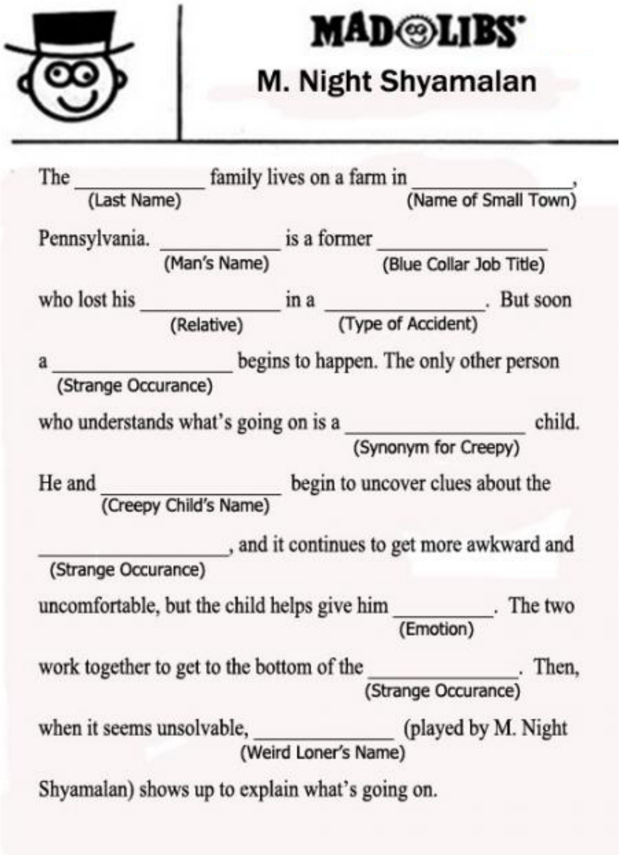
Assignment 01

CS 120: Software Design I

Program 1: `MadLibs`

Mad Libs[1] is a game developed in the 1950s. Traditionally, it is played between two or more players. One person, the *storyteller*, has a short story in which a handful of the words have been replaced with blanks. The storyteller then asks the remaining players for suggestions of words that fulfill particular categories (e.g., adjective, proper noun), but does not tell the players what the short story is about. Once all of the blanks have been filled in, the storyteller reads the constructed story aloud, using the suggested words in the place of the blanks. You can see an example of a blank Mad Lib below:



MAD LIBS

M. Night Shyamalan

The ____________ family lives on a farm in ________________,
(Last Name) (Name of Small Town)

Pennsylvania. ____________ is a former ________________
(Man's Name) (Blue Collar Job Title)

who lost his ______________ in a ________________. But soon
(Relative) (Type of Accident)

a ________________ begins to happen. The only other person
(Strange Occurance)

who understands what's going on is a ________________ child.
(Synonym for Creepy)

He and ________________ begin to uncover clues about the
(Creepy Child's Name)

__________________, and it continues to get more awkward and
(Strange Occurance)

uncomfortable, but the child helps give him _________. The two
(Emotion)

work together to get to the bottom of the ______________. Then,
(Strange Occurance)

when it seems unsolvable, ______________ (played by M. Night
(Weird Loner's Name)

Shyamalan) shows up to explain what's going on.

[1] More info on Mad Libs: `https://en.wikipedia.org/wiki/Mad_Libs`

For your first program, you will be writing a Mad Libs program that takes the place of the storyteller and asks a human player to fill in the words in the story below:

_____________ _____________ arrived on UWL's campus after break and immediately
first name (1) *last name (2)*

realized their _____________ was missing. They went to investigate, checking both
noun (3)

_____________ and _____________. _____________ knew they were getting close when
place name (4) *place name (5)* *first name (1)*

a _____________ person in a _____________ hat handed over an envelope addressed
syn. for shifty (6) *adjective (7)*

to "_____________ _____________." Inside were a set of cryptic instructions. "This
first initial (1) *last name (2)*

makes no sense!" _____________ lamented. "How will I ever find my _____________?"
first name (1) *noun (3)*

"It's _____________ my dear _____________!" the person with the _____________ hat
adjective (8) *first name (1)* *adjective (7)*

exclaimed. "No, it isn't." _____________ said sadly, and they turned around to leave.
first name (1)

"Wait! _____________-!" the person with the _____________ hat called out, but by
first three letters of first name (1) *adjective (7)*

then _____________ had already left and was ordering a new _____________ on
first name (1) *noun (3)*

_____________ from their smartphone.
online storefront (9)

Your program will be written in a class called `MadLibs`. Your program will start by prompting the player for all nine words necessary to fill in the Mad Lib. Note that you will use some of the words multiple times, but you will only ask the player for that word once; the blanks above have been annotated with numbers to help illustrate reuse. Additionally, you might need to modify a provided word for the output in some cases (e.g., the initial of the person's first name plus a period). Once the user has provided all nine words, the program will display the completed Mad Lib. For the purposes of this program, you can assume that the first name provided by the user will always have a length of at least three. Additionally, any input the user gives to a prompt might contain more than one word (e.g., see **Example 1**). Finally, I recommend retyping the prompts and Mad Lib from below, as some characters (e.g., quotation marks) do not copy correctly into Eclipse.

Below are two example runs of the program with user input `Highlighted`. Note that in the version you will display, you should print one sentence of the Mad Lib per line; due to space constraints, some of the sentences are wrapped in the examples in this document.

Example 1

```
Enter a first name: Allie
Enter a last name: Sauppe
Enter a noun: alarm clock
Enter a proper place name (e.g., "Centennial Hall"): The Root Note
Enter another proper place name (e.g., "Centennial Hall"): Wing
Enter a synonym for shifty: wily
Enter an adjective: bedazzled
Enter another adjective: fuzzy
Enter the name of an online storefront (e.g., "Amazon"): ebay

Allie Sauppe arrived on UWL’s campus after break and immediately realized
   their alarm clock was missing.
They went to investigate, checking both The Root Note and Wing.
Allie knew they were getting close when a wily person in a bedazzled hat
   handed over an envelope addressed to "A. Sauppe."
Inside were a set of cryptic instructions.
"This makes no sense!" Allie lamented. "How will I ever find my alarm clock?"
"It’s fuzzy my dear Allie!" the person with the bedazzled hat exclaimed.
"No, it isn’t." Allie said sadly, and they turned around to leave.
"Wait! All-!" the person with the bedazzled hat called out, but by then Allie
   had already left and was ordering a new alarm clock on ebay from their
   smartphone.
```

Example 2

```
Enter a first name: AAA
Enter a last name: BBB
Enter a noun: CCC
Enter a proper place name (e.g., "Centennial Hall"): DDD
Enter another proper place name (e.g., "Centennial Hall"): EEE
Enter a synonym for shifty: FFF
Enter an adjective: GGG
Enter another adjective: HHH
Enter the name of an online storefront (e.g., "Amazon"): III

AAA BBB arrived on UWL’s campus after break and immediately realized their CCC
    was missing.
They went to investigate, checking both DDD and EEE.
AAA knew they were getting close when a FFF person in a GGG hat handed over an
    envelope addressed to "A. BBB."
Inside were a set of cryptic instructions.
"This makes no sense!" AAA lamented. "How will I ever find my CCC?"
"It’s HHH my dear AAA!" the person with the GGG hat exclaimed.
"No, it isn’t." AAA said sadly, and they turned around to leave.
"Wait! AAA-!" the person with the GGG hat called out, but by then AAA had
   already left and was ordering a new CCC on III from their smartphone.
```

Your Program

Some prompts and questions to consider *before* starting to write the program:

- How can you save the user's input for each prompt?
- How can you read in more than one word for a single prompt?
- How can you make modifications to some of the words to use without overwriting the original value? (e.g., getting the first initial of the first name such that you still have access to the whole first name)
- What order will the prompts occur in? (e.g., prompting for first name before last name?)
- How will you ensure that the prompt and user input are on the same line?
- How will you ensure that there is a space between the prompt and the user input?
- How will you print a blank line?
- How will you add quotation marks in the output?

The following is a high level checklist of requirements for your program:

- □ Your class is named `MadLibs` and is placed in the project created for this assignment
- □ Your code is commented according to the guidelines in the Java Style Guide found on D2L
- □ Your code is formatted according to the guidelines in the Java Style Guide found on D2L
- □ Your code fulfills the functionality outlined above
- □ You have all the prompts listed in the examples in the exact same order
- □ You have a blank line between the prompts and the displayed Mad Lib
- □ You have a line break between each sentence of the displayed Mad Lib
- □ Your strings (both prompts and output) are formatted **exactly** as shown in the examples

Program 2: `TongueTwister`

The second program you will be writing will be in a class named **`TongueTwister`**. Tongue twisters[2] often contain the same sound multiple times, such that rapid repetition of the tongue twister is likely to result in mispronunciation.

Your program should start by printing a prompt to the user, asking them to input a tongue twister. Your program should then print a second prompt, asking the user to input a sound they want to replace in their tongue twister, followed by a third prompt asking what sound they would like to replace it with. Finally, your program should print out the original tongue twister, the length of the original tongue twister, the new tongue twister with the replacement provided by the user, and the length of the new tongue twister.

Below are three example runs of the program with user input `Highlighted`. Note that, due to space constraints, some of the sentences are wrapped in the examples in this document.

Example 1

```
Please enter a tongue twister: I scream you scream we all scream for ice cream
Please enter the sound you want to replace: cream
Please enter the sound you want to replace "cream" with: milk

Your original tongue twister: "I scream you scream we all scream for ice cream
    " (length: 47)
Your new tongue twister: "I smilk you smilk we all smilk for ice milk" (length
    : 43)
```

Example 2

```
Please enter a tongue twister: Betty Botter bought a bit of butter
Please enter the sound you want to replace: tt
Please enter the sound you want to replace "tt" with: nn

Your original tongue twister: "Betty Botter bought a bit of butter" (length:
    35)
Your new tongue twister: "Benny Bonner bought a bit of bunner" (length: 35)
```

Example 3

```
Please enter a tongue twister: She sells sea-shells on the sea-shore
Please enter the sound you want to replace: zzz
Please enter the sound you want to replace "zzz" with: ll

Your original tongue twister: "She sells sea-shells on the sea-shore" (length:
     37)
Your new tongue twister: "She sells sea-shells on the sea-shore" (length: 37)
```

[2] Read more: `https://en.wikipedia.org/wiki/Tongue-twister`

Your Program

Some prompts and questions to consider *before* starting to write the program:

- How can you save the user's input for each prompt?
- How can you produce a new `String` with the replacement?
- What order will the prompts occur in? (e.g., prompting for the sound to replace before the replacement?)
- How will you ensure that the prompt and user input are on the same line?
- How will you ensure that there is a space between the prompt and the user input?
- How will you print a blank line?
- How will you add quotation marks in the output?

The following is a high level checklist of requirements for your program:

- □ Your class is named `TongueTwister` and is placed in the project created for this assignment
- □ Your code is commented according to the guidelines in the Java Style Guide found on D2L
- □ Your code is formatted according to the guidelines in the Java Style Guide found on D2L
- □ Your code fulfills the functionality outlined above
- □ You have all the prompts listed in the examples in the exact same order
- □ You have a blank line between the prompts and the output
- □ You have a line break between each sentence of the output
- □ Your strings (both prompts and output) are formatted **exactly** as shown in the examples