

## Assignment 4 – Part 2: Palindromes

Ryan Brinson

9/15/2023

Output:

```
rbrinson2@DESKTOP-U8KJ40P:~/Documents/CS3305/A4$ java A4P2
Enter word to test:
abcdefg

The word is not a palindrom.
rbrinson2@DESKTOP-U8KJ40P:~/Documents/CS3305/A4$ java A4P2
Enter word to test:
racecar

The word is a palindrome!
rbrinson2@DESKTOP-U8KJ40P:~/Documents/CS3305/A4$ |
```

Code:

```
// Name: Ryan Brinson
// Class: CS 3305 W04
// Term: Fall 2023
// Instructor: Carla McManus
// Assignment: 04-Part-2-Palindromes

import java.util.Scanner;

public class A4P2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Get a word from the user and pass it to the constructor
        System.out.println("Enter word to test: ");
        Palindrom pali = new Palindrom(input.nextLine());

        if (pali.IsPalindrom() == true)
            System.out.println("\nThe word is a palindrome!");
        else
            System.out.println("\nThe word is not a palindrom.");

        input.close();
    }
}
```

```

class Palindrom{
    private char[] Stack_1;
    private char[] Stack_2;
    private char[] Stack_3;
    private int top_1;
    private int top_2;
    private int top_3;

    // Constructor that takes in word and fills the three stacks
    Palindrom(String word){
        // Initialize the tops of the stacks
        top_1 = -1;
        top_2 = -1;
        top_3 = -1;

        // Convert the user word to a char array and
        // store in Stack 1
        Stack_1 = word.toCharArray();
        // update the top of stack 1
        top_1 = Stack_1.length - 1;

        // Clone stack 1 into stack 2
        Stack_2 = Stack_1.clone();

        // Initialize the length of stack 3
        Stack_3 = new char[Stack_1.length];
        while (!is_empty_1()){
            push_3(pop_1());
        }
    }

    // Push values onto Stack 3
    private void push_3(char c){
        //Check if the stack is full
        if(top_3 == (Stack_3.length - 1)) System.out.println("Stack 3 full");
        // If it is not full add the char to the top of the stack
        else {
            Stack_3[top_3 + 1] = c;
            top_3++;
        }
    }

    // Test if Stack 1 is empty or not
    private boolean is_empty_1(){

```

```

        return top_1 < 0;
    }

    // Pop the stack of stack 1
    private char pop_1(){
        // set a temp value to the top value
        char temp = Stack_1[top_1];
        // set the top value to the null character
        Stack_1[top_1] = '\0';
        // decrement the stack
        top_1--;
        // return the temp value
        return temp;
    }

    public boolean IsPalindrom(){
        // Convert Stack 2 and 3 to strings
        String temp1 = String.valueOf(Stack_2);
        String temp2 = String.valueOf(Stack_3);
        // Use the string .equals to compare if they are the same
        return temp1.equals(temp2);
    }
}

```