About the AP Computer Science Labs

Introduction to Labs

The College Board has provided three labs to give you practice editing and working with larger software programs. They cover topics that you will definitely see on the AP Exam, so they're a really good way to use what you've learned. We'll start with the Magpie Lab, and cover the other two labs later in the term:

Magpie Lab (Week 3)

In the Magpie Lab, you will explore some of the basics of Natural Language Processing. You will work with a variety of methods of the String class and practice using the if statement. You will also trace a complicated method to find words in user input.

Elevens Lab (Week 9)

This lab is related to a simple solitaire game called Elevens. You will learn the rules of Elevens, and will be able to play it by using the supplied Graphical User Interface (GUI).

Picture Lab (Week 15)

In this lab, you will write methods that modify digital pictures. In writing these methods, you will learn how to traverse a two-dimensional array of integers or objects. You will also work with nested loops, binary numbers, interfaces, and inheritance.

How to complete the labs

Mrs. Dovi will introduce you to each of the labs in a short video. You will then have a series of activities to complete related to the lab. You will not submit these lab activities in the course. Rather, complete the activities and share your responses on Piazza.

While these labs are not explicitly tested on the AP Computer Science Exam, it is highly likely that you will be asked to complete similar types of processing in the free response section of the exam. For this reason it is important that you spend the time coding and solving problems using the labs.

Magpie ChatBot Lab

Description

In this lab you will be working with a ChatBot program. A ChatBot is a program designed to mimic conversation. The main emphasis of this lab is String processing, so in these programs you will be chatting back and forth with the program. As you work through the lab you will be able to change how it responds.

This uses Natural Language Processing – a field of computer science based on artificial intelligence. Examples include Siri on iPhones and Google Translate. While the programs you will be creating in the Magpie Lab are relatively simple, they provide a view into this very exciting field of computer science.

Mrs. Dovi's introduction video: https://www.youtube.com/watch?v=SRrleiNde58

If you are unable to stream videos over YouTube, use the following link t**o download this lesson's videos**: http://term.2.videos.s3.amazonaws.com/T2_Magpie.zip

Get started

The Magpie Lab has five activities that range from trying a ChatBot to changing the Magpie classes to modifying the Magpie classes. There are several classes that implement Magpie. You will need to use a runner program to instantiate the class in order to get it working.

College Board provided a Student Activity Guide available to help you work through the lab and the activity starter codes. We've provided worksheets for you to complete for each activity.

Required downloads:

- Student Activity Guide: http://2014-15.s3.amazonaws.com/Labs/Magpie/MagpieLabStudentGuide_updated_2014_Final.pdf
- Activity starter code: http://2014-15.s3.amazonaws.com/Labs/Magpie/MagpieActivityStarterCode.zip
- Activity worksheets: http://2014-15.s3.amazonaws.com/Labs/Magpie/MagpieActivityWorksheets.pdf

Complete each activity and worksheet. Share them with your teachers, classmates and on Piazza. If you have any questions, head to the forum.

- Activity 1 - Try out a ChatBot online to clarify what a ChatBot is and how they are used in computing.
- Activity 2 - Work with the Magpie code by implementing the existing class, Magpie2.
- Activity 3 - Modify the Magpie classes by working with String methods.
- Activity 4 - Make the ChatBot more responsive by changing what the user types in.
- Activity 5 - Use arrays in the ChatBot for processing.

Resources and Links:

- Elbot, a sample ChatBot: http://www.elbot.com/
- Try more ChatBots: https://sites.google.com/site/webtoolsbox/bots
- CS Bits and Bytes (vol. 1 issue 4) Language Processing: http://www.nsf.gov/cise/csbytes/newsletter/vol1/vol1i4.html
- CS Bits and Bytes (vol. 1 issue 10) Artificial Intelligence: http://www.nsf.gov/cise/csbytes/newsletter/vol1/vol1i10.html

Term 2 – Week 3 Name: _____

Magpie Lab: Part 1 – Activity 1 – Overview

Overview – What is a ChatBot?

A ChatBot is a program designed to simulate conversation and is a part of the branch of computer science known as Artificial Intelligence. The main emphasis of this lab is String processing similar to what we have seen in the Tweet Tester and Wild Card labs. Today you will try out some ChatBots to see what they do.

Link: http://www.pandorabots.com/pandora/talk?botid=f5d922d97e345aa1

1. Open the link above. Try out several of the ChatBots linked there like Fake Captain Kirk or Claudio.
2. Pick two to compare. How do they respond to the following:

	ChatBot 1	**ChatBot 2**
a. Hi there	_____	_____
b. I like apples	_____	_____
c. Where are you	_____	_____
d. How are you	_____	_____

3. Comparing these answers, how were the two ChatBots the same? Different?

4. Many ChatBots work by searching for specific keywords, like **Where**. What keywords did your two ChatBots use?

5. As the field of Artificial Intelligence grows how do ChatBots change how we interact with technology? How do you think they might be used as smart phones and the Internet become more widespread?

Term 2 – Week 3 Name: _____

Magpie Lab: Part 2 – Activity 2 – First Look at the Code

To explore the code make sure you have all the Magpie lab materials saved on your computer. The Magpie lab consists of several files in different directories. Each Activity is in a separate folder in the guide.

Today we will be working through the first part of Activity 2. The files can be found here: **`MagpieActivityStarterCode\activity2.`**

1. Open the files **`MagpieRunner2.java`** and **`Magpie2.java`**. Compile both and run the Runner program. What does it do?

2. How does the Runner program stop repeating the conversation?

3. As you use the program, what keywords do you notice?

4. Look at the **`Magpie2.java`** file. What do each of the following methods do:

   a. **`getGreeting()`**

   b. **`getResponse(String statement)`**

   c. **`getRandomResponse()`**

5. Compare the keyword list you developed in question 3 above to the if statement in the **`getResponse()`** method. How would you add more keywords to this ChatBot?

Term 2 – Week 3 Name: _____

Magpie Lab: Part 3 – Activity 2 – Changing the Code

To explore the code make sure you have all the Magpie lab materials saved on your computer. The Magpie lab consists of several files in different directories. Each Activity is in a separate folder in the guide.

Today we will be working through the second part of Activity 2. The files can be found here: **`MagpieActivityStarterCode\activity2.`**

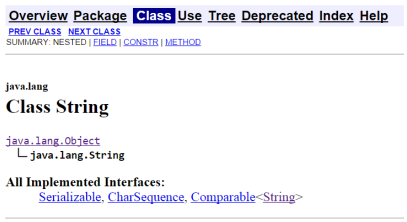
1. Open the files **`MagpieRunner2.java`** and **`Magpie2.java`**.
2. Complete Activity 2 in the Magpie Student Guide on page 3.
3. Answer the question on page 4 here:

Term 2 – Week 3 Name: _____

Magpie Lab: Part 4 – Activity 3 – String APIs

This activity asks you to work with the String class. While we covered this class back in Term 1, this activity asks you to investigate and use some of the methods in the String class that are not covered on the APCS A Exam.

In order to use these methods you need to read the String class' API. An API (Application Programming Interface) is a listing of the features and behaviors of a system that tell a programmer how to use them. We have looked at these several times this year in the lessons and webinars, specifically for the Math and String classes.



Overview Package Class Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: NESTED | FIELD | CONSTR | METHOD

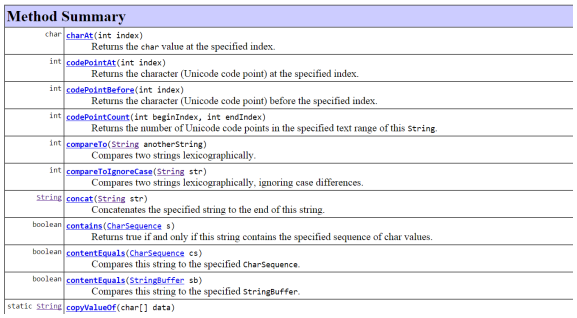
java.lang

Class String

java.lang.Object
  └ java.lang.String

All Implemented Interfaces:

Serializable, CharSequence, Comparable<String>



Method Summary

char	charAt(int index) Returns the char value at the specified index.
int	codePointAt(int index) Returns the character (Unicode code point) at the specified index.
int	codePointBefore(int index) Returns the character (Unicode code point) before the specified index.
int	codePointCount(int beginIndex, int endIndex) Returns the number of Unicode code points in the specified text range of this String.
int	compareTo(String anotherString) Compares two strings lexicographically.
int	compareToIgnoreCase(String str) Compares two strings lexicographically, ignoring case differences.
String	concat(String str) Concatenates the specified string to the end of this string.
boolean	contains(CharSequence s) Returns true if and only if this string contains the specified sequence of char values.
boolean	contentEquals(CharSequence cs) Compares this string to the specified CharSequence.
boolean	contentEquals(StringBuffer sb) Compares this string to the specified StringBuffer.
static String	copyValueOf(char[] data)

This screenshot on the left shows the String API. This information at the top shows where the String class fits in to the hierarchy of all of the Java classes, and gives some information about how it works. On the right the screenshot shows the beginning of the method summary listing everything available through the String class.

Use the String API to answer the following:

Link: http://docs.oracle.com/javase/8/docs/api/java/lang/String.html

1. How many constructors are in the String class?
2. How many **`indexOf`** methods are in the String class?
3. What does the **`replaceAll(String regex, String replacement)`** method do?
4. Are there any methods there that would have made the TweetTester lab easier to complete?

Term 2 – Week 3 Name: _____

Magpie Lab: Part 5 – Activity 3 – Magpie API

In the last worksheet you explored the String API. These API's exist for all classes built into Java and in fact any programmer can create an API for a new class they create.

1. Open The Magpie3 lab. Look at the **`findKeyword`** method. Read the comments above this method. What does the method appear to do?

2. JavaDoc is a specific way of coding comments so that an HTML file with the class's API can be created. The comments you see in the **`findKeyword`** method follow this format. Notice it starts with /** and uses several keywords with @ in front. These tell Java how to create the HTML page with the API.

   Dr. Java has a simple way of creating JavaDoc from our code. Go to the **Tools** menu and select **Preview JavaDoc for Current Document**. This should open up an HTML file in your browser in the JavaDoc format.

3. In the Magpie3 API open in your browser look for the **`findKeyword`** method, you will notice it is not there. Look back at the code for the **`findKeyword`** method. What Java keyword in the header (first line) of the method might make this not show up in the JavaDoc?

4. Change the **`findKeyword`** method from **`private`** to **`public`**. Recompile, then using the JavaDoc tool preview the API. What is different?

5. Looking at the JavaDoc for the **`findKeyword`** method what do the following appear to do?
   a. @param statement _____
   b. @param goal _____
   c. @param startPos _____
   d. @return _____

6. Let's try creating JavaDoc from scratch. Add a method to Magpie3:

   ```
   public void doNothing (int dud)
    {       //does nothing     }
   ```

   Write your comment here:

7. Why do you think Java allows programmers to create APIs using JavaDoc?

   Dr Java offers a more complete tutorial on using JavaDoc and it can be found here:
   Link: http://www.drjava.org/docs/user/ch10.html

Term 2 – Week 3 Name: _____

Magpie Lab: Part 6 – Activity 3 – The findKeyword Method

In this worksheet you will finish up Activity 3 by exploring what the **`findKeyword`** method does.

Open the **`Magpie3.java`** file and use it to answer the following questions.

If you changed the **`findKeyword`** method to be **`public`** in Part 5 of this guide you should change it back to **`private`** and recompile.

1. Since the **`findKeyword`** method is private it cannot be called from a runner program. Where in the Magpie3 file is the code called?

2. There are two versions of the **`findKeyword`** method in the Magpie3 lab. What is the difference between the two headers?

   Having two methods with the same name in the same class is called _____.

   Java tells which of the two versions to use based only on the _____.

3. While the two versions of the **`findKeyword`** method performs a similar task, they do so in a slightly different way. How are they different?

4. On page seven of Activity 3 complete the chart tracing the **`findKeyword`** method.

Term 2 – Week 3 Name: _____

Magpie Lab: Part 7 – Activity 4 – More responses

This section of the Lab implements more sophisticated ChatBot algorithms. In this version some keywords, like **want** cause the Magpie to respond using you own words.

Open the Magpie4 Lab and Runner and compile them both, then run the Runner program.

1. Try chatting. Are there any new keywords you notice?

2. Try the following phrases. What response do you get?

   a. I want to program. _____
   b. Do you like cheese? _____
   c. Do you like me? _____
   d. You like me? _____
   e. You type me. _____

   What patterns do you recognize in these phrases?

3. Work through the exercises on pages 9 and 10.

Term 2 – Week 3 Name: _____

Magpie Lab: Part 8 – Activity 5 – Using Arrays

This version of the Magpie Lab behaves in the same way as the version in Activity 4. The difference is how it generates a random response when no keywords are found.

Open the Magpie4 Lab and Runner and compile them both, then run the Runner program.

1. Try chatting. How does the behavior compare to version 4 of Magpie:

2. At the bottom of the Magpie5 Lab find the code:

```
private String [] randomResponses = {"Interesting, tell me more",
   "Hmmm.",
   "Do you really think so?",
   "You don't say."
 };
```

This array of Strings holds the responses the ChatBot randomly says. Add two more random responses and compile, then run the runner for Magpie5.

3. Look at the **`getRandomResponse`** method in the Magpie5 Lab and compare it to the same method in the **`getRandomResponse`** method in Magpie4. Compare and contrast how the two methods work.

4. Let's say you wanted to increase the number of random responses to 30 different phrases. Which of the two methods, the one in Magpie 4 or Magpie5, would be most efficient? Why?

5. How does storing the responses in array enable increasing the number of random responses the Magpie ChatBot can use?