

March 16, 2016

Computer Science 182 Programming Assignments #4

Due: April 4, 2016

Complete the classes below so that all of the methods do what they are supposed to do in the way they are supposed to be done. As always, leave all variable and method names exactly as they appear and keep in mind that the specification, required algorithms, etc. may change according to lab discussions. I might be adding a few more Word methods when we get back from Spring Break. You are responsible for making sure your program adheres to all the most recent requirements. With the three obvious exceptions of one of the constructors, the toString method, and the myName method I do not want to see any String variables in your program. There should be no arrays anywhere in your program. Methods that can (and should) be done recursively can be written iteratively if you prefer but there will be a 5% total points penalty every time you do this.

Soon I will post a test program so that you can test your methods, but there are two very good reasons not to wait for mine: (1) The sooner you start testing, the better, and (2) Using my test program is a terrible way to debug – but you already know this! Speaking of debugging, don't forget to quickly and generously use print statements and/or learn to use the debugger.

As always, I will need a printout of your program listing, a printout of the output produced by my *final* test program, and an email with an attached file named prog4.java that contains your classes. The email subject should be “Alan Turing – prog4” if your name is Alan Turing.

Remember no sharing of code. Providing code for others or taking code from others are equally serious offenses.

```
class Node {

    // instance variables for the usual linked list object
    private char letter;
    private Node next;

    // constructor
    public Node(char ch, Node link) {

        // getters and setters
        public void setLetter(char letter) {

        public char getLetter() {

        public void setNext(Node next) {

        public Node getNext() {
    }
}

class Word {

    // instance variable pointing to the head of the linked list
    private Node head;

    // default constructor
    public Word() {

    // copy constructor
    public Word( Word ) {

    // constructor from a String
    public Word( String s ) {

    // for output purposes -- override Object version
    // no spaces between the characters, no linefeeds/returns
    public String toString() {
```

```

// how long is this Word
public int length() {

// append ch to the head of this Word
public void addHead( char ch ) {

// append ch to the tail of this Word
public void addTail( char ch ) {

// modify this Word so it is reversed
public void reverse() {

// remove all occurrences of ch from this Word
public void removeChar( char ch ) {

// remove the first occurrence of the Word w from this Word
public void removeWord( Word w ) {

// concatenate a copy of s to the end of this Word
// (Why do I want to use a copy?)
public void concatenate( Word s ) {

// reverse the first occurrence of w in this word
// for example, if this word is "helloworld" and w is "low" then
// this word should end up being "helwolorld"
public void flipWord( Word w ) {

// who are you?
public static String myName() {

}

```