Before creating my public class, I imported three packages. I imported java.lang, java.util, and java.io so that I could use the contain() method. This method allows for my code to determine if my string has a specified character (such as an opening or closing delimiter). The first line in my class was the string containing the delimiters. Then I declared multiple variables. I declared one variable per each delimiter. One for each opening delimiter and one for each closing delimiter. This allows me to later use if statements to compare if an opening delimiter is found, and if it can be matched to a closing delimiter.

For Loop checking for matching closing delimiter:

* The closing delimiter check if statement was strategically placed inside of the for loop checking for the open delimiter and outside the for loop checking for a closing delimiter.
* The algorithm starts at a for loop that iterates through each character in the string comparing it to a variable that has already been declared. This for loop continuously runs for the input string length – 1. Each character of the input string must be converted from a character to a string for my comparison to work. To do this, I create a string variable “letter” in which I store the converted character of the input string at index “i”. This allows for each individual letter of the input string to be looked at, rather than the input string as a whole. The first if statement in the for loop that stores each input string character to “letter”, checks to see if there is an opening delimiter. This first if statement checks for an opening brace delimiter. If an opening brace delimiter is found, a message is outputted: “Opening BRACE found”. In addition to this, “letter” which is also the opening brace is pushed into the stack. Once the opening brace has been pushed into the stack, a second for loop begins to iterate. This for loop iterates through for the input string length – 1 times. Inside this for loop, a new string variable must be declared for each character in the input string to be stored. This new string variable that converts each character to a string is named “next\_letter”. Inside this new for loop, a new if statement is comparing “next\_letter” to a matching closing delimiter. If the closing brace is found a message is outputted: “Closing BRACE found”. Then the stack pops the opening brace and integer score is set equal to a number (for brace delimiters score is set equal to five). Outside of the closing brace if statement and next letter for loop, is an if statement. This if statement is inside the opening brace if statement. This runs if an opening brace is found after it has been determined if there is a closing brace or not. This if statement compares if score is not equal to five. If score is equal to five, indicating that a closed brace has been found resulting in the opening brace being popped from the stack, the algorithm will continue onto the next if statement. If score is not equal to five, this indicates that a matching closing delimiter was not found anywhere in the input string. This results in a message being outputted telling the user there is an error and that a closing matching delimiter has not been found.
* each item in this string twice before determining if the integer score was equal to five. If score is equal to five this represents that a closing delimiter has been found. With the closing delimiter found, the opening delimiter can then be popped from the stack. If score is found to be not equal to five, the algorithm outputs a statement saying “error, no closing delimiter found!”.