**Chandler Ford | CSS 490 A   
PROGRAM 3 DESIGN and TESTING**

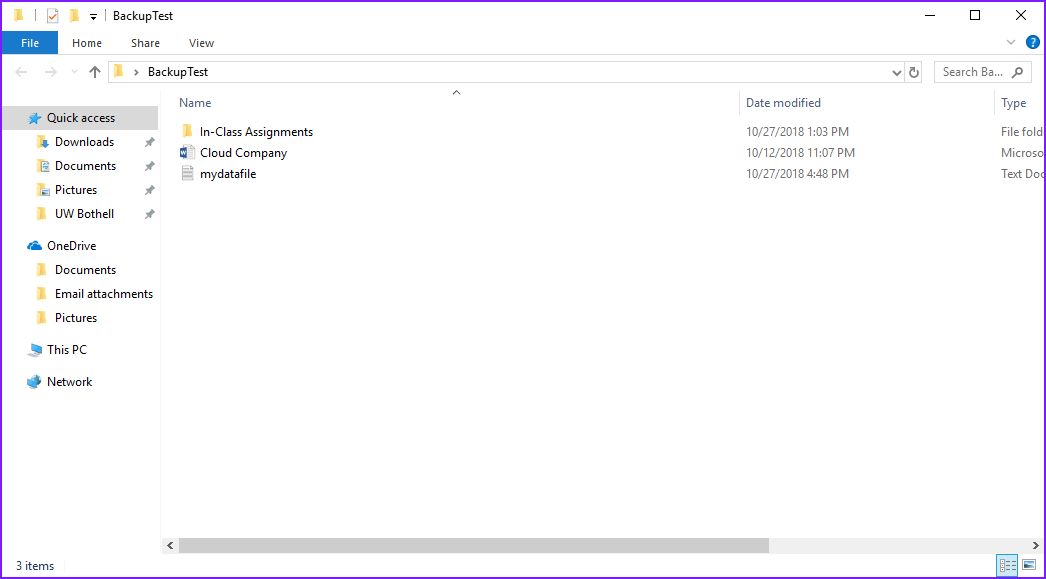
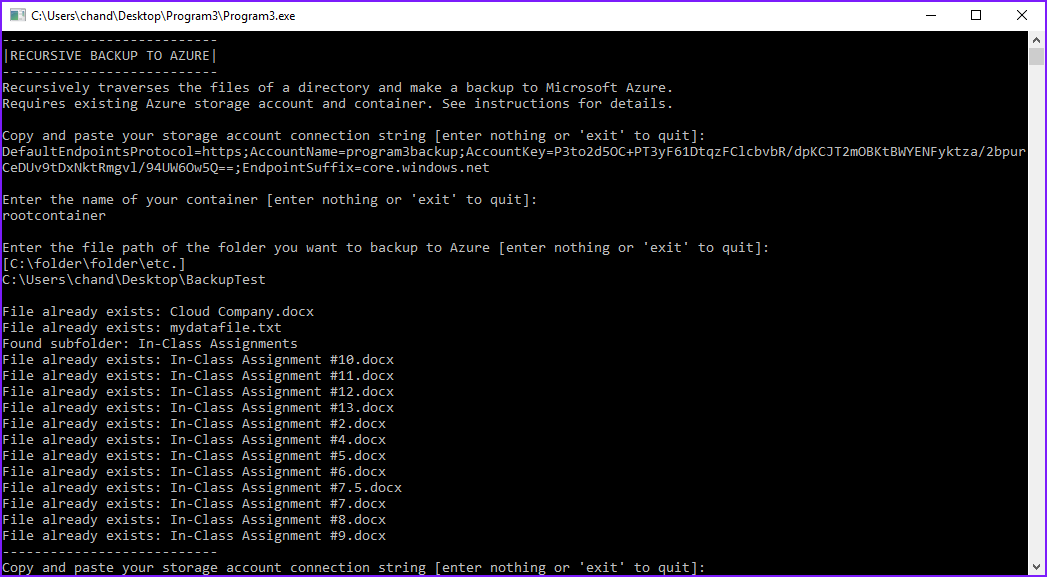
**Design**My program has two methods: main and RecursiveTraverse.

Main starts with some introductory text that explains what the assignment does and that an existing Azure storage account and container are needed. The program then enters the main loop. The user is prompted to copy and paste their storage account connection string before entering the name of their container. The program then connects to Azure by making CloudStorageAccount, CloudBlobClient, and CloudBlobContainer objects. The user enters the file path of the directory they want uploaded to Azure. This string gets put into a DirectoryInfo object that gets put into the initial call of the RecursiveTraverse method along with the container object.

RecursiveTraverse uses arrays to collect the files and subfolders in the current directory. If files exist that haven’t been uploaded to Azure or have been modified since last be uploaded to Azure, the program uploads them to Azure. It prints out the files being uploaded and prints out the names of subfolders. Each subfolder is called recursively using the RecursiveTraverse method.

**Testing**  
I tested the program using my own storage account connection string and its storage container named rootcontainer. I had a test folder named BackupTest located on my desktop. It had multiple files including a subfolder with additional files. Results can be seen by selecting the Storage Explorer option in the storage account submenu and then selecting your container under the “BLOB CONTAINERS” option on the Azure Portal. You may only see the C:/ folder at first- you’ll need to select the correct folders on the path to your test folder to see the files on Azure. See the three pictures below for reference.

**Picture 1 and 2: Program running and contents of test folder**



**Picture 3: Files in Azure container (See below)**

