Thank you for taking part in this study of the Azure Storage SDK.

This study is all about how well the SDK meets your expectations.

Throughout the study you will use a storage account that has already been created for you. You will be given access to this during the session and can refer to the Azure portal at any time you need to.

You will also be given access to some documentation. The documentation is not yet fully complete but we would like your feedback on the content that you are given access to during the study.

**Task 1**

Create a console app project and set it up so that it uses the Azure Blob Storage Preview 3 SDK.

Inside the app, write a method that asks for the name of a container and the name of a blob and then stores the content of the local file “C:\Users\uxstudyppt\Documents\localfile.txt” in the blob.

Please make use of any async methods that are available to you for this and subsequent tasks.

Task 1a

If you were to run the same code that you wrote in task 1 again and pass in the same container name and same blob name, what do you expect would happen? Why?

Task 1b

Now run your code again and pass in the same container name and blob name and see what happens. Is it what you expected?

Modify your code so that if you pass in the name of a container that already exists, your code will use that existing container instead of creating a new container.

Now that you have this working, notice what happens when you pass in the name of a blob that already exists in the container. Is this what you expected? Why?

Task 1c

Now modify your code so that if you pass in the name of a blob that already exists inside the given container your code does nothing and does not overwrite content in that blob.

**Task 1d**

How do you think you would deal with the case where there were multiple clients trying to access and/or write to this container? How would you expect to handle such concurrent access to the container? In particular imagine a case where two or more clients are processing blobs inside the container, performing some calculations and then updating some container properties based on the processing. How can you be sure that this happens consistently? In other words, how can you prevent one client from processing the content of the container incorrectly because of updates made by another client?

**Task 2**

Add a new function to your project and write code in the function that deletes every empty container in the storage account.

Make sure when you write your code that you use any async methods that are relevant and available to you.

**Task 3**

Add a new function to your project and write code that creates 100 empty containers in the storage account.

Make sure when you write your code that you use any async methods that are relevant and available to you.

**Task 4**

Add a new function to your project and write code in this function that copies the blob at <https://javastoragestudy.blob.core.windows.net/unprocessed/enwik8.pmd> to a new blob in the ‘processed’ container.

This is a large blob (more than 20MB) so this may be a long running operation, depending on how fast the network connection is. Make sure to write your code such that you output some status message to the console (e.g., “Copying blob…”) while the copy operation is still running.

Once the operation has completed, output a message to the console with the number of bytes copied, if the operation was successful. If it was not successful, output a message to the console saying that the operation failed.

**Task 5**

Add a new function to your project and write code in this function that enables public access to blobs inside the container named ‘processed’.

**Task 6**

Add a new function to your project and write code in this function that audits all of the containers in the storage account, checking for those that are publicly accessible. If it finds any, write a message to the console showing the names of the containers that are publicly accessible.