Congratulations! You passed!

Grade received 100% **Latest Submission Grade** 100% **To pass** 80% or higher

Go to next item

To pass this course item, you must complete the activity and receive at least 80%, or 8 out of 9 points, on the questions that follow. Once you have completed the activity and questions, review the feedback provided. You can learn more about graded and practice items in the course overview.

Activity Overview

In this activity, you will create a new portfolio document to demonstrate your experience using Python to develop algorithms that involve opening files and parsing their contents. You can add this document to your cybersecurity portfolio, which you can share with prospective employers or recruiters. To review the importance of building a professional portfolio and options for creating your portfolio, read Create a cybersecurity portfolio.

To create your portfolio document, you will review a scenario and follow a series of steps. This scenario is connected to the Create another algorithm lab that you have just completed. You will explain the code you developed in that lab, and this will help you prepare for future job interviews and other steps in the hiring process.

Be sure to complete this activity before moving on. The next course item will provide you with a completed exemplar to compare to your own work.

Scenario

Review the following scenario. Then complete the step-by-step instructions.

You are a security professional working at a health care company. As part of your job, you're required to regularly update a file that identifies the employees who can access restricted content. The contents of the file are based on who is working with personal patient records. Employees are restricted access based on their IP address. There is an allow list for IP addresses permitted to sign into the restricted subnetwork. There's also a remove list that identifies which employees you must remove from this allow list.

Your task is to create an algorithm that uses Python code to check whether the allow list contains any IP addresses identified on the remove list. If so, you should remove those IP addresses from the file containing the allow list.

Note: This scenario involves developing the same algorithm that is developed in Tasks 2-7 of the Create another algorithm lab. (You do not need to reference Task 1 and Tasks 8-10 of the lab to complete this portfolio activity.) You should revisit the lab to get screenshots to include in your portfolio document.

Step-By-Step Instructions

Follow the instructions to complete each step of the activity. Then, answer the 9 questions at the end of the activity before going to the next course item to compare your work to a completed exemplar.

Step 1: Access the template

To use the template for this course item, click the link and select *Use Template*. (In this step, you will just open the template. More instructions for how to use the template will be included in later steps.)

Link to template: Algorithm for file updates in Python

OR

If you don't have a Google account, you can download the template directly from the following attachment.

Algorithm for file updates in Python
DOCX File

Step 2: Access supporting materials

The following supporting material will help you complete this activity. The document **Instructions for including Python code** provides instructions and best practices for including samples of Python code in your portfolio activity. Keep it open as you proceed to the next steps.

To use the supporting material for this course item, click the link and select *Use Template*.

Link to supporting material: Instructions for including Python code

OR

If you don't have a Google account, you can download the supporting material directly from the following attachment.

Instructions for including Python code
DOCX File

Step 3: Open the file that contains the allow list

The file that you want to open is called `"allow_list.txt"`. Assign a string containing this file name to the `import_file` variable. Then, use a `with` statement to open it. Use the variable `file` to store the file while you work with it inside the `with` statement.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Open the file that contains the allow list** section of the **Algorithm for file updates in Python** template. In the **Task 2** section of **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

Step 4: Read the file contents

Next, use the `.read()` method to convert the contents of the allow list file into a string so that you can read them. Store this string in a variable called `ip_addresses`.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Read the file contents** section of the **Algorithm for file updates in Python** template. In the **Task 3** section of the **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

Step 5: Convert the string into a list

In order to remove individual IP addresses from the allow list, the IP addresses need to be in a list format. Therefore, use the `.split()` method to convert the `ip_addresses` string into a list.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Convert the string into a list** section of the **Algorithm for file updates in Python** template. In the **Task 4** section of the **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

Step 6: Iterate through the remove list

A second list called `remove_list` contains all of the IP addresses that should be removed from the `ip_addresses` list. Set up the header of a `for` loop that will iterate through the `remove_list`. Use `element` as the loop variable.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Iterate through the remove list** section of the **Algorithm for file updates in Python** template. In the **Task 5** section of the **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

Step 7: Remove IP addresses that are on the remove list

In the body of your iterative statement, add code that will remove all the IP addresses from the allow list that are also on the remove list. First, create a conditional that evaluates if the loop variable `element` is part of the `ip_addresses` list. Then, within that conditional, apply the `.remove()` method to the `ip_addresses` list and remove the IP addresses identified in the loop variable `element`.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Remove IP addresses that are on the remove list** section of the **Algorithm for file updates in Python** template. In the **Task 6** section of the **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

In addition, include a sentence that explains that applying the `.remove()` method in this way is possible because there are no duplicates in the `ip_addresses` list.

Step 8: Update the file with the revised list of IP addresses

Now that you have removed these IP addresses from the `ip_address` variable, you can complete the algorithm by updating the file with this revised list. To do this, you must first convert the `ip_addresses` list back into a string using the `.join()` method. Apply `.join()` to the string `"\n"` in order to separate the elements in the file by placing them on a new line.

Then, use another `with` statement and the `.write()` method to write over the file assigned to the `import_file` variable.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Update the file with the revised list of IP addresses** section of the **Algorithm for file updates in Python** template. In the **Task 7** section of the **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

Step 9: Finalize your document

To finalize the document and make its purpose clear to potential employers, be sure to complete the **Project description** and **Summary** sections of the **Algorithm for file updates in Python** template.

In the Project description section, give a general overview of the scenario and what you accomplished in Python. Write three to five sentences.

In the Summary section, provide a short summary of the algorithm by highlighting its main components. Write four to six sentences.

Pro Tip: Save a copy of your work

Finally, be sure to save a copy of your completed activity. You can use it for your professional portfolio to demonstrate your knowledge and/or experience to potential employers.

What to Include in Your Response

Be sure to address the following in your completed activity:

- Screenshots of your Python code or typed versions of the code
- Explanations of the syntax, functions, and keywords in the code
- A project description at the beginning
- A summary at the end
- Details on using a `with` statement and the `open()` function in your algorithm
- Details on using the `.read()` and `.write()` methods in your algorithm
- Details on using the `.split()` method in your algorithm
- Details on using a `for` loop in your algorithm
- Details on using the `.remove()` method in your algorithm

Step 10: Assess your activity

The following is a self-assessment for your **Update a file through a Python algorithm** portfolio activity. You will use these statements to review your own work. The self-assessment process is an important part of the learning experience because it allows you to objectively assess your **Update a file through a Python algorithm** portfolio activity.

There are a total of 9 points possible for this activity and each statement is worth 1 point.

To complete the self-assessment, first open your **Update a file through a Python algorithm portfolio** activity. Then respond yes or no to each statement.

When you complete and submit your responses, you will receive a percentage score. This score will help you confirm whether you completed the required steps of the activity. The recommended passing grade for this project is at least 80% (or 8/9 points). If you want to increase your score, you can revise your project and then resubmit your responses to reflect any changes you made. Try to achieve at least 8 points before continuing on to the next course item.

1. Your document includes screenshots or typed versions of your Python code. **1 / 1 point**

- (selected) Yes
- No

Correct

2. Your document includes a description of the project at the beginning. **1 / 1 point**

- (selected) Yes
- No

Correct

3. Your document includes explanations of your Python code. **1 / 1 point**

- (selected) Yes
- No

Correct

4. Your document includes a summary at the end. **1 / 1 point**

- (selected) Yes
- No

Correct

5. Your document includes details on using a `with` statement and the `open()` function in your algorithm. **1 / 1 point**

- (selected) Yes
- No

Correct

6. Your document includes details on using the `.read()` and `.write()` methods in your algorithm. **1 / 1 point**

- (selected) Yes
- No

Correct

7. Your document includes details on using the `.split()` method in your algorithm. **1 / 1 point**

- (selected) Yes
- No

Correct

8. Your document includes details on using a `for` loop in your algorithm. **1 / 1 point**

- (selected) Yes
- No

Correct

9. Your document includes details on using the `.remove()` method in your algorithm. **1 / 1 point**

- (selected) Yes
- No

Correct