**AP Computer Science I**

Performance Task

# **Create — Mad Libs**

## **Overview**

In this project, you’ll write a program that generates Mad Lib stories based on user input!

Mad Lib stories are stories that start off as a template, with many holes in the story that need to be filled in, and a person who doesn’t know about the story provides the missing details. The result is often a hilarious nonsensical story.

For example, if our template was the following:

The [Adjective] [Color] fox [Past tense verb] the [Adjective] dog.

And our user input the following for each placeholder:

* quick
* brown
* helped
* lazy

The resulting Mad Lib would be:

The quick brown fox helped the lazy dog.

We’ve provided an example Mad Lib template for you in the file madlib.txt, but feel free to change it to your own!

## **Assessment**

You will be provided with 12 hours of class time to complete and submit the following:

* A video of your program running
* Written responses about your program and development process
* Program Code

Your teacher will share submission guidelines that include suggestions for creating video and PDF files.

## **General Requirements**

You are required to:

* Iteratively design, implement, and test your program.
* Independently create at least one significant part of your program.
* Create a video that displays the running of your program and demonstrates its functionality.
* Write responses to questions about your program.
* Include your entire program code.

## **Program Requirements**

Your program must demonstrate:

* Students are able to create a new class and attempt to create a constructor with the same name as the class, but incorrectly include a return type for the constructor
* Students are able to write program code to declare instance variables and write headers for constructors with or without parameters, but fail to assign value to the instance variables in the constructor. Often they unknowingly create local variables in the constructor with the same name as the instance variables instead.
* Students are able to use variables that store a single value, but can’t understand how related values can be stored and accessed together using a single variable name.
* Students are able to represent related data using an array or ArrayList, but can’t write program code that requires them to manipulate the elements while traversing the array or ArrayList.
* Students are able to write program code using some of the data available, but students often are unaware of what variables (instance variables/ parameters) they have available to solve a problem and the associated data that can be accessed by calling the methods of reference data.
* Students are able to traverse and manipulate elements in a single String, 1D array, 2D array or ArrayList object, but can’t write program code to perform parallel/simultaneous traversals of two or more structures.

## **Submission Requirements**

### 1. **Video**

Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. Your video must not exceed 1 minute in length and must not exceed 30MB in size.

### 2**. Written Responses**

Submit one PDF document in which you respond directly to each prompt. Clearly label your responses 2a – 2e in order. Your response to all prompts combined must not exceed 750 words, exclusive of the Program Code.

## **Program Purpose and Development**

1. Provide a written response or audio narration in your video that:

Identifies the programming language.

* Identifies the purpose of your program.
* Explains what the video illustrates.

(Approximately 150 words)

1. Why did you choose to implement this program? Describe the development process used in the completion of the project.

(Approximately 200 words)

1. Data structures such as arrays and ArrayLists are used to organize data and simplify data processing. Discuss one data structure that you used in the completion of this lab. How did it help in the processing of data?

(Approximately 200 words)

1. Would you have been able to complete this lab without the use of a data structure? Why or why not?

(Approximately 200 words)

1. Capture and paste your entire program code into the PDF.

* Include comments or citations for program code that has been written by someone else.

## Tasks

### **Activity 1 - Explore**

**Description**

Students will explore the program Mad Libs and plan out how to break down the program in groups or with partners. They will identify what they might need to know in order to complete the project.

* Mad Libs Demo
* Plan out the project

Time To Complete: 1 Hours

### **Activity 2 - Research**

**Description**

In this lesson, students will explore and learn the anatomy of classes. They will take a deeper dive into what the access specifier does, and how it can be used within programs to make data public or private. Students will learn about encapsulation and the responsibility programmers have to choose whether data should be accessible, modifiable, both or neither. Students will learn about and create arrays. The use of array objects allows multiple related items to be represented using a single variable. Students will learn about and use ArrayLists. ArrayLists are similar to arrays, except that they are a mutable list of object references. ArrayLists provide a convenient way to create adjustable arrays.

* Unit 5 - Writing Classes
* Unit 6 - Arrays

Time To Complete: 14 Hours

### **Activity 3 - Ideate**

**Description**

You will be creating this program piece by piece over the next several challenges.

Your job in this challenge is to collect all the placeholders in the template (ie "[Adjective]" and "[Color]") and store them in an ArrayList.

Write the following method:

/\*\*

\* Returns an ArrayList holding every placeholder in the template,

\* in the order they appear.

\* Placeholders will be stored in the form "[Adjective]",

\* including the "[ ]".

\* If the template is malformed (ex: not every placeholder has

\* a closing character), this method returns null.

\*/

private ArrayList<String> getPlaceholders(String template)

For example, the following code:

String template = "The [Adjective] [Noun] jumped over the [Adjective] [Noun].";

ArrayList<String> placeholders = getPlaceholders(template);

for(int i = 0; i < placeholders.size(); i++)

{

System.out.println(placeholders.get(i));

}

Should output:

[Adjective]

[Noun]

[Adjective]

[Noun]

Now that we’ve collected all the placeholders in the template, it’s time to get input from the user so we know what each placeholder should be replaced with! We’re going to allow for the user to fill in the holes in the story.

In this challenge, you will be asking the user to input a word or phrase for each placeholder in the template, and storing these responses for later.

For example, if the template looks like:

The [Adjective] [Color] fox [Past tense verb] the [Adjective] dog.

You will prompt the user once for each placeholder:

Enter a Adjective: (user inputs something here)

Enter a Color: (user inputs something here)

Enter a Past tense verb: (user inputs something here)

Enter a Adjective: (user inputs something here)

And store each of these responses, in the same order as the corresponding placeholders.

* Unit 7
* Collecting Placeholders
* Getting User Input

Time To Complete: 8 Hours

### **Activity 4 - Evaluate**

**Description**

Students will review and test each other’s code with test cases and provide feedback to the creator on how they might improve their program or to OK the program to move onto the next phase.

Now it’s time to write the code that will replace a given placeholder with a given replacement String. Writing this method will make it much easier to go through and replace every placeholder in the template.

Write the following method:

/\*\*

\* Replaces the first instance of the String placeholder

\* inside the String template with the String replacement.

\* Returns a new String that is the result of this replacement.

\* If the placeholder does not exist in the template,

\* then the template passed in is returned, no replacement

\* is made.

\*/

private String replacePlaceholder(String template, String placeholder, String replacement)

For example, the following code:

String template = "The [Adjective] [Noun] jumped over the [Adjective] [Noun].";

String replaced = replacePlaceholder(template, "[Adjective]", "silly");

System.out.println(template);

System.out.println(replaced);

Should print out:

The [Adjective] [Noun] jumped over the [Adjective] [Noun].

The silly [Noun] jumped over the [Adjective] [Noun].

* Replacing Placeholders

Time To Complete: 1 Hours

### **Activity 5 - Construct a Prototype**

**Description**

Now we have every placeholder in the template, each corresponding replacement, and a way to replace a given placeholder with a given replacement. Great job so far!

Now it’s time to put all the pieces together. In this challenge you will generate the final Mad Lib by replacing each placeholder in the placeholder ArrayList with each corresponding replacement in the replacement ArrayList.

Output the final Mad Lib to the user once it is generated.

You may find it helpful to implement the following method:

/\*\*

\* Asks the user to input replacement Strings

\* for each placeholder in the template.

\* Replaces each placeholder with the corresponding replacement.

\* Returns the resulting Madlib.

\*/

private String replaceAllPlaceholders(String template)

Make sure your final program is robust. Make sure it can handle incorrect input without breaking. How does your program deal with a malformed madlib.txt template? A template with no placeholders? A template with only placeholders? Think about other edge cases you might need to account for.

* Generating the Mad Lib
* Writing your own Mad Lib

Time To Complete: 3-5 Hours

### **Activity 6 - Improve the Design**

**Description**

Make any extensions you’d like to your Mad Libs program.

This is a blank program for you to experiment and add any features you think would be interesting.

Possible Extensions

* Print a / an based on whether the following word starts with a vowel. (ie “Enter an Adjective” instead of “Enter a Adjective”)
* Capitalize the replacement word if it starts a sentence.

Allow for placeholders to be repeatable throughout the story to make the story more consistent. For example the template could look like:  
[Main Character, REPEAT] ate a [Food].

* [Main Character, REPEAT] was very [Emotion].  
  And the user would only have to input the Main Character once for it to be used throughout the story.
* Anything else you want!

Have another student review your work and provide feedback for your additions.

* Open Ended Activity

Time To Complete: 3-5 Hours

### **Activity 7 - Share Solutions**

**Description**

Students share their solutions and how they differentiate from the original design.

Time To Complete: 1-3 Hours

### **Activity 8 - Reflect**

**Description**

Students answer the reflections questions and submit all evidence

Time To Complete: 1 Hour