

IBCS

## File Processing

Mr. Brennan

Introduce WHILE loops and file processing

### Part I

### While Loops and Output Files

Description:

The WHILE..WEND loop  
LINE INPUT  
OPEN file\$ FOR OUTPUT AS #<channel>  
PRINT #, output-to-write

Deliverable:

copyme.bas  
copyme.out

### Part II

### Input Files

Description:

OPEN file\$ FOR INPUT AS #<channel>  
EOF

Deliverable:

copyfile.bas

### Part III

### File Processing

Description:

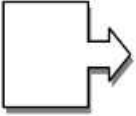


Command Line Arguments  
INSTR  
CLOSE

Deliverable:

scanfile.bas

# Lab 5 File Processing

## Instruction Conventions

	<p>When the “process” icon appears in the box on the left, then this box will contain one or more instructions that you will need to follow.</p> <p>The instructions will be as specific as is practical but could be different for different users and computer configurations.</p>
	<p>When the “note” icon appears in the box to the left, this box will contain notes, hints, or tips that may be helpful to the lab activities.</p>
	<p>&lt;user input&gt;</p>
	<p>When the keyboard icon appears in the box to the left, the box above will contain a line of input to be entered by the user, and this box will contain an explanation of what the user input will do.</p>

## IBCS

## Lab 5 Input Files

## Part II



In part II of this lab you will open a file and read from it, rather than using console (keyboard) input or having the operating system redirect input into your program.

The syntax of the open command is:

```
OPEN file$ FOR mode AS #channel
```

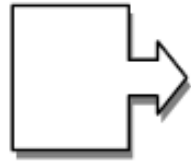
where

file\$ is a string containing the name of the file  
mode is how we are going to open the file, for this  
next step we will open the file for INPUT  
which will not alter the contents of the file  
#Channel is the file number we will use. For the  
input file use channel 1.

## IBCS

## Lab 5 Input Files

## Part II



1. Save your copyme.bas file as copyfile.bas
2. Download the file testcopy.dat [Save Link As or other methods]



3. Edit the copyfile.bas program to prompt the user for the name of an input file to copy.
4. Add a line in your program to open the file for input. For example, if you stored the file name entered by the user in variable F\$ you would create an open statement such as

```
OPEN F$ FOR INPUT AS #1
```

5. Change the LINE INPUT statements to use input channel #1  
LINE INPUT #1, A\$

## IBCS

## Lab 5 Input Files

## Part II



You will not need to prompt the user for data input, the program will get all of the data from the test file. Remove



6. Test your program. When prompted for an input file, tell your program to use `echo.dat`.

This is a test

This is another test, longer than the first one

What if I enter a list, such as apples, pears, grapes

stop program

If your program is successful, then the output file `copyme.out` will contain:

This is a test

This is another test, longer than the first one

What if I enter a list, such as apples, pears, grapes

## IBCS

## Lab 5 Input Files

## Part II



The program stopped copying input when it encountered the string "stop program" but there is a more general purpose way to see if you have reached the End Of a File, and that is the EOF function. A function call to EOF(1) returns TRUE if the end of channel #1 has been reached.



7. Change the WHILE statement to continue until the end of the input file is reached using:  

```
WHILE NOT EOF(1)
```
8. Test your program with the echo.dat file.
9. What changes should you make so even the last line of the input file is copied to the copyme.out file? Make the necessary program changes so the copyme.out file is an exact copy of the test file.

## IBCS

## Lab 5 Input Files

## Part II



10. Test your program with the `testcopy.dat` file.

Check to make sure that the output file from your program, `copyme.out` is an exact copy of `testcopy.dat`.

11. Submit your program `copyfile.bas` for grading. It will be evaluated on working with several test files.

## Attachments

---

echo.bas