

# Palindrome Mastery ReflectionLog

First rendition of Palindrome.java:

```
package Mastery;

import java.util.Scanner;
import java.util.ArrayList;

public class Palindrome {

    public static void main(String[] args) {

        ArrayList<Character> letter = new ArrayList<Character>();

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a word: ");
        String word = input.nextLine();

        int length = word.length();

        for(int i = 0; i < length; i++) {
            letter.add(word.charAt(i));
        }
        System.out.print(letter);

    }

}
```

First I created an ArrayList named letter that would store all of the letters in the inputted string. Then I saved the length of the string using the .length() method, and ran it through a for statement that would append each letter of the string to the ArrayList.

First rendition of Palindrome.java:

```
package Mastery;

import java.util.Collections;
import java.util.Scanner;
import java.util.ArrayList;

public class Palindrome {

    public static void main(String[] args) {

        ArrayList<Character> letter = new ArrayList<Character>();
        ArrayList<Character> reverseLetter = new ArrayList<Character>();

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a word: ");
        String word = input.nextLine();
        word = word.replaceAll(" ", "");

        int length = word.length();

        for(int i = 0; i < length; i++) {
            letter.add(word.charAt(i));
        }

        System.out.print(letter);
        Collections.reverse(letter);
        System.out.print(letter);
        System.out.println();

    }

}
```

In my second rendition I declared a second ArrayList that would serve as the inverse to the first, this way I could compare the lists to find if the inputted string is a palindrome. I also removed all spaces from the inputted string by using a `.replaceAll` object so that sentence-type palindromes would also work. Then I saved the second ArrayList as the first, and used the `Collections.reverse()` to reverse the second ArrayList, unfortunately this worked as a shallow copy, so by reversing the second ArrayList the first would be reversed as well.

Third rendition of Palindrome.java:

```
package Mastery;

import java.util.Collections;
import java.util.Scanner;
import java.util.ArrayList;

public class Palindrome {

    public static void main(String[] args) {

        ArrayList<Character> letter = new ArrayList<Character>();

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a word: ");
        String word = input.nextLine();
        word = word.replaceAll(" ", "");

        int length = word.length();

        for(int i = 0; i < length; i++) {
            letter.add(word.charAt(i));
        }

        ArrayList<Character> reverseLetter = (ArrayList<Character>)letter.clone();
        Collections.reverse(reverseLetter);

        System.out.print(letter);
        System.out.print(reverseLetter);
        System.out.println();

    }

}
```

In the third rendition, instead of just referencing the second ArrayList to the first, I created a copy using the `.clone()` method. This worked as a deep copy, so reversing the second would retain the values of the first ArrayList.

Fourth rendition of Palindrome.java:

```
package Mastery;

import java.util.Collections;
import java.util.Scanner;
import java.util.ArrayList;

public class Palindrome {

    public static void main(String[] args) {

        ArrayList<Character> letter = new ArrayList<Character>();

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a word: ");
        String word = input.nextLine();
        word = word.replaceAll(" ", "");

        int length = word.length();

        for(int i = 0; i < length; i++) {
            letter.add(word.charAt(i));
        }

        ArrayList<Character> reverseLetter = (ArrayList<Character>)letter.clone();
        Collections.reverse(reverseLetter);

        System.out.print(letter);
        System.out.print(reverseLetter);
        System.out.println();

        if (letter.equals(reverseLetter) == true) {
            System.out.print(letter + " IS a palindrome.");
        } else {
            System.out.print(letter + " is NOT a palindrome.");
        }

    }

}
```

In the fourth rendition I added an if else statement that would compare the two strings, if the comparison returned true the program would print that it is a palindrome, otherwise it would print false. Unfortunately the program was case sensitive resulting in scenarios such as the one below:

```
Enter a word: Mom
[M, o, m][m, o, M]
[M, o, m] is NOT a palindrome.
```

Final rendition of Palindrome.java:

```
package Mastery;

import java.util.Collections;
import java.util.Scanner;
import java.util.ArrayList;

public class Palindrome {

    public static void main(String[] args) {

        //Creates letter ArrayList object
        ArrayList<Character> letter = new ArrayList<Character>();

        //Preparing for user input
        Scanner input = new Scanner(System.in);

        //Prompts and records user input
        System.out.print("Enter a word: ");
        String word = input.nextLine();

        //Converts to lowercase and removes spaces from the word/phrase
        word = word.toLowerCase();
        word = word.replaceAll(" ", "");

        //Gets length of user input
        int length = word.length();

        //Loops for the number of characters in the word/phrase
        for(int i = 0; i < length; i++) {
            //Adds char from word/phrase to ArrayList
            letter.add(word.charAt(i));
        }

        //Creates a second ArrayList object and reverses it
        ArrayList<Character> reverseLetter = (ArrayList<Character>)letter.clone();
        Collections.reverse(reverseLetter);

        //Checks if both ArrayList objects are equal
        if (letter.equals(reverseLetter) == true) {
            //Prints object is a palindrome
            System.out.print(word + " IS a palindrome.");
        } else {
            //Prints object isn't a palindrome
            System.out.print(word + " is NOT a palindrome.");
        }

    }

}
```

In my final rendition I added a `.toLowerCase()` method that would ensure capitalization wouldn't matter, along with adding comments.

Final output example:

Enter a word: <b>Never odd or even</b> neveroddoreven IS a palindrome.	Enter a word: <b>Always even or odd</b> alwaysevenorodd is NOT a palindrome.
---	---