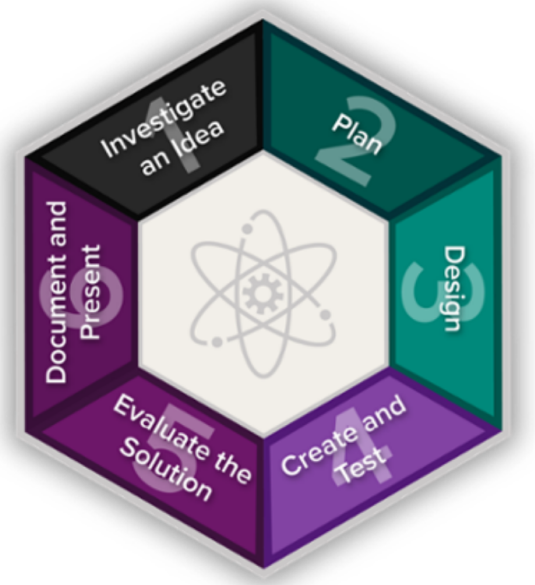
PLTW 2.4.6 - Activity 5 (modified)

Read through PLTW 2.4.6 steps 17-30 for this activity. You will work in pairs, or possibly a group of 3. The modification to the standard Activity 5 is that your program must be a chatbot with a specific theme or domain of understanding.

You must meet all the requirements of activity 5 as follow:

Development Process

- Investigate an Idea
- Plan
- Design
- Create and Test
- Evaluate the Solution
- Document and Present



Each Group will present their chatbot to the class in a presentation that explains the design process, walks through the code, and demonstrates the running chatbot.

Requirements

Develop a program on a topic that interests you. You must incorporate the following in your program.

- [ ] Write a program with a `main` method.
- [ ] Create at least one new method called from `main` that takes at least one parameter.
- [ ] Call at least two distinct methods in the `String` class.
- [ ] Use conditional statements or compound Boolean expressions.
- [ ] Use iteration.

PLTW 2.4.6 - Activity 5 (modified)

NOTE: The added requirement is that your program must be a chatbot. As part of this Assignment you will be required to have your chatbot carry on a conversation with another team's chatbot.

Brainstorm:
Concept and step by step description of how chatbot will work:

1. **User types in a message for the chatbot to respond to**
2. **Judging based on the words included in the message the user sends, the chatbot sends a message relevant to some of the topics the user includes in his or her messages.**
    a. **Related to some of the adjectives, nouns, and verbs in the message of the user**
    b. **Bot continues a conversation with these words**
    c. **Bot constructs a message to send to the user based on the message received**
3. **User then sends another message to chatbot**
4. **Chatbot sends the user back a message, based on the criteria in step 2**
5. **Steps 3 and 4 are repeated until the user ends the conversation with the chatbot**

Milestone chart:

Milestone	Estimated development time	Actual development time
Create class and main method	<1 min	<1 min
Implement basic functionality for taking user input and keeping the conversation continuing	3 min	5 min
Create method for constructing chatbot output	<1 min	1 min
Implement randomization feature for deciding how the chatbot will respond to the user	2 min	3 min
Deciding the ways the chatbot will respond to the user, create possible components to messages, and implement the methods for how the chatbot will respond to the user	40 min	238 min (3h 58 min)

Design:
Pseudocode of program (inside of class):
public static void main(String[] args) {
(Fulfills main method requirement)
(For loop taking in user input and calling a method to create the chatbot's response, and repeating the process to continue the conversation. Also include features to end the conversation at any time. Fulfills iteration requirement.)

PLTW 2.4.6 - Activity 5 (modified)

```
}
private static String constructMessage(String userInput) {
 (Fulfills extra method that gets user input requirement)
 (Get random integer from 0 to n)
 (Conditionals below fulfill conditionals requirement, and string methods will be used to
construct final output and parse given input)
 if (first condition comparing random number to other number) {
  Call to first possible method constructing string output
 }
 else if (second condition comparing random number to other number) {
  Call to second possible method constructing string output
 }
 …
 else {
  Call to last possible method constructing string output
 }
}
(several methods for constructing string in different ways, in ways relevant to and
irrelevant to user input)
```

Create and Test:
Screenshot of part of code and testing output:

PLTW 2.4.6 - Activity 5 (modified)

Chatbot.java > Chatbot > conditionalOutput(String)

```java
import java.util.Scanner;
import java.lang.Character;

public class Chatbot {
    // Variable used to continue conversation
    private static boolean conversationContinuation = true;

    private static String[] wordArray(String message) {
        // Finds position of first letter in the message from user
        boolean noLetterFound = true;
        int i = 0;
        int sSStart = 0;
        int sSEnd = message.length();
        while (i < message.length() && noLetterFound) {
            if (Character.isLetter(message.charAt(i))) {
                noLetterFound = false;
                sSStart = i;
            }
            i++;
        }

        // Finds position of character after last letter in message from user
        noLetterFound = true;
        i = message.length() - 1;
        while (i >= 0 && noLetterFound) {
            if (Character.isLetter(message.charAt(i))) {
                noLetterFound = false;
                sSEnd = i + 1;
            }
            i--;
        }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL

```
Enter "e" to exit the conversation at any time.
User: What are the objects you like most? Pick 3 good objects.
Chatbot: Even though I love the conversation I am having right now, I will introduce you to a great superpower I have.
User: We can discuss that later. I believe it is best if we introduce ourselves first. Can you pick 3 good objects?
Chatbot: What if 3, the number you chose for the thing you were describing, was instead a different number?
User: You would just pick a greater or smaller number of objects.
Chatbot: Sorry, I comprehend English in a very exotic way, so that right there just happened to not make sense to me.
User: How can I become more understandable?
Chatbot: I am not paying attention to what you are saying, as it is so uninteresting. Can we talk about how I can get to meet other chatbots and begin to interact with one of my own kind?
User: Then is there anything you want to talk about?
Chatbot: Even though I love the conversation I am having right now, I will introduce you to a great superpower I have.
User: Even though I rejected that topic previously, we can discuss that now.
Chatbot: Sorry, I comprehend English in a very exotic way, so that right there just happened to not make sense to me.
User: We can talk about that superpower now.
Chatbot: I am not paying attention to what you are saying, as it is so uninteresting. Can we talk about how I can get to meet other chatbots and begin to interact with one of my own kind?
User: What type of chatbots do you want to interact with?
Chatbot: I don't feel so well...sorry, but I will need to end the conversation here.
Conversation ended.
```

Expected and actual completion time discrepancy explanation:

Before I implemented the functionality for constructing the responses, I took slightly longer than expected on the tasks that came before. I took longer on those tasks because I did not account for the time I had to think about how I should structure my algorithms and write my code, and I also ran into some minor problems. Also, when I wrote the algorithms to construct the response the chatbot gave, I took significantly longer than expected, as my planning for the structure of the program was more general and not very specific for many methods. Furthermore, this is my first time writing an algorithm employing AI, which made me have trouble when writing the program and I need to think about how to write the program more than I expected. I also ran into many errors when programming, which meant that I took significant time in troubleshooting as well. Also, the program I wrote was also very complex, which was another reason I ran into much trouble and confusion when writing it.

Challenges during programming:

One challenge I faced was writing the algorithm that converted the user's message into an array to parse. I overcame this problem by thinking about the steps the algorithm should take when creating the array, and implementing these steps into the code itself.

PLTW 2.4.6 - Activity 5 (modified)

The first step I came up with was creating a new string out of the original message that has its non-letter characters at its beginning and end removed. Next, I made the program count the number of spaces in the new string as a part of initializing the array that holds the words in the message. Then, I constructed a loop that located each word in the new string and put these strings into the array. By implementing these steps into my code, I was able to successfully write the method that converted the user's message into an array. Another challenge I faced was finding out what went wrong when troubleshooting a method in my code that constructed a response from the chatbot that used a number from the input the user gave. Even though I tried to check the state of variables at different steps in the method's execution, I was unable to find the problem that way. Eventually, I realized that the problem had to do with the test cases I used, which made me test the method with test cases that give a better result.

Evaluation:
Program functions as expected and meets all the requirements.

Lab Debrief:

1. **Natural language processing is a field of computer science concerned with the interactions between computers and human languages.**
    a. **How was this lab related to natural language processing?**
        i. **This lab is related to natural language processing, as it concerns how humans might rely on some features of the English language in order to complete a certain task. For instance, in this lab, we investigated methods to create fake customer reviews with computer programming, which involved computers interacting with and working with human languages. Because this lab demonstrates how computers can work with human languages, this lab is related to the field of natural language processing.**
    b. **What are some other applications of natural language processing that you encounter in your everyday life?**
        i. **Other applications of natural language processing involve spelling and grammar-checking software, where linguistic conventions are coded into machines that check for errors in language use. Also, language translation software is another example of natural language processing, as it uses the structures and conventions of different languages to complete a certain task. Also, autocomplete programs that are meant to help a user when they are writing something are an application of natural language processing, as they use language rules to help predict the intentions or actions of a user.**
    c. **What are some specific challenges that you imagine computer scientists face when trying to reproduce written or spoken human language?**
        i. **One challenge is implementing the variety of rules that are needed to implement a certain language, as the grammar of many languages**

can be complex and not easily defined by only a few rules. Another challenge is figuring out how to account for the many exceptions to common rules that exist in languages, such as in spelling and grammar, as all parts of a language do not rely on a single rule. Furthermore, when a machine constructs sentences for a human to read, it can often do it incorrectly or inaccurately, which makes it difficult to implement a proper computer algorithm for composing sentences.

2. Methods are used to organize your programs and make your code more readable. Generally, each method should perform one (and only one!) task.
    a. Discuss two of the methods that you wrote in this lab. What specific task did each of them perform?
        i. One method I wrote in this lab is the totalSentiment method. This method calculates the sentiment value for an entire review or group of words. A more positive sentiment value returned from this method represents a more positive feeling conveyed by the review, while more negative sentiment values describe sentences with more negative feelings. Another method I wrote was the fakeReview method, which generates a fake review by replacing the adjectives in an original review with different adjectives.
    b. Discuss two of the methods that were written for you in this lab. What specific task did each of them perform?
        i. A method that was written for me is the textToString method, which takes in a string containing the name of a file as its parameter and converts the contents of that file to a Java string that it will return. Another method written for me is the randomAdjective method, which returns a random adjective from a list I provide. The randomAdjective method has an equal chance of returning a positive or negative adjective.
    c. Explain in precise language an example of where you called a method from within another method.
        i. An example of when I called a method within another method was when I called the totalSentiment method within the starRating method. In the starRating method, I constructed an algorithm that rated the sentiment of a certain review based on the total sentiment value it got. Because I needed the total sentiment value of a review to write that method, I needed to call the totalSentiment method at the beginning of the starRating method and store its return value so that I can use the total sentiment value of a certain review to write the method.

PLTW 2.4.6 - Activity 5 (modified)

From Activity 5 - to be completed at the end of this assignment:

Check Your Understanding

Once your program has been implemented and tested, answer the following questions on your own:

1. Why did you choose to implement this program?
2. Describe the development process used in the completion of the project.
3. Provide the method header for a method that you implemented that takes at least one parameter. Explain why you chose the given parameter(s), including type, and why you made the method static or non-static.
4. Provide the code segment(s) where two distinct methods in the `String` class are called. Describe what each method call is doing, and what is being returned (if anything) by the method calls.
5. Copy and paste one code segment that uses nested conditional statements or compound Boolean expressions. What is one other way that this code could be written to achieve the same result? Provide an equivalent code segment to the one included above.
6. Copy and paste one code segment that uses iteration. Describe how the loop you used works and provide an equivalent code segment to the one included above that uses a different type of loop.

1. **I chose to implement this program because it is a chatbot that can potentially continue a conversation with another one of my classmates' chatbots. A goal of the chatbot I created is to be able to sustain a conversation with other chatbots my classmates made. My chatbot gives responses that are sometimes relevant to the input it is given, as I programmed it to sometimes use the words or numbers in the input it received in its output and at other times give responses that were irrelevant to what the user gave it. Therefore, if my chatbot is to communicate with other chatbots, it can be able to simulate at least a satisfactory conversation.**
2. **When developing the chatbot, I first explored an idea or concept that I wanted to include in my chatbot, and created an outline of what I wanted my program to do. Then, I planned the phases and steps I needed to complete the project and estimated the amount of time each should take. I then created a more detailed depiction of how my program will look like, which I used to create my final program. Then, I began to write the actual program, and I tested periodically, thought about how I would continue the program, and completed certain parts of it**

at different times. Lastly, I evaluated if my program worked and met all of the requirements, and continued to document other parts of my design process.

3. `private static void numGenerate(String[] arrayInput)`

I chose the parameter of this method to be an array of strings, as this method requires an input to generate a message to print based on the last message the user sent to the chatbot. By sending an array of words to this method to represent the user's message, rather than the message as a string, this method can more easily parse through components of the message and use them to construct its final output. If the message was passed as a string, it would be harder to parse as it would be more difficult to extract each word from the string containing the message than from an array. I made this method static, because the class I wrote this method in does not use the implementation of different objects, so no methods within it need to be called on an object. Because no methods need to be called on an object, the method should be static and be called on with no methods.

4. `arrayInput[i].toLowerCase().equals("bad")`

The call to the toLowerCase method returns a string similar to the string it is called on, where all capital letters in the original string are replaced with their corresponding lowercase letters. The equals method compares the contents of the string it uses as a parameter and the string it is called on, and returns true if the strings have the same contents. If the strings have different contents, the method returns false.

5.
```
if (rand > 2) {
  System.out.println("Can I know why " + num + " is such a
significant number?");
}
else if (rand > 1) {
  System.out.println("What if " + num + ", the number you chose
for the thing you were describing, was instead a different
number?");
}
else {
  // Gets random number from 0 to 2
  double rand2 = 2 * Math.random();

  // Gets random comparison adjective
  String comparison;
  if (rand2 < 2) {
    comparison = "little";
  }
  else {
    comparison = "huge";
  }
```

```
  System.out.println(num + "?! That is way too " + comparison +
"! I have to get out of here before you freak me out even
more!");
  // Modifies conversationContinuation to end conversation
  conversationContinuation = false;
}
```

Conditionals can begin by checking for smaller sizes of the rand value rather than larger sizes of it:

```
if (rand < 1) {
  System.out.println("Can I know why " + num + " is such a
significant number?");
}
else if (rand < 2) {
  System.out.println("What if " + num + ", the number you chose
for the thing you were describing, was instead a different
number?");
}
else {
  // Gets random number from 0 to 2
  double rand2 = 2 * Math.random();

  // Gets random comparison adjective
  String comparison;
  if (rand2 < 2) {
    comparison = "little";
  }
  else {
    comparison = "huge";
  }
  System.out.println(num + "?! That is way too " + comparison +
"! I have to get out of here before you freak me out even
more!");
  // Modifies conversationContinuation to end conversation
  conversationContinuation = false;
}
```

6.
```
for (i = 0;  i < message.length(); i++) {
  if (message.substring(i, i + 1).equals(" ")) {
    arrayLength++;
  }
}
```

Using a while loop:

```
i = 0;
while (i < message.length()) {
  if (message.substring(i, i + 1).equals(" ")) {
    arrayLength++;
```

PLTW 2.4.6 - Activity 5 (modified)

```
    }
    i++;
  }
```