Sort (Name "MyRadixSort") in Radix 10
         A. Input: The numbers, 20,000 of them, are stored in an array (i.e each number is 10 digits long) in the following way:

number 1	number 2	number 3	number 4	...	number 20,000

         Each slot represent 10 elements in the array.
         B. Output : the sequence of indexes in the order of the numbers after sorting

         We use this because we do not have pointer in Java, and actually this will be the best way to implement this.

   Please use package MyQueue for the interface and the two different implementations.

3. Implement the ADT dictionary by using hashing and separate chaining. Use a chain of linked nodes as each bucket.

   (a) Implement the Universal Hashing with the modules (It was explained in class)

   (b) The name of the java class is "MyHahsTable"

   (c) Package name HashTable

4. Design a class PatientDataBase that stores instances of PatientRecord. Each record contains an name for a patient and strings for the date, the reason for the visit, and the treatment prescribed.. The class should provide at least three query operations, as follows. Given a patient name and date, the first operation should return the reason for the visit, and the second operation should return the treatment. The third query operation should return a list of dates, given a patient name.

5. Please Implement the following interfaces for a binary tree

```
package TreePackage;
public interface BTNode {
   public int gKey();
   public BTNode gLeft();
   public BTNode gRight();
   public void sKey(int AValue);
   public void sLeft(BTNode AValue);
   public void sRight(BTNode AValue);
}

public interface BTree {
   public boolean IsEmpty();
   public BinaryNode gRoot();
   public int Count();
   public int Size(BinaryNode ATree);
   public int Height(BinaryNode ATree);
   private int Size()
}
```

   (a) Use the following names "BinaryNode" and "BinaryTree" for the Java classes.

Note: We will copy and run our own main into the code, please use the names described in the class.