

## **CMSC 131 Fall 2004 Quiz 5 Solution**

Here is one possible solution. We write a loop that goes through half of the array (from 0 up to  $\text{length}/2$ ). We test whether the character at index  $i$  is equal to the character at index  $\text{length}-1-i$ . (Beware of off-by-1 errors. When  $i$  is 0, we want to test  $\text{word}[0]$  against  $\text{word}[\text{length}-1]$ .) If we ever find a pair of values that differ, we return false. Otherwise, if all the characters match, we return true.

Observe that we could have allowed  $i$  to run through the entire array, but this would have involved unnecessary effort. A common error was to set a boolean variable to true or false with each iteration of the loop, and then return its value on termination of the loop. Note that this would not achieve the desired result, since it only returns the result of the very last comparison. If you were to use a boolean variable, the right way is to set it to true before starting the loop, and then set it to false within the loop whenever a mismatch is found. This way, a later match cannot override a previous mismatch.

```
public static boolean palindrome(char[] word) {
    for (int i=0; i<word.length/2; i++) {
        if (word[i] != word[word.length - 1 - i])
            return false;
    }
    return true;
}
```