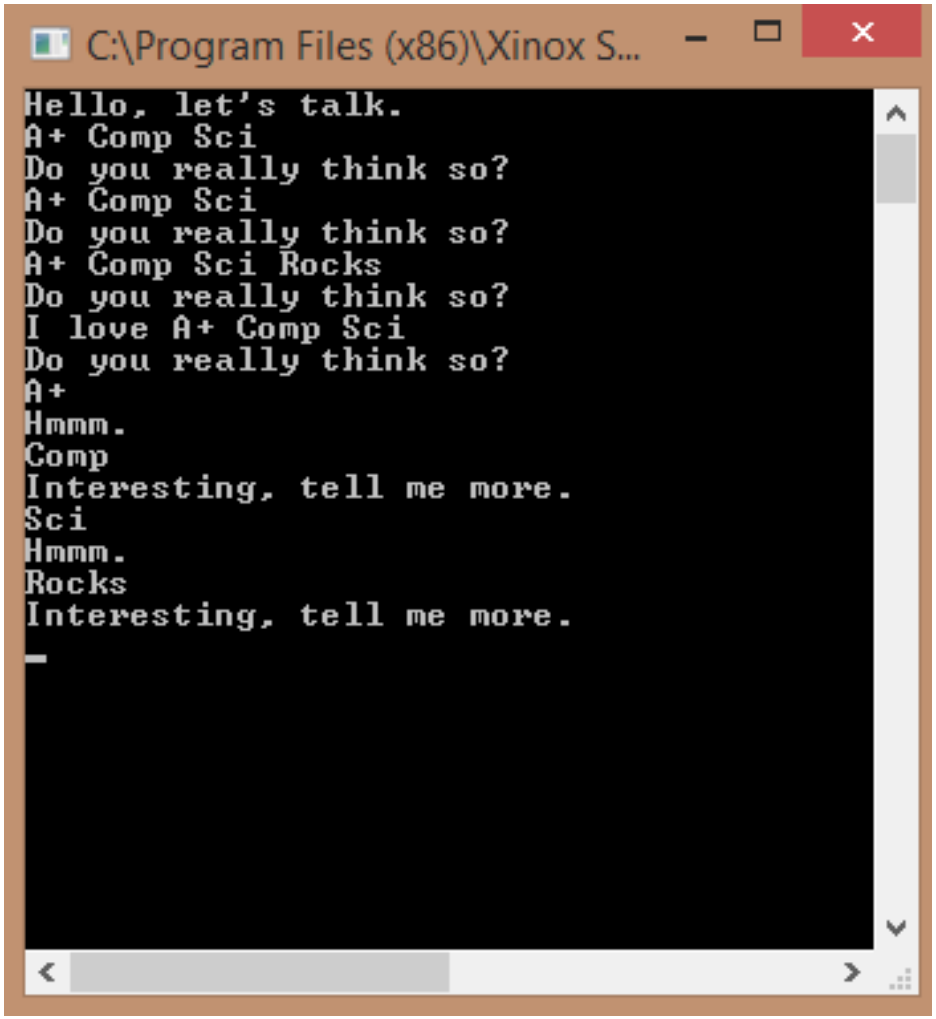
Magpie



What is Magpie?

Magpie is a lab that focuses on classes, randomness, and Strings.

This lab will make sure that you know how to use the String methods substring and indexOf.

Both substring and indexOf have multiple forms as these methods have been overloaded.

What is NLP?

NLP stands for Natural Language Processing. It is a field of Computer Science that studies how computers can understand human language. The Magpie Chatbot lab is designed to help us explore some of the basics of NLP.

To get started with these activities, we will first review the cascading `if else` structure and `String` methods.

Cascading if else

```
String letter = "C";
int ascii=0;
if(letter.equals("A")) {
   ascii=65;
}
else if(letter.equals("B")){
   ascii=66;
}
else if(letter.equals("C")){
   ascii=67;
}
else if(letter.equals("D")){
   ascii=68;
}
else{
   ascii=69;
}
out.println(ascii);
```

OUTPUT

67

```
public String getResponse(String statement)
{
        String response = "";
        if (statement.indexOf("no") >= 0)
            response = "Why so negative?";
        else if (statement.indexOf("mother") >= 0 ||
                 statement.indexOf("father") >= 0 ||
                 statement.indexOf("sister") >= 0 ||
                 statement.indexOf("brother") >= 0)
        {
            response = "Tell me more about your family.";
        }
        else
        {
            response = getRandomResponse();
        }
       return response;
}
```

Cascading if else

String

Methods from AP CS Subset

Name	Use
`int length()`	Returns length of String
`int indexOf(String str)`	Returns first position of str in the string if found, -1 if not found
`String substring(int from)`	Returns a substring of the string starting at from to length() – 1
`String substring(int from, int to)`	Returns a substring of the string starting at from to to – 1

String

Methods from AP CS Subset

Name	Use
`boolean equals(Object other)`	Returns true if the other and this String match
`int compareTo(String str)`	Returns : a positive number if this string > str, a negative number if this string < str, 0 if this string is equal to str.

String

Methods not in the AP CS Subset

Name	Use
`String toLowerCase()`	Returns length of String
`int indexOf(String str, int startPos)`	Returns position of str in the string starting at `startPos` if found, -1 if not found
`String trim()`	Returns a substring of the string without all leading and trailing whitespaces

open
Magpie Lab \
Magpie Code \
Magpie2.java
MagpieRunner2.java

Start work on Activity 2 Lab

Activity 3
Better Keyword Detection

As you complete Activity 2, you will notice that your chatbot does not distinguish between whole word matches and partial word matches.

For example, if your statement is:

"Catch me if you can!"

and you are looking for *"cat"* the `getResponse` method will return 0 instead of -1.

Activity 3
Better Keyword Detection

To facilitate whole word matches, Magpie3.java adds the `findKeyword` method.

This method uses the `String` method `int indexof(String str, int startPos)` to determine if a whole word instead of a partial word is found.

Start work on Activity 3 Part One Lab

Activity 3
Better Keyword Detection

Now that you have completed Part One of Activity 3 Lab, run the new version of your chatbot to see how it has changed. Use some of the original examples from Activity 2.

open Magpie Lab \ Magpie Code \ Magpie3.java MagpieRunner3.java

findKeyword method

Study the `findKeyword` method. Use the Activity 3 Worksheet to trace calls to this method.

Questions:

What is the purpose of the local variables `before` and `after`?

Why are they initialized to a space at the beginning of each loop iteration?

findKeyword method

Questions:

What is the purpose of the local variables `before` and `after`?

ANSWER: To determine if the found string is a whole word or partial word. If the characters before and after goal are not letters, a whole word version of goal has been found.

findKeyword method

Questions:

Why are `before` and `after` initialized to a space at the beginning of each loop iteration?

ANSWER: If `goal` is found at the beginning of statement, there are not characters before it, so `before` is initialized to a space. This logic is used for `after`, if `goal` comprises the last characters of the string.

Continue work on Activity 3 Part Two Lab

Activity 4
Responses that Transform Statements

In this activity, your chatbot will respond to certain phrases, not just specific keywords.

This revised version responds to phrases in the form of "*I want to* something" and "Whatever *you* something *me*..."

Run the new version of your chatbot.

open
Magpie Lab \
Magpie Code \
Magpie4.java
MagpieRunner4.java

Adding to getResponse

```
//transform statements addition to getResponse
else if(findKeyword(statement, "I want to", 0) >= 0)
  response = transformIWantToStatement(statement);
else
{
  int pos = findKeyword(statement, "you", 0);

  if (pos >= 0 &&
      findKeyword(statement, "me", pos) >= 0)
  {
    response = transformYouMeStatement(statement);
  }
  else
     response = getRandomResponse();
 }
```

transformIWantStatement method

```
private String transformIWantToStatement(String statement)
{
    statement = statement.trim();
    String lastChar =
            statement.substring(statement.length() - 1);
    if (lastChar.equals("."))
       statement = statement.substring(0,
                                statement.length() - 1);
    int pos = findKeyword(statement, "I want to", 0);
    String restOfStatement =
                statement.substring(pos + 9).trim();
    return "What would it mean to " +
            restOfStatement + "?";
}
```

Note: The length of *"I want to"* is 9.

transformYouMeStatement method

```
private String transformIYouMeStatement(String statement)
{
    statement = statement.trim();
    String lastChar =
       statement.substring(statement.length() - 1);
    if (lastChar.equals("."))
       statement = statement.substring(0,
                   statement.length() - 1);
    int posOfYou = findKeyword(statement, "you", 0);
    int posOfMe = findKeyword(statement, "me",
                              posOfYou + 3);
    String restOfStatement =
           statement.substring(posOfYou + 3,
                               posOfMe).trim();
    return "What makes you think that I " +
            restOfStatement + " you?";
}
```

Start work on the Activity 4 Lab