Mastery Palindrome Reflection Log

Credit Name: CSE 2120 Data Structures 1
Assignment Name: Mastery Palindrome

How has your program changed from planning to coding to now? Please explain?

From planning to coding I took the definition of the application from the computer science textbook and started to create each part of code to meet the necessary requirements. Each part became more complex as I made a line of code for the Array. Although my original code wasn't case-insensitive I switched this after realizing as further explained in my error log.

From coding to now the only change I made was fixing the case-insensitivity in my program as explained more in my error log.

Steps

(the package this file is in is Mastery)

I started off by importing java.util.scanner which allows me to get different data types from user inputs. The public class is declared as Palindrome.

```
package Mastery;

import java.util.Scanner;

public class Palindrome {
```

Below it shows the public (accessible from other classes) method "main" is static (you can access it without creating an object), void (method doesn't return any value), main (method name), and String[ ] args (String represents the data type [ ] indicates that it will be an array, and args (arguments) is the name of the array).

```
public static void main(String args[]) {
```

The Scanner object is created below to take in inputs from the user. (I named mine scanner)

```
Scanner scanner = new Scanner(System.in);
```

I then prompt the user to enter a word that they want to know is a palindrome or not.

```
System.out.println("Enter a word to check if it is a palindrome:");
```

I save this as a String variable called palindromeCheck

```
String palindromeCheck = scanner.nextLine();
```

Next, I switch this variable to all lowercase so that it is case-insensitive when being checked. For example if the user entered Kayak then reversed this would be kayaK, and when the program compares these they would not be equal since the uppercase letters are in different places (this isn't right because kayak is a palindrome).

```
palindromeCheck = palindromeCheck.toLowerCase();
```

After this, I create a character array (Char), that takes in the user's input (palindromeCheck). This array will split the word into its individual characters (letters) and make each one of these an element within the array (the part of code that puts palindromeCheck into the array is the palindromeCheck.toCharArray() method).

```
char[] charArray = palindromeCheck.toCharArray();
```

I then create an empty String variable called reversed that will be used in the for loop (below this code) to check against the original to see if the inputted String is an array.

```
String reversed = "";
```

After this, I create the for loop. I make the variable "i" (index) equal to the letters in the character array through the line charArray.length - 1. charArray.length just means that all of the characters (length) will be "i" (become an index), the -1 part of this code is to ensure that the last letter is checked first (the length hence .length of Char array may be 5 elements for example length = 5, but since indexes "i" begin at 0 you subtract (-1) from the length (Ex. index 4 = element 5))(NOTE: that arrays also start at 0 but we are not counting the place value of the array but the total length (amount of elements)). The next part says as long as i >= 0 the loop will continue; this is because you're going through the word backwards (right to left) so once you get to the last letter (index("i") = 0) so you stop the loop here. And the lastly, you subtract 1 from "i" so that once it finishes with one letter (index) from the Char array it will move onto the next one (we're going backwards hence why you -1 not +1)

```
for (int i = charArray.length - 1; i >= 0; i--) {
```

I then took the string variable reversed that I initialized before the for loop as "" (nothing), and added each index from the char Array (these will be added in reversed because of how the for loop reversed each character); this made the variable reversed equal to the word backwards (Ex. meat would become team reversed)

```
(explained more in my reflection log)
*/
    reversed += charArray[i]; // The variable reve
} // Close the part of the for loop that adds char
```

NOTE: from here on my screenshots have a white background because I used my eclipse at home

Next I created an if-else statement. The first part of the this statement (if part) was for if the original inputted string “palindromeCheck” was equal (I just use the .equals method, this will check if the reversed string is equal to the original string input, it would be equal if all of the letters backwards are the same as the letters forwards, hence it’s a palindrome). If this is true (correct) then the program will output the print statement saying “The entered word is a palindrome.”.

```
if (palindromeCheck.equals(reversed)) { // Checks if the input
    System.out.println("The entered word is a palindrome."); //
```

The second part of this if-else statement (if part) is for if the reversed string variable does not equal the original imputed string value. If it checks each character against each other and they aren’t equal somewhere within the word, then the message “The entered word is NOT a palindrome.” will be printed.

```
} else { // For if the reversed input does NOT equal the original
    System.out.println("The entered word is NOT a palindrome.");
} // close the else statement (end of if-else statement)
```

I then close the scanner with the .close() method. This releases the system.in resources that are used for inputs from your console (keyboard). It is not super necessary in this code as no errors would occur if it wasn’t there, however after reading some comments from StackOverflow and W3schools it became clear that this is a good common practice for future longer and more complex programs.

```
scanner.close();
```

After this I close the main method with a brace.

```
}
```

And close the public Palindrome class with a brace.

```
}
```

Together they look like this.

```
    }
} //
```