Lab 4

Learning about Mathematical Functions, Characters, and String Methods.

This Lab focuses on the use of the Java API to help you find and learn about various Mathematical Functions and String Methods that you may use in your programs. You will also be doing more practice with the character variable data type.

In-Lab Activities

Don't forget to answer the **Guided Inquiry Questions** in your notebooks. It is also encouraged to use it frequently while working through activities.

Warm-Ups

Go to the Google Forms link on the board and complete your Warm-Up Questions. You have 10 minutes to complete this. If you finish early, feel free to begin the Lab Activities, but make sure your partner is ready as well.

TIP You will need to use the **Java API** for this entire Lab.

Activity 1: Ad Lib

For this Activity, write a program that instructs the user to enter four specific values: a verb, an adjective, an adverb, and a noun. With these four values, create a sentence by **concatenating** them together into one final string and display it on the console. For example, if the noun is Bob, verb is running, adverb is fast, and adjective is crazy; the string should display something like "Crazy Bob is running fast."

For this activity, you must use a **string method**. Look through the **API**; which **method** will you use?

1. Display **messages** to instruct the user to enter the specific words. Don't forget to also save these words. What variables must you **declare** and of what **data types** will they be?
2. **Concatenate** these words into one final **string**. While concatenating, make sure it is in the order you wish, so that the final result is a sentence that makes sense. (What **String Methods** will you use to achieve this? Look in the **Java API** under the **String** class.)
3. Display this string back to the user.

TIP Don't forget to **import** packages that are necessary for specific methods to be used.

Activity 2

Add on to your code for Activity 1 for this one. After printing the final string, also display:

1. The character at **index** of 0
2. The character at the last **index**
3. The character at the middle **index**. (Think about how you can calculate the middle index by also finding the first and last index positions.)

You must use **String methods** for these, which you can look up in the **Java API.**

What is the **dot operator**? Put a comment at the top stating which lines of your program use the **dot operator**.

CHECKPOINT

30 MINUTES

Brainstorming Individually (3-5 minutes)

Read the next Activity and start brainstorming; think about ways you could start solving this problem and write down any notes. Do this *individually* for 3 to 5 minutes and then come together afterwards to compare, discuss, and code. ☺

Activity 3

Write a program that allows the user to create an account by saving a username and password; display a message saying that their account has been created. Next, allow the user to log back into their new account, having some verification to check whether or not the entered account information is equal. If the user successfully logged in, display a success message. Otherwise, tell the user that they entered incorrect information.

Activity 4

Create a String using the quote below, or find your own programming quote, and display it:

"The programmers of tomorrow are the wizards of the future. You're going to look like you have magic powers compared to everybody else."

Now do the following operations on the string:

1. Find the **length** of the string and display it. Now count how many characters there are. Do these values match? Why or why not?
2. Ask the user to enter a string, and check whether or not that string is contained in the quote. Display a message accordingly, as well as the index (if found).
3. Ask the user to enter a character, and display the index of that character. What is the **index** value if that character is not located in the string? Now vice versa, allow the user to enter an index position, and display the character at that index.
4. Display the quote, but with the second word being all uppercase. Do not type out any of it manually – only use string methods to do this.

Which **methods** are used? Are they **static methods** or **instance methods** (Hint: the **API** may help you answer this)? What is the difference between a **static** and **instance method**?

5. Create a string with the text " – Gabe Newell", or the name of whoever said your quote. Now display the original quote with this new string attached. Use a method from the API to do so.

CHECKPOINT

45 MINUTES

Activity 5

This lab activity will show how you can make your mathematical expressions more or less precise. Write a new program that divides two numbers and gives a precision of only two places after the decimal point. Make sure if a number is divided by 0, you do not get a runtime error.

Activity 6

In this activity, you will make a program to simulate a die roll. Using a random number generator, make a standard 6-sided die. Have a random number from 1 through 6 be generated and displayed.

Now, simulate three more dice rolls, using three separate variables. Display each new number on a **new line** using a **whitespace character**.

Write the **pseudocode** in the header comments before starting this activity.
(You don't need to write in your notebook for this one.)

Activity 7

Download and import the Java class called Calculator. This will have a menu and various math functions. However, there are both syntax *and* logic errors in this code that you must fix. Work with your partner to fix everything while keeping a list of what errors you found and what made each particular error. Your TA will check this list before the end of Lab.

CHECKPOINT

30 MINUTES

Reflections

Go to the Google Forms link given by your TA to work on your Reflection for Lab 4. This counts as a part of your participation grade, and it is important that you put as much detail as possible.