

In this lab, you'll use various methods to manipulate Strings in Java. You will check String length, shorten Strings, and search for specific parts of a String.

Let's use Riley as an example. Riley has a problem. She is developing a library management system for clients to better organize and catalog their book inventory. However, there is an issue with capturing book titles: when they are too long, they do not display correctly in the management system.

To ensure book titles display correctly, they must be no more than 20 characters long. Riley needs to check the title length and shorten it if it exceeds this limit. Additionally, Riley wants the system to search for specific words in a book title to enhance functionality.

You've been provided with the starter code. To help Riley, you must write the necessary code to accomplish the requested tasks. Let's get started!

Reflection point: Understanding arrays

Previously, you learned that an array in Java is a collection of elements of the same type stored in contiguous memory locations. Arrays allow efficient data access and manipulation using an index and help handle multiple values with a single variable. Therefore, a String is a sequence of characters represented internally as an array of the char data type.


Understanding arrays is crucial because it forms the basis of how Strings and other data structures work in Java. By grasping the concept of arrays, you will be better equipped to manipulate data efficiently, as you will be doing in this lab.

Goal

Can you help Riley by checking book title lengths, shortening long titles, and searching for specific words in book titles?

In your lab environment, open IntelliJ by double-clicking on the icon.



 It's time to get coding!

Pay attention to the starter code. The main method of the main class is placed in a file named **Main.java** containing:

```
public class Main {  
  
    public static void main(String[] args) {  
  
  
    }  
  
}
```

First, you will create a book title, write code to check its length, and then print the book title's length. After confirming its length, you'll limit its length to 20 characters if necessary.

Tip

In Java, `//` is used to create single-line comments. Any text following `//` will be ignored by the compiler, meaning that it won't execute as part of your program.

If you delete a `TODO` statement but leave the `//` in place, any code you write on that line will remain commented out and won't run. To ensure your code works properly, remove the `//` as well.

Step 1: Declare and initialize a variable, then check its length

Inside the main method, declare a variable name `bookTitle`, and assign a name to it. Remove the `TODO` block in the starter code and insert your code to find the length of `bookTitle`.

```
String bookTitle = "The Adventures of Captain Fantastic and the Magical Unicorn";
```

```
int titleLength = //TODO: Complete the code statement to find out the length of  
bookTitle
```

```
System.out.println("Length of the book title: " + titleLength);
```

Step 2: Declare and initialize a variable then check its length

Once you've confirmed the length of `bookTitle`, check if the book title is more than 20 characters long. If it is longer, use the `substring()` method to limit the book title length to 20 characters.

Tip

The `substring` method takes two arguments.

For example, consider the String 'Dictionary'. If you want to extract five characters from the second character, you would enter `substring(1,5)`. The result would be 'ictio'.

Note: The first character is always in index 0.

Remove the TODO block and complete the code to shorten the book title and limit it to 20 characters.

```
//Declare and initialize the length of the shortened book title

int maxLength = 20;


// Shortened the book title

String shortBookTitle = // TODO: Write the code to shorten the bookTitle to 20
characters

System.out.println("Original title: " + bookTitle);

System.out.println("Shortened title: " + shortBookTitle);
```

Tip

In Windows, you use the Control (ctrl) key to cut and paste. For example:

Copy: Ctrl + C


Paste: Ctrl + V

However, if you're using a Mac, remember that copying and pasting is done using the Command (CMD) key, not the Control key. For example:

Copy: CMD + C

Paste: CMD + V

Riley also wants her system to be searchable, particularly for finding words or characters within book titles. Now you'll search for a word in the book title you created.

 It's time to get coding!

Step 1: Find a word in the book title

Use the `contains()` method to check if a particular word is present in the book title.

Declare and initialize a String variable `searchWord`, let's say the String "Captain". Remove the TODO block and complete the code to check if the `searchWord` is in `bookTitle`.

```
// Declare and initialize the searchWord

String searchWord = "Captain";


// Check if the searchWord is present in book title

boolean containsWord = //TODO: Write the code snippet to check if searchWord is
present in bookTitle.

System.out.println("Does the title contain the word \"" + searchWord + "\"? " +
containsWord);
```

In this lab, you learned how to use the `length()`, `substring()`, and `contains()` methods in Java to manipulate Strings. You checked the length of a String, shortened a too-long String, and searched for a part of a String—all in the context of Riley's library management system! You saw how manipulating Strings has real-world uses in Java programs.

Consider working on an example where you can use different String methods. Practice makes perfect, after all!

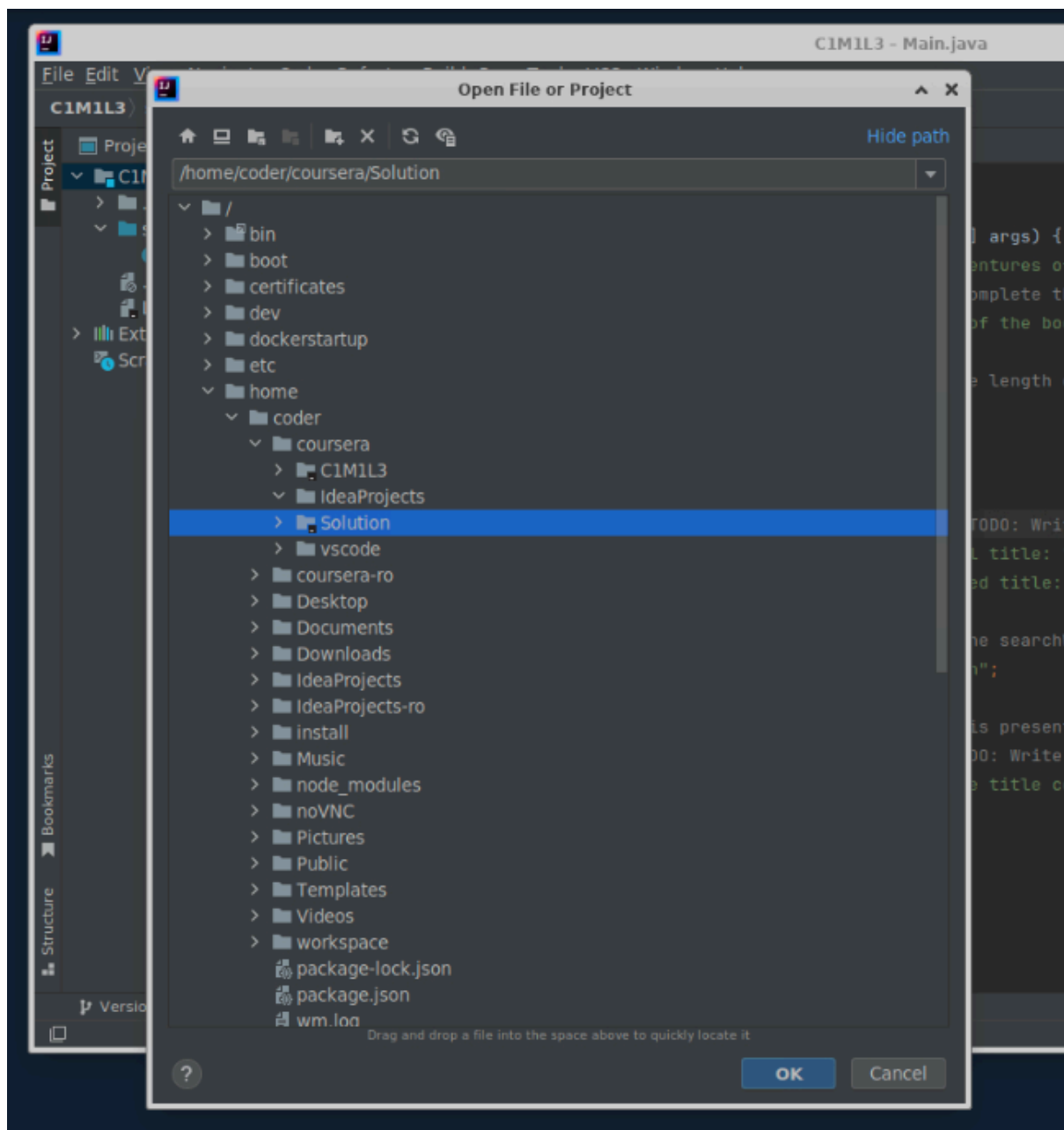
 It's time to get coding!

Using the sentence "Java programming is fun and educational", complete the following tasks:

1. Check if the sentence contains the word “fun”.
2. Replace the word "educational" with “awesome”.
3. Find the length of the original sentence.
4. Shorten the sentence to 17 characters.

After completing your project, take a moment to review the exemplar solution. This sample solution can offer insights into different coding techniques and approaches. You can view the exemplar solution project in the solution folder located here:

`/home/coder/coursera/Solution`



- Open the project in IntelliJ by going to File > Open and navigating to the /home/coder/coursera/Solution folder.
- When prompted, select "in a new window" to see your project and the exemplar solution at the same time.
- Reflect on what you can learn from the exemplar solution to improve your coding skills.
- Remember, multiple solutions can exist for a problem; the goal is to learn and grow as a programmer by exploring various approaches.

Use the exemplar solution as a learning tool to enhance your understanding and refine your approach to coding challenges.