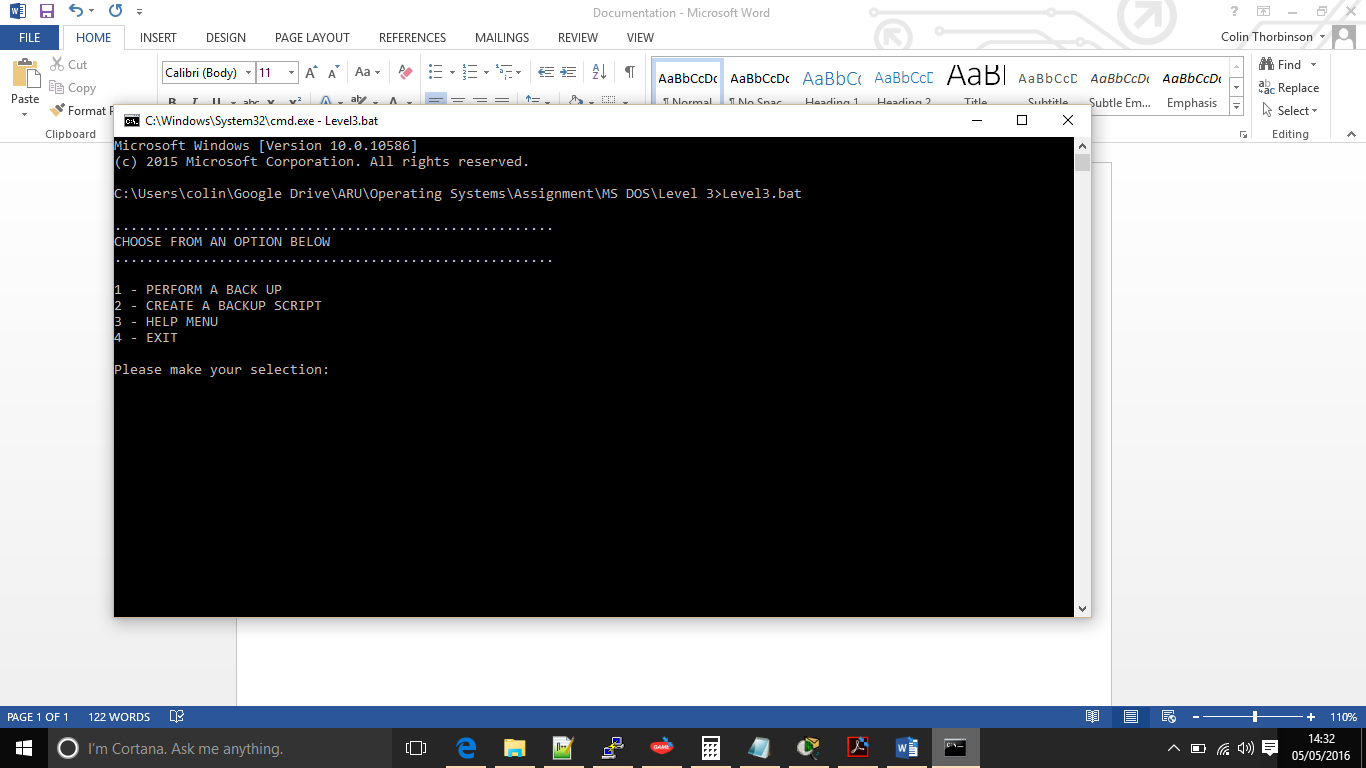
**Back-Up Utility Introduction**

The Back-Up Utility is a program designed to help users easily back up files and folders using the command line. The Back-Up Utility comes in two different files, a batch file for Windows based operating systems and a Bash Shell file for compatible versions of Linux. Users are given the option to run a ‘Quick Back-Up’, run a single back up and the facility to create their own back up scripts.

**Back-Up Utility – Windows MS DOS**

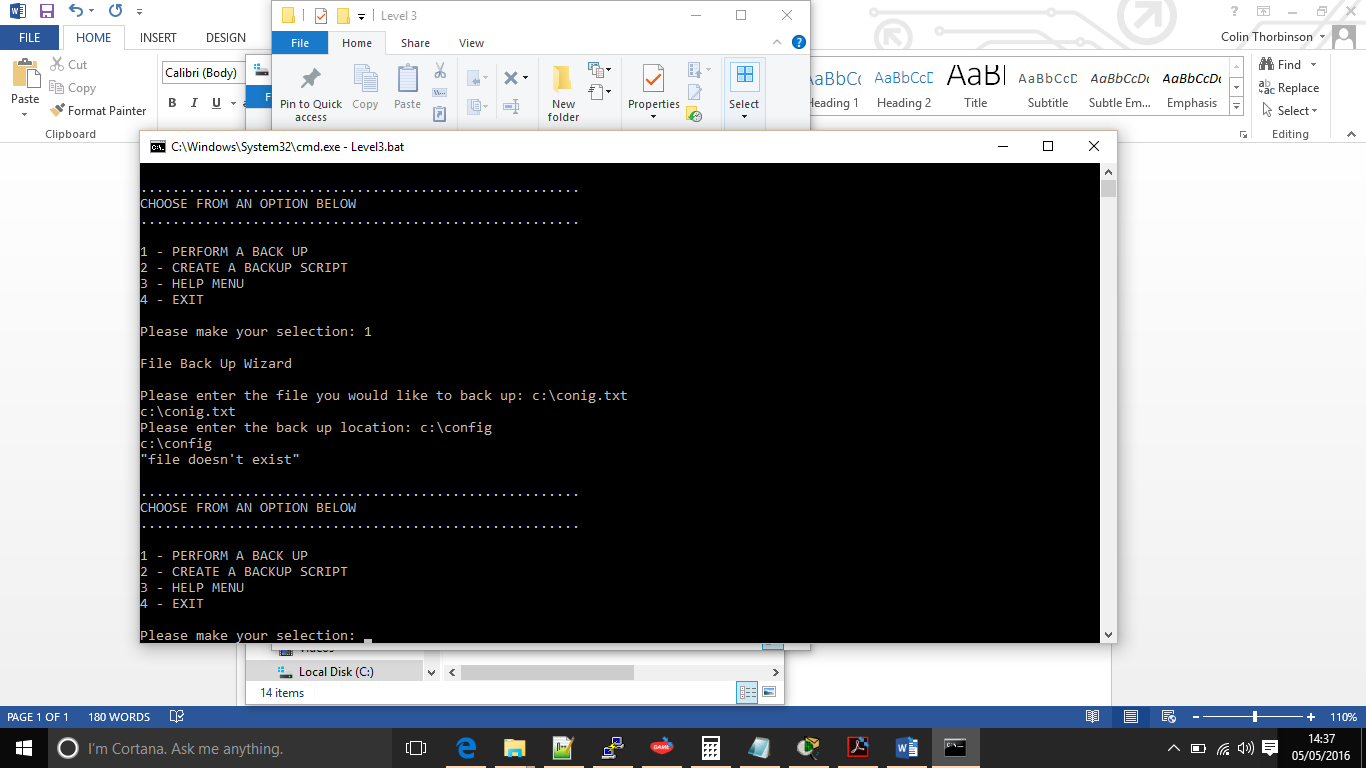
Menu Mode

When users run the program without using the ‘Quick Back-Up’ mode they will see the Main Menu screen. This screen gives users the option to either perform a back-up, create a back-up script, view the help section or exit the program.



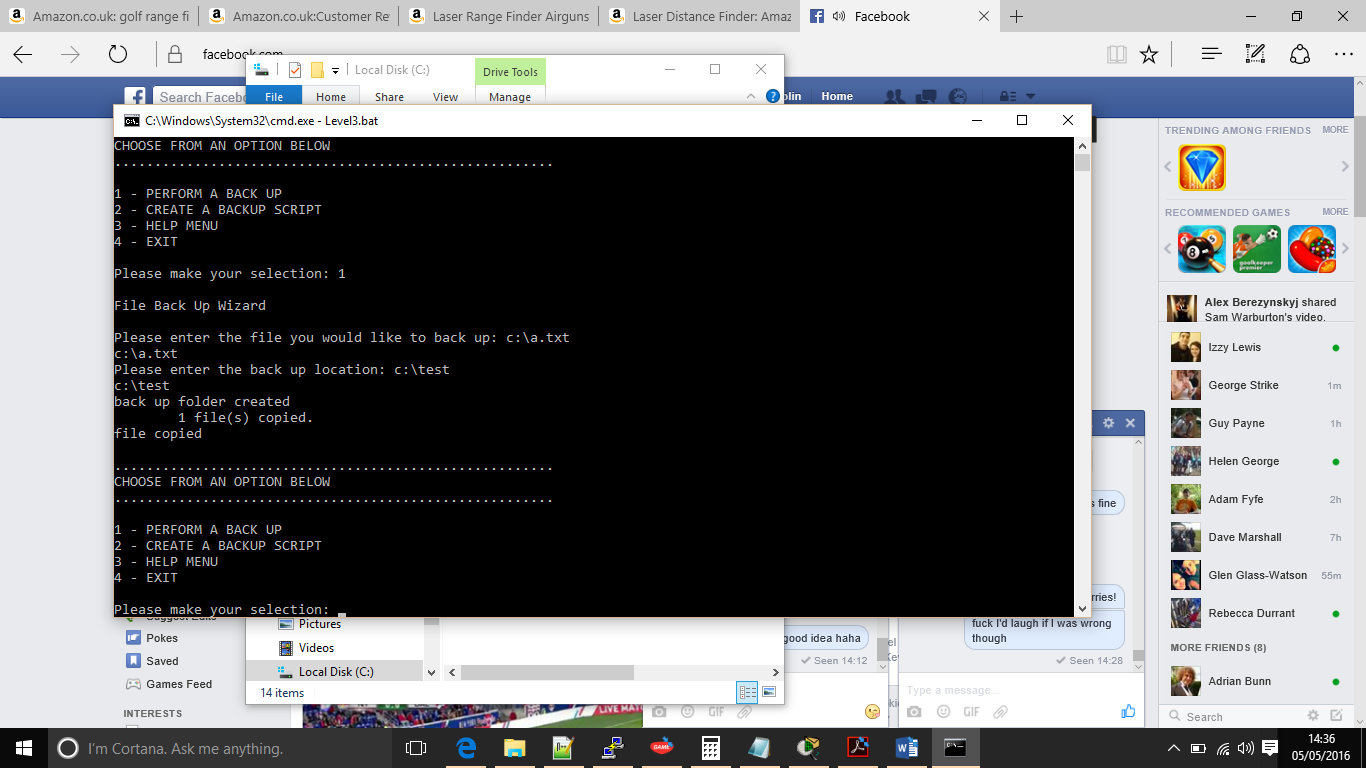
Option One – Perform a Back-Up

When users wish to perform a back-up they should select option one at the Main Menu, They will then be prompted to enter the file they wish to back-up and the back-up destination. Once the user has chosen these options then the program will begin checking that both file and folders exist. If the file does not exist then it will fail displaying the user a message on screen then returning to the main menu (screenshot 1). If the folder does not exist then the program will simply create the folder, once both file and folder the program will proceed and back up the required file (screenshot 2).



Screenshot 1.

Screenshot shows that the user has entered an incorrect file before returning to the Main Menu.

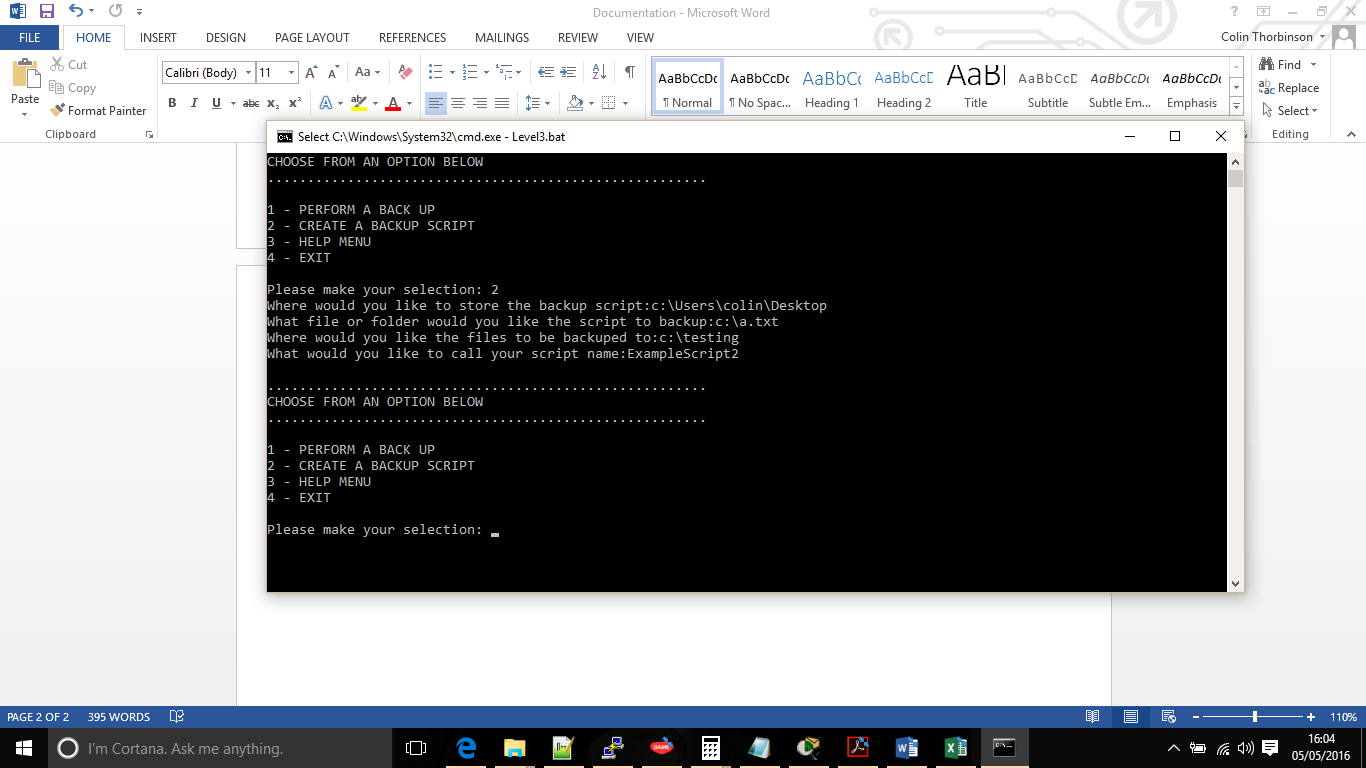


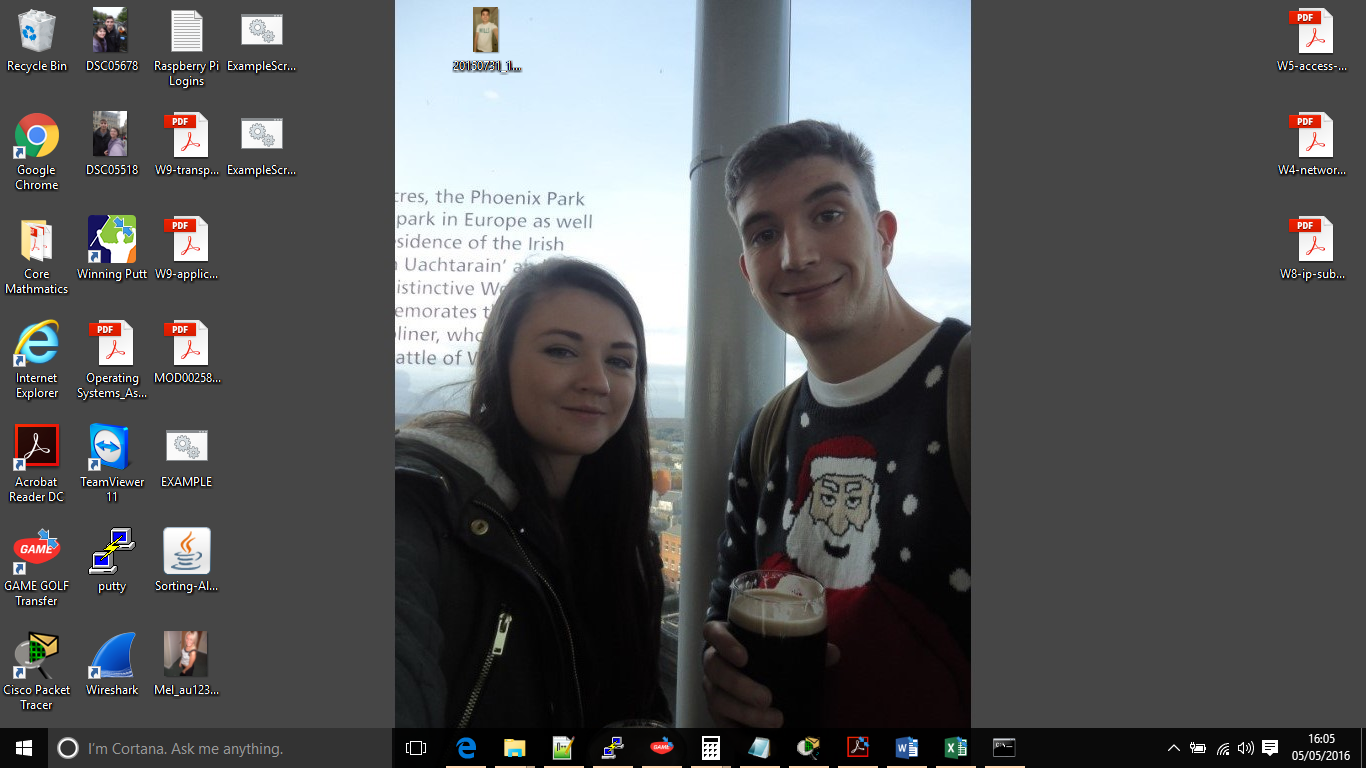
Screenshot 2.

Screenshot shows that the user has entered a correct file but the folder does not exist. Back-Up Utility has created the folder and performed a successful back up.

Option Two – Creating a back-up script

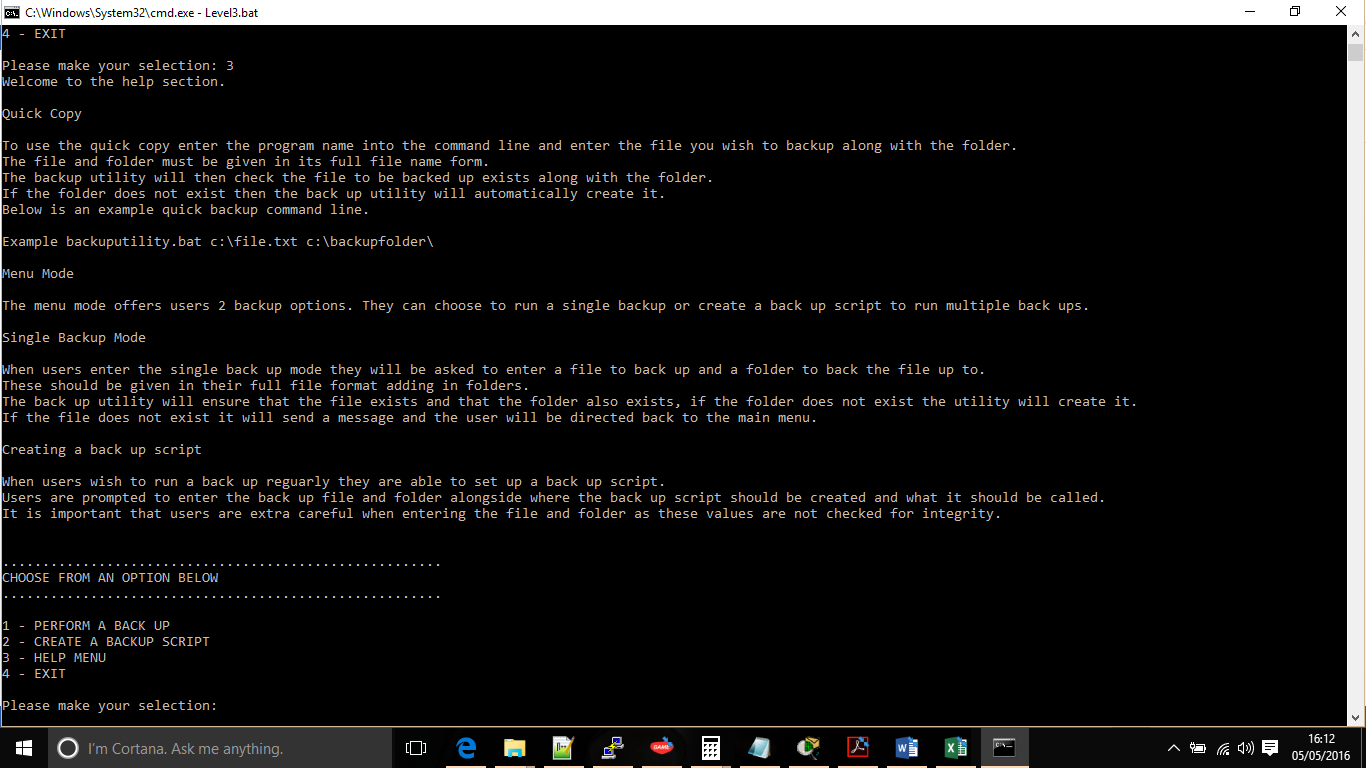
For many users performing back-ups are now routine tasks, if they are not they should be. The back-up utility allows users to create their own back-up scripts that can be ran by running a simple batch script. Like performing a single back-up the user must enter the file they wish to back up along with a folder destination, extra information is required such as where the user would like to store the back-up script and what they would like to call it. Creating a back-up script requires users to be very careful with their input as it does not check to see if the files exist.

Screenshot showing all information that the user must enter to create a backup script.

This screenshot is an example of the scripts created outputted onto this users’ desktop. From here the scripts can be ran by either double clicking or advanced users may wish to add these to a task scheduler so that they will run automatically at set times or dates.

Option Three – Help!

The help function gives users an overview of how the back-up utility should be run and how the program is expecting their given information. This help function should be used alongside this document to fully understand the program.

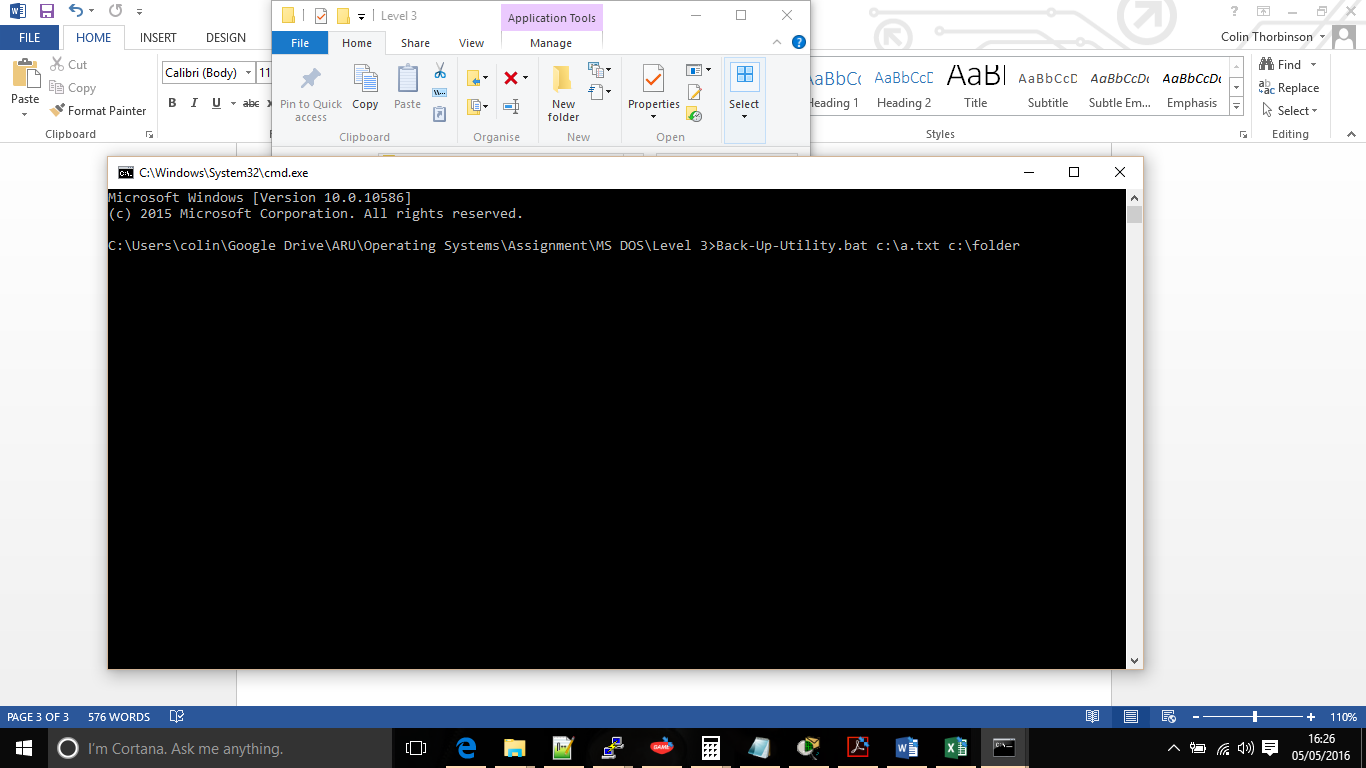
  
Screenshot shows and overview of the help function.

Option Four – Exit

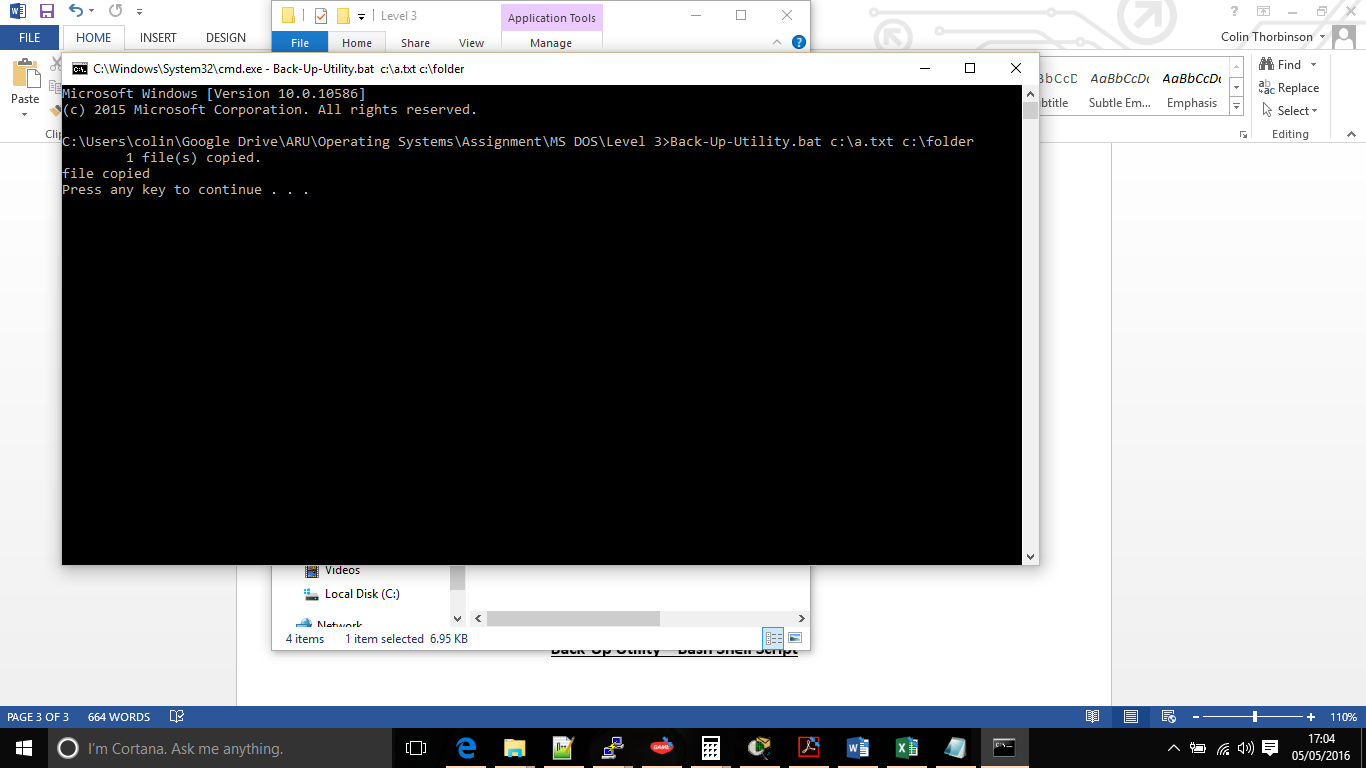
Once users have completed their back-up’s they should select option 4 to exit.

Using Quick Copy

Users have the option of running a back-up without having to use the menu via quick copy. In the command line users must simply add the file they wish to back-up along with the back-up location next to the program name to perform a quick back-up.

Screenshot shows an example of how the syntax should look when users wish to perform a quick back-up.

A quick back-up will check that the back-up file exists, if the file does not exist then the back-up utility will default back to the main menu. The back-up folder will also be checked and if the back-up folder does not exist it will be created. Once these checks are complete the copy will take place, a message will be displayed to the user and the back-up utility will close.



Screenshot shows how the quick copy has been performed, successfully copying then displaying a message to the user before closing the back-up utility.

**Back-Up Utility – Bash Shell Script**