

ITSC 1212 Module 6 Lab

In this lab students work with the String class and conditionals in a program that acts as a chatbot. A chatbot is a computer program that tries to hold a conversation with a user.

This lab derives from the published AP CS A Curriculum - you can find the original Student Guide here: [magpie-lab-student-guide.pdf \(collegeboard.org\)](https://collegeboard.org/magpie-lab-student-guide.pdf) but be aware that there are slight differences in that lab and this one.

Concepts covered in this lab:

- Strings
- Conditionals

Required files or links

- Magpie.java
- MagpieDriver.java

Part A: Setup and understand the chatbot

Familiarize yourself with the two classes we'll study in this lab.

1. Start by downloading the Magpie.java and MagpieDriver.java files found on the Canvas page for this lab and copying them into your VSCode project folder. We'll use this as a starting point for this lab.
2. Run the MagpieDriver program. It displays "Hello, let's talk" then waits on the next line for your input. Enter a phrase and the program responds based on what you entered. You can continue entering phrases and seeing the response as long as you like. When you're finished, enter "Bye" and the program will stop.

We haven't talked about while loops yet but the code at line 24 says "As long as the user enters something other than the String "Bye", execute the loop block (lines 25 thru 27)". That block calls the `getResponse` method of the Magpie class, prints a response based on what the user entered, then calls the Scanner class `nextLine` method to get another response from the student and assigns that String value to the variable named `statement`. This loop repeats until the user enters "Bye" when the program exits. This loop is simply to get a response to the user's entry then get another entry from the user and continue until the user enters a response of "Bye."

ITSC 1212 Module 6 Lab

This file is just for running and controlling the execution of the program; you do not have to modify it. All your work will be done in the Magpie class file.

3. Now examine the Magpie class and answer the following:
 - a. How many constructors does this class have? _____
 - b. How many fields does this class have? _____
 - c. How many methods does this class have? _____
 - d. What does the method getGreeting() return? _____

4. The getResponse method looks complicated but you can evaluate it if you go through it carefully line-by-line. Line 21 looks to see if there is the String “no” anywhere in what the user entered. If so, it sets the response String variable to the value “Why so negative?”. If “no” isn’t found the program looks to see if the String “mother” is present. If so, the program sets the response variable to “Tell me more about your mother.” If that String is not in the user’s entry, the program calls the getRandomResponse method that returns a random value.

5. Now review the getRandomResponse method. Lines 37 thru 40 set up some variables:
 - a. NUMBER_OF_RESPONSES is how many responses will be generated by the code at line 39.
 - b. The double r is a random number from zero to not quite one.
 - c. The int whichResponse is a value computed from the random number and the NUMBER_OF_RESPONSES variable. It will have a value of 0, 1, 2, or 3 (remember why?).
 - d. The String response (line 40) is set to null now but will be changed in the if statement below.

The rest, lines 42 thru 53, is a collection of **if** statements that cover the possible number of random responses we want the chatbot to handle. For each value of whichResponse, the String response is set to a specific String. Finally, the method returns the response String.

6. Once you are familiar with these two classes, run the MagpieDriver class to answer the following questions:
 - a. What is the chatbot response when you enter the word **no**?

 - b. What is the chatbot response when you enter the word **mother**?

 - c. What is the chatbot response when you do not enter any words?

ITSC 1212 Module 6 Lab

- d. What happens when a keyword is included in another word? What is the chatbot response when you enter the word **another** or **I know all the state capitals**, or **I like vegetables smothered in cheese** ? Explain the problem with the responses to these statements.
-
-

7. Continue interacting with the chatbot trying out various combinations of words. Try to guess what the response will be and compare that to what the chatbot returns. Understanding why the responses are different than what you expected will help you develop a good understanding of the logic in this program.
8. Show your answers to the questions in this part to your instructor or a TA to receive credit for this part and proceed to Part B.

Part B: Enhancing the chatbot

In the part we want to make alterations to the `getResponse` method of the `Magpie` class to be able to respond to some other words if they happen to appear in what the user inputs.

1. Alter the existing code so it responds favorably when it sees the name of your teacher. Be sure to use appropriate pronouns! For example, a possible statement and response would be:

Statement: **Mr. Finkelstein is telling us about robotics.**

Response: **He sounds like a good teacher.**

2. Now, have the code check that the user did not just hit enter or a bunch of spaces by checking that the statement (i.e., the statement variable) has at least one character. You can do this by using the `trim()` method of the `String` class to remove spaces from the beginning and end, then checking the length of the trimmed string (For example, the `String` " apple " when trimmed would be "apple". Remember what you have to do when you modify a `String` to save the modified version. Hint: `Strings` are immutable!). If there are no characters, the response should tell the user to enter something. For example, a possible statement and response would be:

Statement:

Response: **Say something, please.**

ITSC 1212 Module 6 Lab

3. Pick three more keywords, (other than “no”, and “mother” which have already been used) and edit the `getResponse` method to respond to each of these. Be careful how you order these and make sure you add them before the return statement.
4. When you are satisfied with your code, show it to your instructor or a TA to receive credit for this section.

Part C: Adding more random responses

In this part we will make alterations to the `getRandomResponse` method of the `Magpie` class to have more non committal responses.

1. Add two more non-committal responses to the possible random responses. Make sure you change the `whichResponse` calculation so these possibilities can occur.
2. When you are satisfied with your code, show it to your instructor or a TA to receive credit for this section.

Part D: Dealing with more than one keyword

1. What is the chatbot response when you enter the sentence **I can't say no to my mother**?

2. What happens when more than one keyword appears in your response?

3. Try the input **My mother has a dog but no cat**. Which response did you get – was it the one about mother or the one about pets or the negative one for no?

4. What if we wanted to have the chatbot always use the response for the word mother even if other keywords are present as well in the user input. There are multiple ways of structuring this decision process. We can have the chatbot first check for the keyword mother and only if it does not find it then branches off to evaluate other keywords using a sequence of if statements (similar to what we have).

ITSC 1212 Module 6 Lab

```
if (condition){
    ...
}else{
    if (condition){
        ...
    }
    if (condition){
        ...
    }
    ...
}
```

Or we can have the chatbot first check for the keyword mother if no match it checks the next keyword and if no match it goes on to the third keyword and so on until it either finds a match or we reach the end and use the random response.

```
if (condition){
    ...
}
else if (condition){
    ...
}
else if (condition){
    ...
}
else if (condition){
    ...
}
...

```

If this seems confusing to you try to sketch a flowchart. It can really help visualize the flow of a program. It's vitally important that you understand which **else** goes with which **if**. If you have any confusion, please ask the instructional team.

5. Let's start by changing the order of the if statements to make it so that we check for the keyword mother first.
6. Now either surround the other if statements in an else block or change all the subsequent if statements to be else if.
7. Run your program and test it with an input that contains mother and some other keyword(s).
8. When you are satisfied with your code, show it to your instructor or a TA to receive credit for this section.