

Capstone Project: Six Little Words

Puzzle Class

```
package org.coursera.lab.capstone;

import java.util.ArrayList;

/*
 * Your solution for OOAD Course 1 Capstone Project
 * Kate Pendavinji
 * March 1, 2025
 * Each file highlights a thorough breakdown of the OOAD concepts
 * and instructions followed to achieve the results for the puzzle
 * game six little words.
 */

/* Create a class called "Puzzle" to contain the main for
 * the Java program and instantiate the three class instance
 * objects you'll use to run the program.
 */

/*****
/////          Class Definition          /////
*****/
public class Puzzle {
    /*****
    /////          Method Instantiation          /////
    *****/
    public static void main(String[] args) {
        /*****
        /////          Instantiation          /////
        *****/
        Reader reader = new Reader();
        Tokenizer tokenizer = new Tokenizer();
        PuzzlePrint puzzlePrint = new PuzzlePrint();

        reader.submit("printer", "Device made to waste ink and paper");
        reader.submit("elephant", "Large animal with signature trunk and ears");
        reader.submit("guitar", "Strum it or pluck it to make music");
        reader.submit("umbrella", "Protection from rain");
        reader.submit("computer", "High-speed time wasting calculating machine");
        reader.submit("penguin", "Formally dressed flightless bird");

        ArrayList<String> words = reader.getWords();
        ArrayList<String> clues = reader.getClues();
        ArrayList<String> tokens = tokenizer.getTokens(words);

        puzzlePrint.printTokens(tokens);
        puzzlePrint.printClues(clues);
        puzzlePrint.printAnswerKey(words);
    }
}
```

Reader Class

```
package org.coursera.lab.capstone;

import java.util.ArrayList;

/*
 * Your solution for OOAD Course 1 Capstone Project
 * Kate Pendavinji
 * March 1, 2025
```

```

* Each file highlights a thorough breakdown of the OOAD concepts
* and instructions followed to achieve the results for the puzzle
* game six little words.
*/

/*
 * Create a class called "Reader" with appropriate methods to:
 * o Allow the user to provide a word and a corresponding clue string
 * Ex:
reader.submit("printer", "device made to waste ink and paper")
 * Words should be checked for length (minimum of 5 letters long)
 * Keep an ArrayList of any submitted words and another of corresponding clues
 * Provide a return value of how many words have been submitted
 * o Create and return an ArrayList of the last 6 submitted words
 * Your code should convert each word into uppercase
 * Ex: PRINTER
 * o Create and return an ArrayList of the last 6 submitted clue strings
 * Clue: "Device made to waste ink and paper"
*/

/*****
/////      Class Definition      /////
*****/
public class Reader {
    /*****
    /////      Accessibility:      /////
    /////      Accessed through public methods,      /////
    /////      & Class Attributes      /////
    *****/
    private ArrayList<String> words;
    private ArrayList<String> clues;

    public Reader() {
        words = new ArrayList<>();
        clues = new ArrayList<>();
    }

    // Allow the user to provide a word and a corresponding clue string
    ///// Ex: reader.submit("printer", "device made to waste ink and paper")
    /*****
    /////      Method Definition      /////
    *****/
    int submit(String word, String clue) {
        ///// Words should be checked for length (minimum of 5 letters long)
        if (word.length() >= 5) {
            words.add(word);
            clues.add(clue);
            return words.size();
        } else {
            return words.size();
        }
    }

    /*****
    /////      Method Definition      /////
    *****/
    // Create and return an ArrayList of the last 6 submitted words
    ArrayList<String> getWords() {
        // get last 6 words
        int six = 0;
        // when more than 6 words start @ 6th most recently submitted
        if (words.size() > 6) {
            six = words.size() - 6;
        }
        ///// Your code should convert each word into uppercase
        ArrayList<String> uppercase = new ArrayList<>();
        for (int i = six; i < words.size(); i++) {
            String word = words.get(i);
            uppercase.add(word.toUpperCase());
        }
    }
}

```

```

    }
    return uppercase;
}

// Create and return an ArrayList of the last 6 submitted clue strings
/*****
/////      Method Definition      /////
*****/
ArrayList<String> getClues() {
    // get last 6 words
    int six = 0;
    // when more than 6 words start @ 6th most recently submitted
    if (clues.size() > 0) {
        six = clues.size() - 6;
    }
    // populate arraylist with last 6 submitted clue strings
    ArrayList<String> clueList = new ArrayList<>();
    for (int i = six; i < clues.size(); i++) {
        String theClue = clues.get(i);
        clueList.add(theClue);
    }
    return clueList;
}
}

```

Tokenizer Class

```

package org.coursera.lab.capstone;

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

/*
 * Your solution for OOAD Course 1 Capstone Project
 * Kate Pendavinji
 * March 1, 2025
 * Each file highlights a thorough breakdown of the OOAD concepts
 * and instructions followed to achieve the results for the puzzle
 * game six little words.
 * https://www.geeksforgeeks.org/shuffle-or-randomize-a-list-in-java/
 */

/*
 * Create a class called "Tokenizer" with appropriate methods to:
 * o Break each word from a provided ArrayList into tokens that are two letters
 * long (if the word has an odd length, the last token will be three letters *
 * long) and place the created tokens on a new ArrayList
 * * Example: For input word "PRINTER"; output tokens should be 'PR','IN','TER' *
 * o Create a single ArrayList of all tokens created from all supplied words * o
 * Randomize the order of the tokens in the ArrayList
 */

/****
/////      Class Definition      /////
*****/
public class Tokenizer {
    /****
    /////      Method Definition      /////
    *****/
    ArrayList<String> getTokens(ArrayList<String> words) {
        ArrayList<String> tokens = new ArrayList<>();
        for (int i = 0; i < words.size(); i++) {
            // Break each word from a provided ArrayList into tokens that are two letters
            String word = words.get(i);
            // words 1, 2, or 3 letters long are already in token form
            if (word.length() < 4) {
                tokens.add(word);
            }
        }
    }
}

```

```

    } else {
        if (word.length() % 2 == 0) {
            for (int j = 0; j < word.length(); j += 2) {
                tokens.add(word.substring(j, j + 2));
            }
        }
        // if the word has an odd length, the last token will be three letters
        else {
            for (int j = 0; j < word.length() - 3; j += 2) {
                tokens.add(word.substring(j, j + 2));
            }
            // last token
            tokens.add(word.substring(word.length() - 3));
        }
    }
}
// Randomize the order of the tokens in the ArrayList
Collections.shuffle(tokens);
// Create a single ArrayList of all tokens created from all supplied words
return tokens;
}
}

```

Puzzle Print Class

```

package org.coursera.lab.capstone;

import java.util.ArrayList;

/*
 * Your solution for OOAD Course 1 Capstone Project
 * Kate Pendavinji
 * March 1, 2025
 * Each file highlights a thorough breakdown of the OOAD concepts
 * and instructions followed to achieve the results for the puzzle
 * game six little words.
 */

/*
 * Create a class called "PuzzlePrint" with appropriate methods to:
 * o Print "Tokens", followed by the randomized tokens printed out as a table, 4 * tokens
 * per line, tab delimited (the last line printed may have less than 4 tokens in it) * o
 * Print "Clues", followed by a list of the entered clues
 * o Print "Answer Key", followed by the original words converted to uppercase
 */

/****/
Class Definition
/****/

public class PuzzlePrint {
    // Print "Tokens", followed by the randomized tokens
    /****/
    Accessibility
    /****/
    public void printTokens(ArrayList<String> tokens) {
        System.out.println("Six Little Words");
        System.out.println();
        System.out.println("Tokens");
        // printed out as a table, 4 tokens per line, tab delimited
        for (int i = 0; i < tokens.size(); i++) {
            System.out.print(tokens.get(i) + "\t");
            if ((i + 1) % 4 == 0) {
                System.out.println();
            }
        }
    }
    // the last line printed may have less than 4 tokens in it
}

```

```

        if (tokens.size() % 4 != 0) {
            System.out.println();
        }
        System.out.println();
    }

    // Print "Clues", followed by a list of the entered clues
    public void printClues(ArrayList<String> clues) {
        System.out.println("Clues");
        System.out.println();
        for (int i = 0; i < clues.size(); i++) {
            String clue = clues.get(i);
            System.out.println(clue);
        }
        System.out.println();
    }

    // Print "Answer Key", followed by the original words converted to uppercase
    public void printAnswerKey(ArrayList<String> words) {
        System.out.println("Answer Key");
        System.out.println();
        for (int i = 0; i < words.size(); i++) {
            String word = words.get(i);
            System.out.println(word);
        }
        System.out.println();
    }
}

```

JUnit Testing

```

package org.coursera.lab.capstone;

import static org.junit.jupiter.api.Assertions.assertEquals;

import org.junit.jupiter.api.Test;
import java.util.ArrayList;

/*
 * Your solution for OOAD Course 1 Capstone Project
 * Kate Pendavinji
 * Each file highlights a thorough breakdown of the OOAD concepts
 * and instructions followed to achieve the results for the puzzle
 * game six little words.
 */

/**
 * Visit
 * https://junit.org/junit5/docs/5.4.0/api/org/junit/jupiter/api/Assertions.html
 * for the full list of available methods of assertion types.
 */
public class PuzzleTest {

    @Test
    public void puzzleTest1() {
        Reader r = new Reader();
        r.submit("printer", "Device made to waste ink and paper");
        r.submit("elephant", "Large animal with signature trunk and ears");
        r.submit("guitar", "Strum it or pluck it to make music");
        r.submit("umbrella", "Protection from rain");
        r.submit("computer", "High-speed time wasting calculating machine");
        r.submit("penguin", "Formally dressed flightless bird");
        ArrayList<String> words = r.getWords();
        assertEquals("PENGUIN", words.get(5), "Puzzle test 1 failed (using Reader submit and getWords)");
    }
}

```

```

@Test
public void puzzleTest2() {
    Reader r = new Reader();
    r.submit("printer", "Device made to waste ink and paper");
    r.submit("elephant", "Large animal with signature trunk and ears");
    r.submit("guitar", "Strum it or pluck it to make music");
    r.submit("umbrella", "Protection from rain");
    r.submit("computer", "High-speed time wasting calculating machine");
    r.submit("penguin", "Formally dressed flightless bird");
    ArrayList<String> clues = r.getClues();
    assertEquals("High-speed time wasting calculating machine", clues.get(4),
        "Puzzle test 2 failed (using Reader submit and getClues)");
}

@Test
public void puzzleTest3() {
    Tokenizer t = new Tokenizer();
    ArrayList<String> words = new ArrayList<>();
    words.add("A");
    assertEquals(1, t.getTokens(words).size(), "Puzzle test 3.1 failed (using Tokenizer getTokens)");
    words.add("BC");
    assertEquals(2, t.getTokens(words).size(), "Puzzle test 3.2 failed (using Tokenizer getTokens)");
    words.add("DEF");
    assertEquals(3, t.getTokens(words).size(), "Puzzle test 3.3 failed (using Tokenizer getTokens)");
    words.add("GHIJ");
    assertEquals(5, t.getTokens(words).size(), "Puzzle test 3.4 failed (using Tokenizer getTokens)");
}
}

```

Results:

Six Little Words

Tokens

UM EL IT PE
 CO GU UIN TER
 IN EP NT NG
 BR LA HA UT
 ER AR EL MP
 PR

Clues

Device made to waste ink and paper Large
 animal with signature trunk and ears Strum
 it or pluck it to make music Protection
 from rain
 High-speed time wasting calculating machine
 Formally dressed flightless bird

Answer Key

PRINTER
 ELEPHANT
 GUITAR
 UMBRELLA
 COMPUTER
 PENGUIN

UML Class Diagram

