

Cooney

Cooney - a game which has you guess a word rule based on your input.

- **Authors:** Rudd Fawcett
- **Program:** Cooney.java
- **Due:** 11/05/2015

```
java Cooney
```

```
ch08 — 1
[mbpro:ch08 ruddfawcett$ java Cooney
Try to figure out what Cooney likes:

Cooney likes MOON but he doesn't like SUN
Cooney likes SOCCER but he doesn't like HOCKEY
Cooney likes SUMMER but he doesn't like SPRING

Does Cooney like: grapes
No, Cooney doesn't like GRAPES

Does Cooney like: pears
No, Cooney doesn't like PEARS

Does Cooney like: apples
Yes, Cooney likes APPLES

Does Cooney like: pineapple
Yes, Cooney likes PINEAPPLE

Does Cooney like: raspberries
Yes, Cooney likes RASPBERRIES

Does Cooney like: strawberries
Yes, Cooney likes STRAWBERRIES

Does Cooney like: kiwi
No, Cooney doesn't like KIWI

Does Cooney like: apples
Yes, Cooney likes APPLES

Does Cooney like: pineapple
Yes, Cooney likes PINEAPPLE

Does Cooney like: raspberries
Yes, Cooney likes RASPBERRIES

Does Cooney like: strawberries
Yes, Cooney likes STRAWBERRIES

Does Cooney like: blueberries
Yes, Cooney likes BLUEBERRIES

I think you got the rule... I quit!
Thaanks forr pplaying!
mbpro:ch08 ruddfawcett$
```

Cooney.java

```

1  /**
2   * Cooney
3   * Cooney is a game which has you guess a word rule based on your input.
4   *
5   * @author   Rudd Fawcett
6   * @course   Period 1 AP CS, Dr. Miles
7   * @due      2015-11-05
8   */
9
10 import java.util.Scanner;
11
12 /**
13  * Class declaration of Cooney.
14  */
15 public class Cooney {
16     private static int counter;
17     private static Scanner keyboard;
18
19     /**
20      * Starts the game.
21      */
22     public static void run() {
23         counter = 0;
24
25         System.out.println("Try to figure out what Cooney likes:\n");
26         System.out.println("Cooney likes MOON but he doesn't like SUN");
27         System.out.println("Cooney likes SOCCER but he doesn't like HOCKEY");
28         System.out.println("Cooney likes SUMMER but he doesn't like SPRING\n");
29
30         keyboard = new Scanner(System.in);
31
32         process();
33     }
34
35     /**
36      * Stops the game, and closes the scanner.
37      */
38     public static void stop() {
39         keyboard.close();
40     }
41
42     /**
43      * Handles looping and stops the game if rule is figured out.
44      */
45     public static void process() {
46         while (counter < 5) {
47             ask();
48         }
49
50         System.out.println("\nI think you got the rule... I quit!");
51         System.out.println("Thaanks forr pplaying!");
52     }

```

```

52     ↪
53     ...stop();↪
54 }↪
55 ↪
56 ↪ /**↪
57 ↪ * Asks the user for input, and prints out the formatted result.↪
58 ↪ */↪
59 ↪ public static void ask() {↪
60     ...System.out.print("Does Cooney like: ");↪
61     ...String input = keyboard.nextLine();↪
62     ↪
63     ...String formatted = formattedRule(input);↪
64     ...System.out.println(formatted);↪
65 }↪
66 ↪
67 ↪ /**↪
68 ↪ * Formats a string based on whether or not the word obeys the rule.↪
69 ↪ * @param input The word that the user inputted as an answer.↪
70 ↪ * @return      The formatted string based on the word's validity.↪
71 ↪ */↪
72 ↪ public static String formattedRule(String input) {↪
73 ↪     if (obeysRule(input)) {↪
74         ...return "Yes, Cooney likes " + input.toUpperCase() + "\n";↪
75     }↪
76 ↪     else {↪
77         ...return "No, Cooney doesn't like " + input.toUpperCase() + "\n";↪
78     }↪
79 }↪
80 ↪
81 ↪ /**↪
82 ↪ * Returns whether or not a word is valid.↪
83 ↪ * @param word The user inputted word.↪
84 ↪ * @return      The word's validity.↪
85 ↪ */↪
86 ↪ public static boolean obeysRule(String word) {↪
87 ↪     for (int i = 0; i < word.length() - 1; i++) {↪
88 ↪         if (word.charAt(i) == word.charAt(i + 1)) {↪
89             ...counter++;↪
90             ...return true;↪
91         }↪
92     }↪
93     ↪
94     ...counter = 0;↪
95     ...return false;↪
96 }↪
97 ↪
98 ↪ public static void main(String[] args) {↪
99     ...run();↪
100 }↪
101 }↪
102

```