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

	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. The instructions will be a specific as is practical but could be different for different users and computer configurations.
	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.
	<user input>
	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.

IBCS Lab 5 File Processing Part III

In Lab 5 Part II you created a program copyfile.bas that prompts the user for the name of a file to copy, and then created a copy of the file specified named copyme.out.

To execute your program you issued the command:

```
c:>blassic copyfile.bas
```

If past labs you have used input redirection to have the operating system get program input from a file, rather than the keyboard, and pass it to your program. You have also used output redirection to tell the operating system to redirect output to a file, rather than to the users screen.

IBCS

Lab 5 File Processing

Part III

In this lab, you will process command line arguments that the user specifies when the run your program. You will change your program to use file names for both the input file and the output file that the user specifies when they run your program.

This is a sample command line to run your program

```
c:>blassic copyfile.bas testcopy.dat copyme.out
```

This tells your program to use testcopy.dat as the input file, and copyme.out as the output file.
In general, the syntax for running your program is:

```
c:>blassic copyfile.bas <inputfile>  <outputfile>
```

IBCS

Lab 5 File Processing

Part III

Operating System Notes

When you issue the following command line

```
c:>blassic copyfile.bas testcopy.dat copyme.out
```

The operating system executes program blassic.com. It creates an array of command line arguments that were also specified, and the operating system passes that array to the blassic program. Blassic then creates an array of strings that your program can access.

The name of the array containing command line arguments is **programarg$**

When you issue the following command line

```
c:>blassic copyfile.bas testcopy.dat copyme.out
```

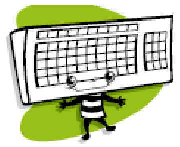
programarg$(1) will contain the name of the input file testcopy.bas
programarg$(2) will contain the name of the output file copyme,out

When you write basic program you can use the **programarg$** array to access command line parameters.

IBCS

Lab 5 File Processing

Part III



1. Change your copyfile.bas program to use the first command line argument as the name of the input file, no need to prompt the user for an input file name.

2. Change the program to use the second command line parameter for the name of the output file, so it does not have to be name copyme.out.

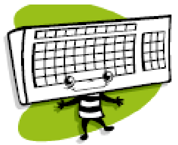
3. It is good programming practice that when a program is done with a file that it has opened, that prior to the program terminating, the program should close the file. You can close each of the files by adding the following two lines to the end of your program:

```
CLOSE #1     : REM Close the input file
CLOSE #2     : REM Close the output file
```

IBCS

Lab 5 File Processing

Part III



4. Test your program using the following command line:

```
c:>blassic copyfile.bas testcopy.dat copyme.out
```

When you are satisfied that your program is working, continue to the next step.

Make some note about the type of error checking that might be appropriate for this program.

IBCS

Lab 5 File Processing

Part III

Your program now reads the contents of a file and writes it to a new file. Now we will add some processing to the input file before writing it to the output file.

You will change your program so that it will read an input file, one line at a time, but only print to the output file the lines that contain the word software.

To do this, you will use the function INSTR which will search one string for the occurrence of another. In this case, we are looking for the word software.

IBCS

Lab 5 File Processing

Part III

If you have a line of text in variable A$ and you want to know the position of the first occurrence of the word software you could use the following command:

```
j = INSTR("I am a software engineer", "software")
```

The value 8 is assigned to variable j. If the word software did not occur in the first string, such as

```
j = INSTR("I am a soft-ware engineer", "software")
```

then the value of j would be 0.

IBCS

Lab 5 File Processing

Part III

There are two general forms of the INSTR function:

INSTR(str1$, str2$) return the position of the first occurrence of str2$ in str1$, or 0 if str2$ is not in str1$.

INSTR(n, str1$, str2$) return the position of the first occurrence of str2$ in str1$ that begins searching at position n in str1$ or 0 if str2$ is not in str1$ past n.

Example:

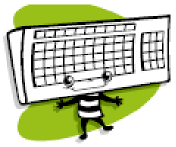
```
p = INSTR(10, "I like software, developing software is fun", "software")
```

Assigns the value 29 to variable p. This format of the instr function is good when you need to process all occurrences of a string.

IBCS

Lab 5 File Processing

Part III



5. Save your program with the new name findstr.bas

6. Change your program to only copy lines from the input file to the output file the lines that contain the word software.

   When you are satisfied that your program is working, continue to the next step.

7. Change your program to use a third command line parameter, a string to search for. Rather than searching for the word software search for the string specified by the third command line argument.

IBCS

Lab 5 File Processing

Part III

9. Test your program using a variety of input files and search strings.
   Suggestions:
   search for NAME in your bas programs to make sure that you have updated them.
   search for WHILE to see if a particular program uses a WHILE loop.

10. Submit your program findstr.bas

IBCS

Lab 5 File Processing

Part III

Attachments

echo.bas