# For batch script program:

The report contains following contents

Complexity 1

Implementing the menu

Complexity 2

Command line backup

Command line arguments

Wrong number of command line arguments

Command line argument to create script

##### Complexity 1

###### Implementing the menu

Implementation of the solution to complexity 1 requires implementing an interactive menu using batch script. To print the menu we need to redirect text to standard output. For example,

Automated Backup Program Task

====================================

1. Backup

2. Create script

3. Quit

To accomplish that we use echo command which prints provided text to standard output. For example, this statement:

echo Automated Backup Program Task

on batch file prints “Automated Backup Program Task” on standard output. This way we can build a menu with as many options as required. Right after this we need to accomplish a second goal which is also part of implementing menu: to take input from users as choice. There is a command in dos shell which can prompt for user input and process it. SET command meets the purpose as it can take value from user and can set a variable. It is why when we add this statement to the batch file after displaying the menu,

set /p userChoice= Enter your choice:

It asks the user to enter a choice. When user enters a character and presses ‘enter’ it is set into the variable userChoice. Depending on the assigned value of userChoice we can run different operations. We do the comparison using if else statement, for example,

if %userChoice% equ 1 (

goto runBackUpMenu

)

Compares value of userChoice variable with ‘1’; if user has entered it program jumps to runBackUpMenu. If the value matches with 3 the program exits.

Implementation of backup

When backup is selected from menu it performs following actions,

* Take source directory from user and set sourcePath variable. It also involves checking if the provided directory exists.
* Take destination directory from user and set destPath variable. It also involves checking if the provided directory exists. If it does not exist it is created.
* Files from source directory are recursively copied to destination to perform the backup.

##### Complexity 2

###### Command line backup

Following operations are performed during command line backup,

* Source directory and destination directory from command line arguments are saved to variables sourcePath and destPath variables.
* Check if the directories exist.
* Files from source directory are recursively copied to destination to perform the backup.

##### Command line arguments

Additionally, it is ensured that command line backup is performed only when total number of command line arguments is 2. To do that we need to calculate total number of arguments provided. Here is the explanation how it works,

We initially set a variable to zero.

set numArgs=0

Then for each of the item in command line arguments (%\* contains all arguments) we increment its value.

for %%x in (%\*) do Set /A numArgs+=1

###### Wrong number of command line arguments

After we have calculated number of command line arguments we can perform rigorous check whether number of command line arguments is two or not. When it is not two we invoke interactive menu. Have a look at following code:

if %numArgs% neq 2 (

echo Number of command line arguments is not two. Invoking interactive menu.

echo.

goto runMenu

)

The menu is called whenever number of arguments is not two.

###### Command line argument to create script

When number of command line arguments is four and user has specified on 3rd argument to create a script:

if %numArgs% equ 4 if "%3"=="/CreateScript" (

goto setScriptFile

)

When both of the cases (previous one and this one) are not met we know that number of command line arguments is two and user has selected to perform backup from command line.

##### Complexity 3

###### How to create batch script

A batch script is a text file. I have already explained in this document that using echo command texts can be redirected toward standard output for displaying. If we redirect standard output to file output for the command statements of a batch file it will generate a script file on the end. For example,

echo echo Performing backup, please wait..> backup.cmd

Writes the statement “echo Performing backup, please wait..” to backup.cmd There are two kinds redirection one “>” which creates a new file and outputs the text to the file and “>>” which appends text to the existing file.

###### Script contents

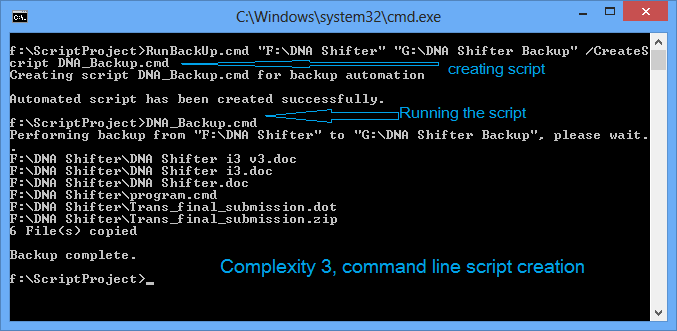
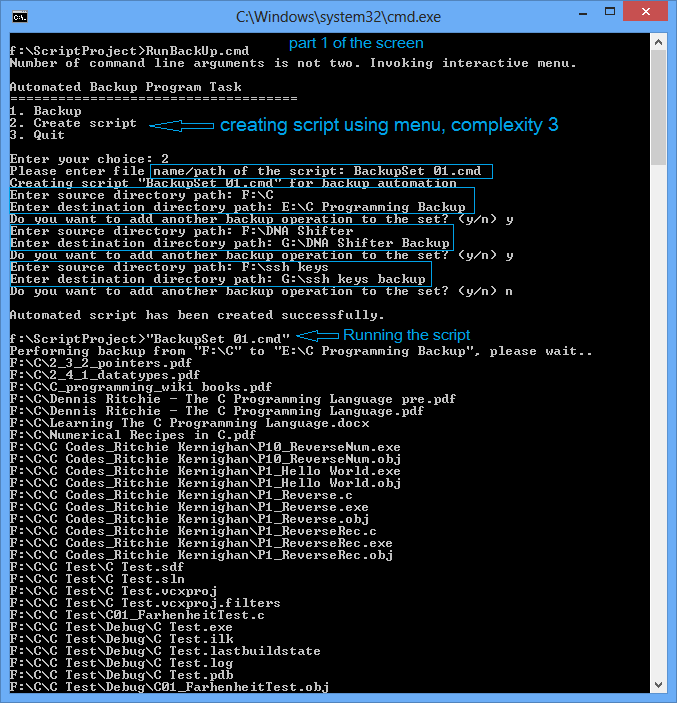
Create script contains 3 simple operations. One is to retain the source and destinations provided by user. Two is to check whether they exist or not. In case of destination directory if it does not exist a new one is created. Three is to perform the backup.

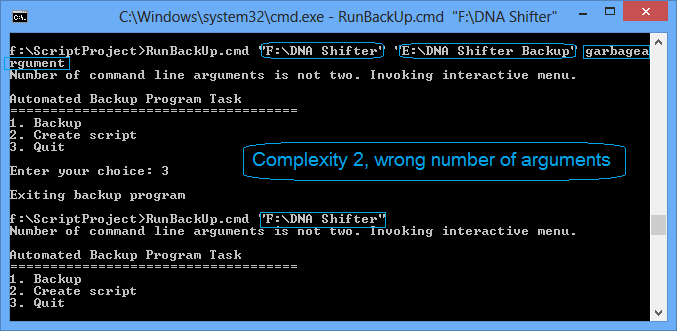
###### Command line batch script creation

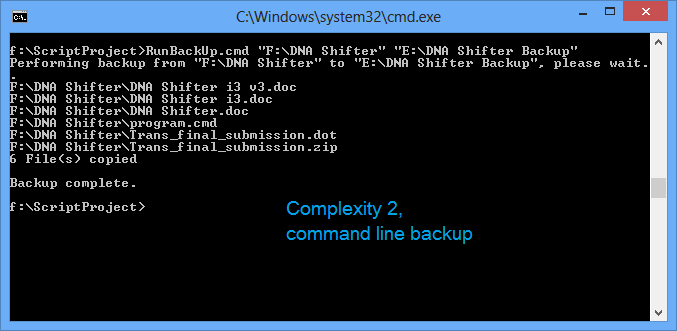
Command line batch script creation is different than menu driven script creation in two ways.

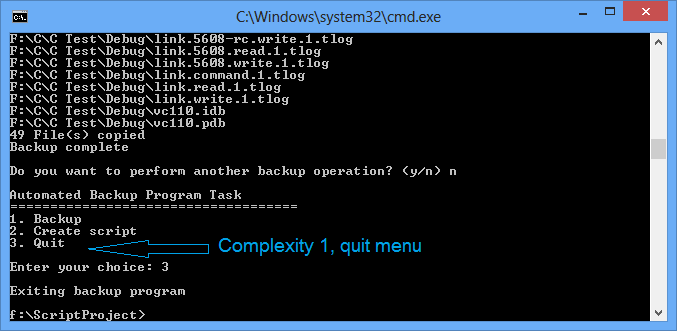
source and destination paths are acquired from command line

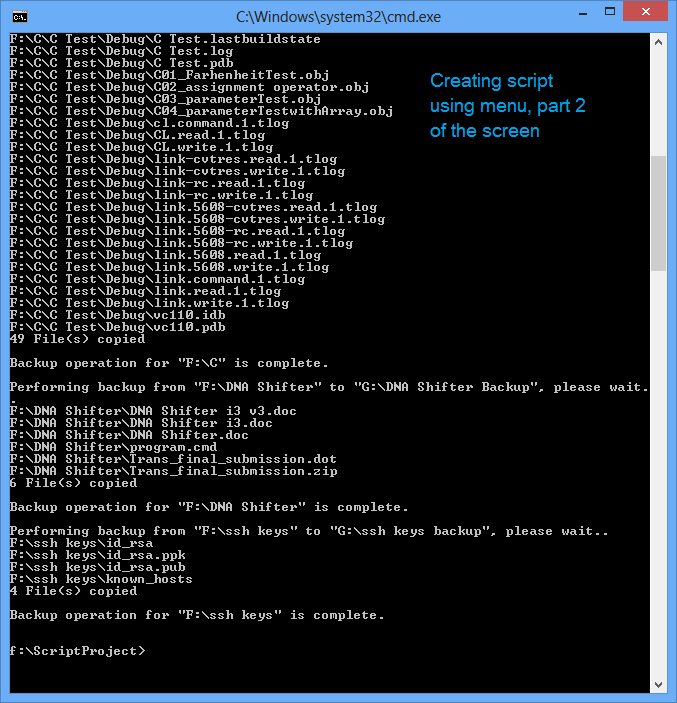
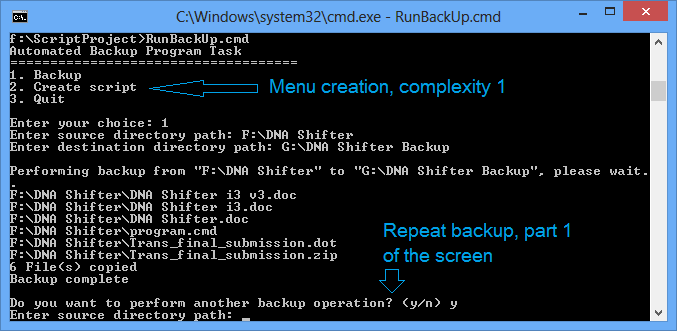
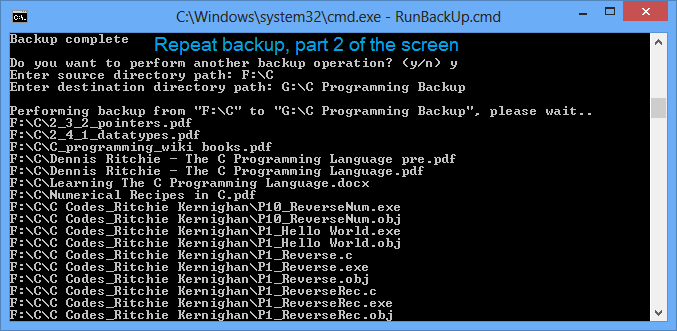
And finally screen shot of the batch program is attached with the doc and annotated below:











# For shell script program:

For shell script program screen shots are given below:

