Introduction to Scripting

Chapter 15

This presentation covers:

- Changing perspective when troubleshooting
- Command line basics
- Scripting basics

Qualities of a Good Technician

“Soft skills” as they are known across many industries are essential

Changing Perspective When Troubleshooting

Sometimes when you are troubleshooting, things get frustrating. Try these tips when that happens:

1. Put yourself in the user’s shoes – think like the user instead of like a technician.
2. Ask yourself what another technician would do.
3. Think back to a similar problem and what you did to solve it – experience is a great teacher.

Command Line Basics

Command Prompt Overview

- When an operating system does not work, the technician must input commands from a prompt
- Commands are used to bring up a Windows tool
- Various ways to access a command prompt when the computer is functional:
  - Access the *Search* function > type `cmd` and press *Enter*
  - Access the *Search* function > type `command` and press *Enter*; note that when this option is used, the keyboard arrow keys do not bring up previously used commands as they do when using `cmd`
  - Access *Accessories* > *Command Prompt* (7 and 10)
  - Access the *Command Prompt* tile (Windows 8/10)

Command Prompt Privileges

- Standard privileges allow users to do some basic commands.
- Administrative privileges allow technicians to do commands that users are not allowed to do.
  - Require elevated privileges
  - Right-click on Command Prompt > *Run as administrator*

Remember

Command Prompt Basics

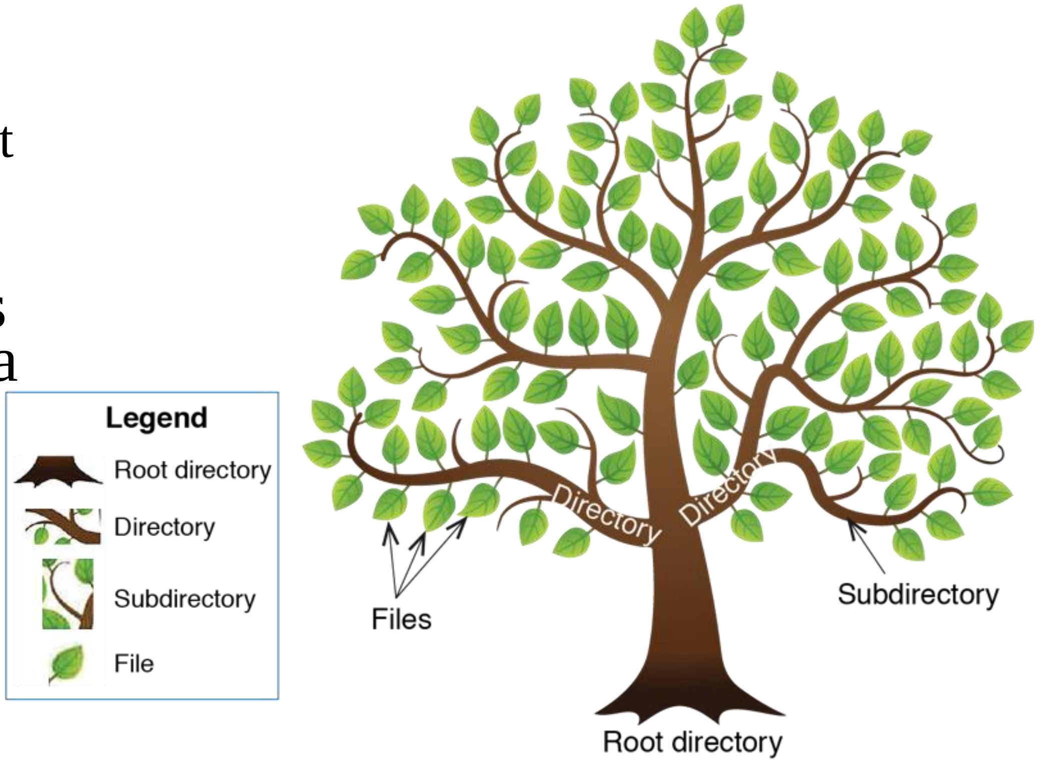
> Drive letters are assigned to hardware devices when a computer boots; e.g. the first hard drive partition gets the drive letter C:. The colon is part of the device drive letter

> All communication using typed commands begins at the command prompt, or simply a prompt; e.g. F:\> or C:\> or C:\Windows>

> File groupings are called a folder (GUI environment) or a directory (command prompt environment)

> The starting point for all directories is the **root directory**

> A subdirectory is created beneath another directory



Tree Structure Concepts

Another Prospective of a Tree Structure

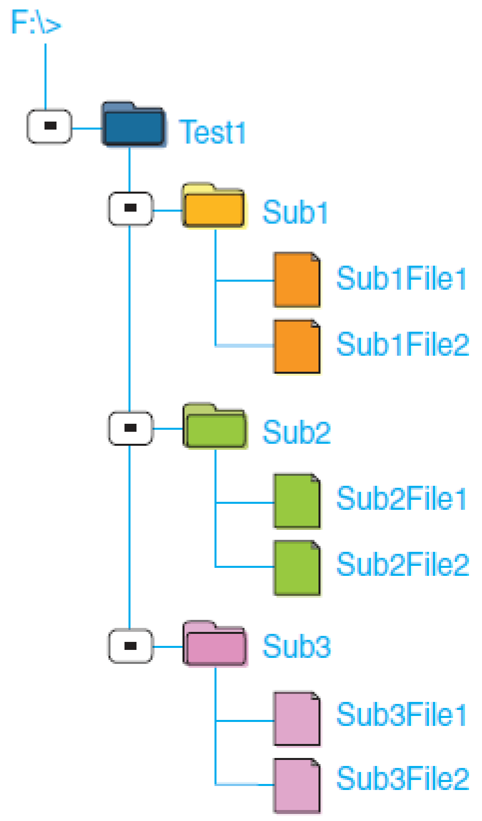
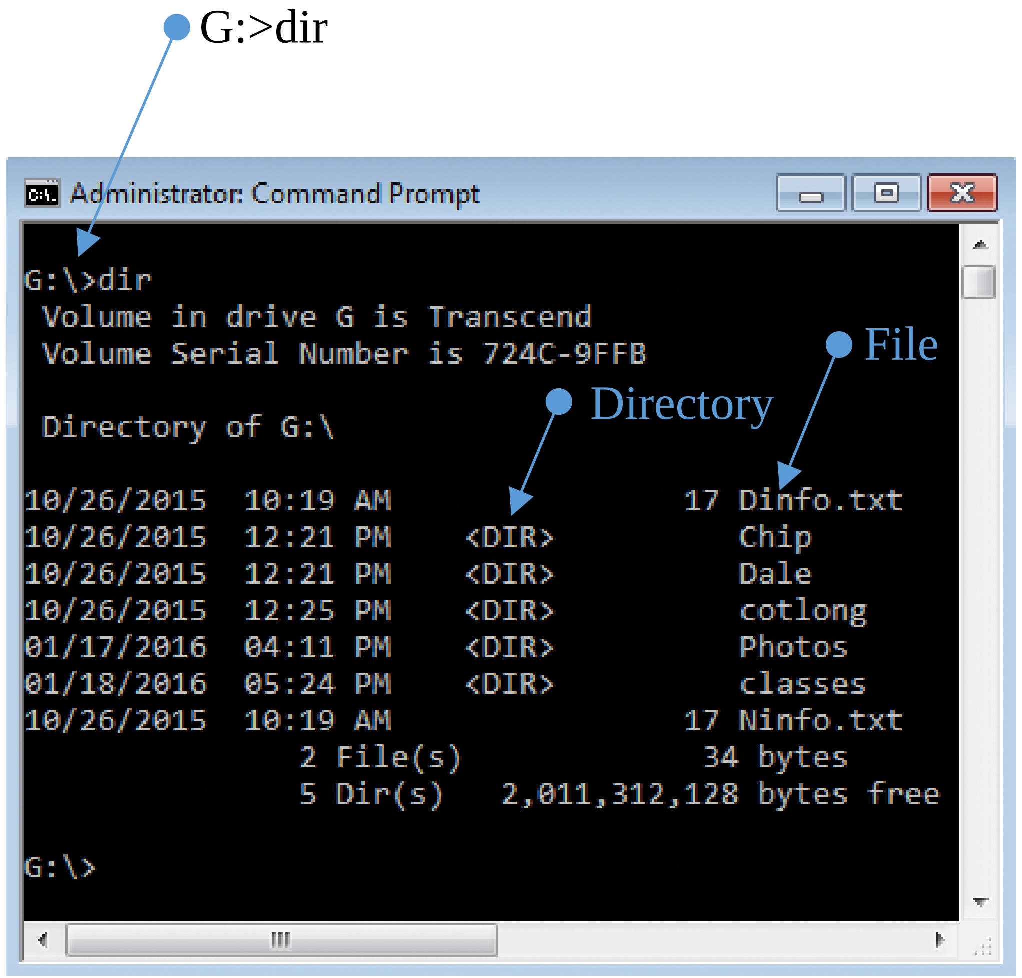


FIGURE 15.5 Sample tree structure

The `DIR` Command

- The `dir` lists all the files and directories from wherever you are at the prompt
- The image shows the `dir` command from the root directory of a flash drive (`G:\>`)

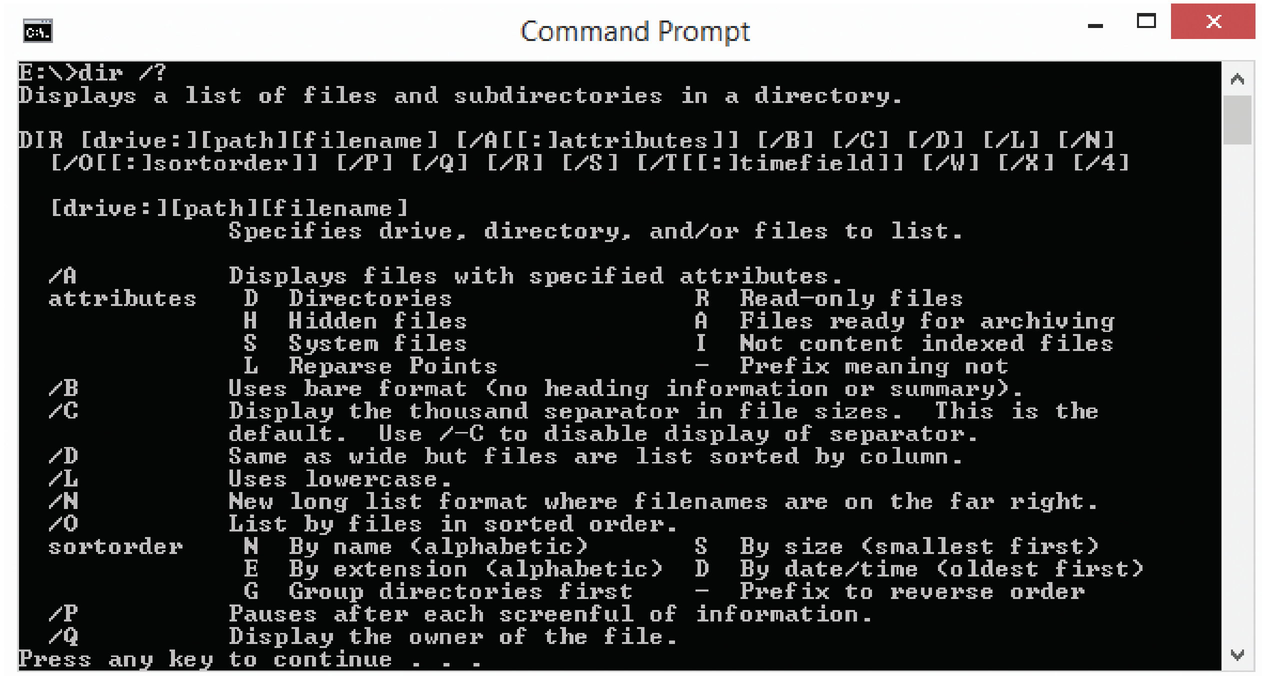


```
G:\>dir
 Volume in drive G is Transcend
 Volume Serial Number is 724C-9FFB

 Directory of G:\

10/26/2015  10:19 AM                17 Dinfo.txt
10/26/2015  12:21 PM    <DIR>          Chip
10/26/2015  12:21 PM    <DIR>          Dale
10/26/2015  12:25 PM    <DIR>          cotlong
01/17/2016  04:11 PM    <DIR>          Photos
01/18/2016  05:24 PM    <DIR>          classes
10/26/2015  10:19 AM                17 Ninfo.txt
               2 File(s)             34 bytes
               5 Dir(s)   2,011,312,128 bytes free

G:\>
```

Other Common Commands

- `md`: make directory
- `del`: delete
- `type`: text (.txt) or batch (.bat)
- Copying Files: `copy`, `xcopy`, and `robocopy`
  - copy command is used to make a duplicate of a file
  - xcopy command is used to copy and back up files and directories
  - robocopy command enables you to copy a directory, its contents, all its subdirectories (and their subdirectories), as well as each attribute
- `attrib` Command: sets, removes, or shows the attribute of a file or a directory

Command Switches – Options for Commands



```
E:\>dir /?
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/A[[:]attributes]] [/B] [/C] [/D] [/L] [/N]
  [/O[[:]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:]timefield]] [/W] [/X] [/4]

  [drive:][path][filename]
              Specifies drive, directory, and/or files to list.

  /A          Displays files with specified attributes.
  attributes   D  Directories                R  Read-only files
               H  Hidden files               A  Files ready for archiving
               S  System files               I  Not content indexed files
               L  Reparse Points             -  Prefix meaning not
  /B          Uses bare format (no heading information or summary).
  /C          Display the thousand separator in file sizes.  This is the
              default.  Use /-C to disable display of separator.
  /D          Same as wide but files are list sorted by column.
  /L          Uses lowercase.
  /N          New long list format where filenames are on the far right.
  /O          List by files in sorted order.
  sortorder    N  By name (alphabetic)       S  By size (smallest first)
               E  By extension (alphabetic)  D  By date/time (oldest first)
               G  Group directories first    -  Prefix to reverse order
  /P          Pauses after each screenful of information.
  /Q          Display the owner of the file.
Press any key to continue . . .
```

Other Commands You Should Review

- [command name] /?
- ..
- cd
- chkdsk
- command
- copy
- dir
- dism
- dxdiag
- exit
- expand
- Explorer
- format
- gpresult
- gpupdate
- Help
- ipconfig
- md
- mmc
- msconfig
- msinfo32
- mstsc
- net use
- net user
- netstat
- netdom
- nslookup
- ping
- rd
- regedit
- regsvr32
- robocopy
- services.msc
- sfc
- shutdown
- taskkill
- tracert
- xcopy

Scripting Basics

Why Learn Scripting

- A script is a group of commands in a file that automate a particular task.
- Benefits
  - Saves time
  - Ensures consistent operation
  - Provides flexibility
- Scripts are interpreted – carried out one line at a time
  - Contrast with compiled which is a program that has to be turned into machine language before it can execute.

Scripts

- Scripts can be written in different languages.
  - Batch file – file ends in .bat
  - PowerShell – file ends in .ps1
  - Linux shell script – file ends in .sh
  - Python – file ends in .py
  - VBScript – file ends in .vbs
  - JavaScript – file ends in .js

Environment Variables

- Controls the environment in which a program runs
- In Windows, the environment variable has a name and a value
  - windir – C:\Windows (windir is the variable that represents the Windows program. The value is C:\Windows.
  - path – The environment variable that tells a program all of the directories or subdirectories of where to look for specific files associated with the program.
  - View using *System* Control Panel > *Advanced system settings* > *Advanced* tab > *Environment Variables* button
- Two types
  - System – global and cannot be changed; used by all user accounts
  - User – settings specific to a user such as where temp files are stored

Syntax

- The set of symbols or rules used in the type of script being used.
  - Python: `print (“Hello, my friend!”)`
  - JavaScript: `console.log (“Hello, my friend!”);`
  - Batch file: `echo Hello, my friend!`
  - Shell script: `echo “Hello, my friend!”`

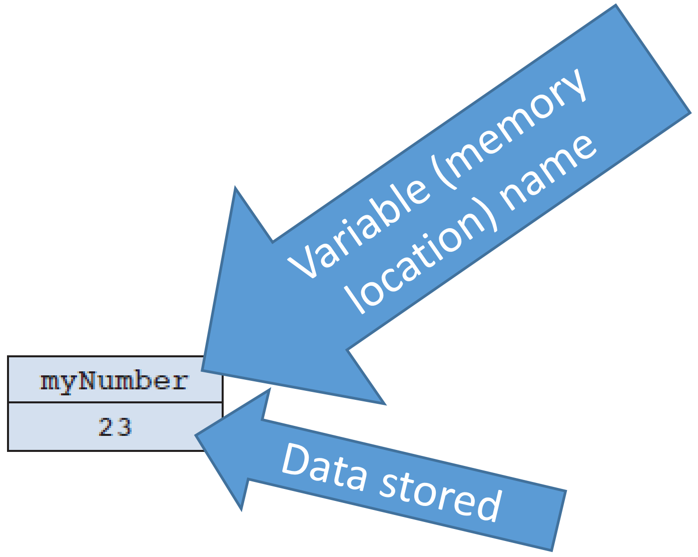
Constructs

The Basics

- Sequence – code is executed one line at a time from top to bottom
- Selection – certain lines of code are run if a specific condition is met
- Repetition (iteration) – certain lines of code can be executed repeatedly

Program Variable

- A named memory location which stores data of a specific type
  - Integer
  - Floating point number
  - Text
- The *value* of that variable is the data contents at that specific memory location.
- Called a variable because its value can change (vary) as the program runs
- It is mailbox – variable is the name on the mailbox and the value of the variable is the notice someone put in the mailbox.



Declaring a Variable

Set the variable to a beginning value

- Initialization
  - Python: myVarName = 0
  - JavaScript: var myVarName = 0
  - Batch file: set myVarName=0
  - Shell script: myVarName=0

Data Types

- Strings – numbers dealt with as text
- Integers – whole numbers (0 and negative numbers too)
- Floating-point numbers – number that can be written in the form of x÷y
  - A number that includes a decimal value
  - Not used in scripts much
- Alphanumeric characters – 0 through 9, punctuation marks, and symbols
  - Must be supported by the program
  - Each program supports different alphanumeric characters

Variable Examples

> Python:

```
num1 = 3
num2 = 4
numResult = num1 + num2
print(numResult)
```

> JavaScript:

```
var num1 = 3;
var num2 = 4;
var numResult = num1 + num2;
console.log(numResult);
```

> Batch files:

```
set num1=3
set num2=4
set numResult=%num1%+%num2%
Echo %numResult%
```

> Shell scripts:

```
set num1=3
set num2=4
set /anumResult=%num1%+%num2%
echo $numResult
```

Variable Examples

> Python:

```
username = "Joey Jones"
message = "Welcome "
result = message + username
print(result)
```

> JavaScript:

```
var username = "Joey Jones";
var message = "Welcome ";
var result = message + username;
console.log(result);
```

> Batch files:

```
set username=Joey Jones
set message=Welcome
set result=message+" "+username
Echo %result%
```

> Shell scripts:

```
set username="Joey Jones"
set message="Welcome "
set result=$message$username
echo $result
```

Comments

- Explain what part of the script is supposed to do and ignored by the program executing the script
- Python – One or more lines; start with # (hash character)
- JavaScript – One line that starts with // (two slashes) or if you want multiple lines start and end with /*
- Batch files – Either put the command `rem` or use :: (two colons)
- Shell – Starts with # (hash character)

Operators Used in Structures

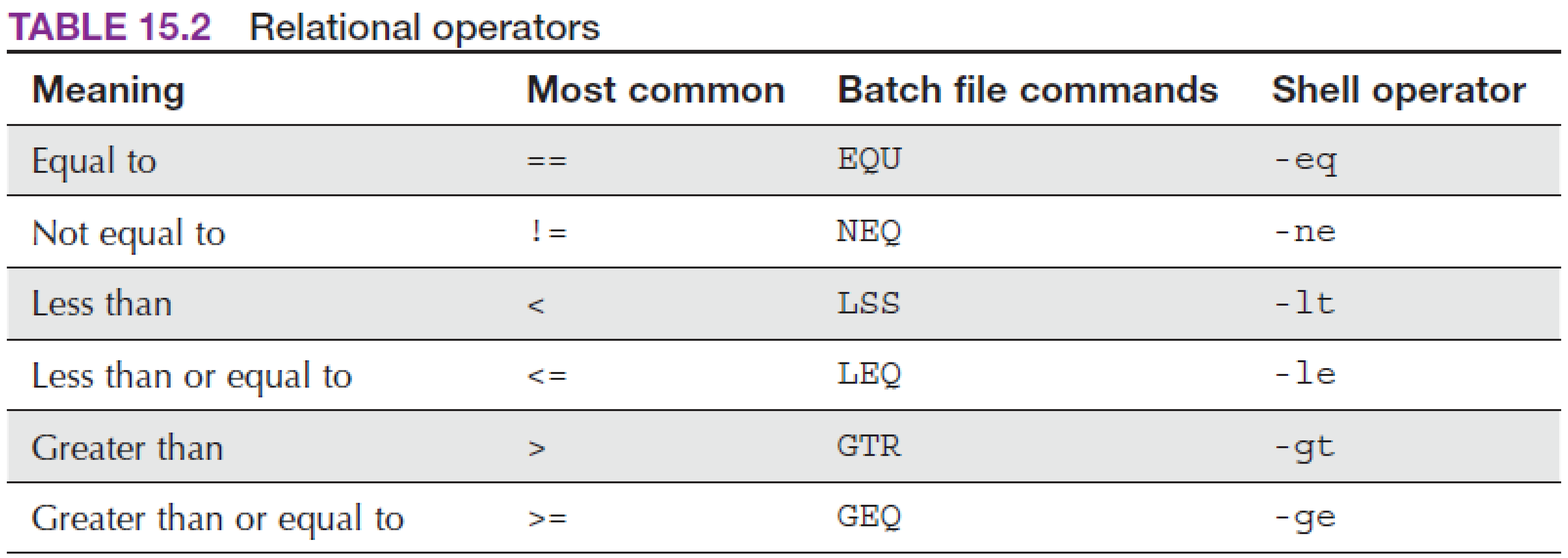


TABLE 15.2 Relational operators

Meaning	Most common	Batch file commands	Shell operator
Equal to	`==`	`EQU`	`-eq`
Not equal to	`!=`	`NEQ`	`-ne`
Less than	`<`	`LSS`	`-lt`
Less than or equal to	`<=`	`LEQ`	`-le`
Greater than	`>`	`GTR`	`-gt`
Greater than or equal to	`>=`	`GEQ`	`-ge`

Selection Structure – Single Alternative

- A test is performed
  - If true – the test condition is met, the next line of code is executed
  - If not true, the program skips down to the next command

```
if name == "Jonas":
    print("user found!")
continue...
```

- == (double equals sign) is a comparison operator (equal operator or equality operator).
  - For most programming a single equals sign sets the value on the right side to the variable name on the left side.
  - A double equals sign compares the two values.

Selection Structure – Dual Alternative

- A test is performed
  - One of two blocks of statements will always execute.
    - If true, one block of statements and then go to next instruction
      after the two blocks
    - If not true, a different block of statements is executed and then goes to the next instruction after the two blocks

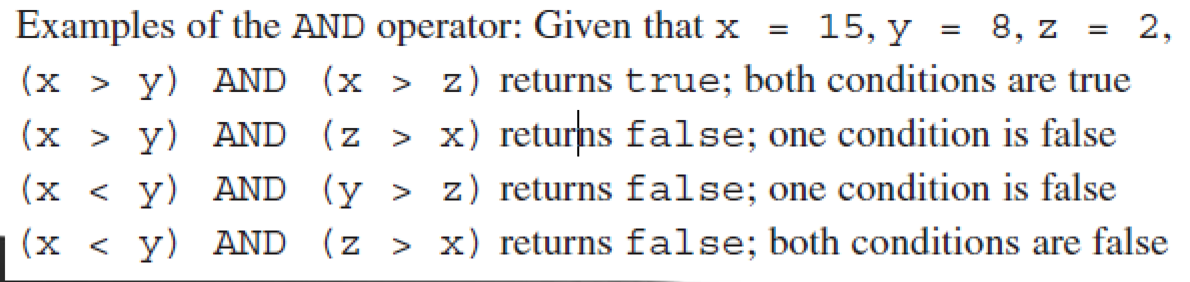
```
if name == "Jonas":
    print("user found!")
else:
    print("user not found!")
continue...
```

Selection Structure – Multiple Alternative

```
if score >= 90:
    print("Congratulations! You have a grade of A.")
elif score >= 80:
    print("You have a grade of B.")
elif score >= 70:
    print("You have a grade of C.")
elif score >= 60:
    print("You have a grade of D.")
else:
    print("You have a grade of F.")
continue...
```

Compound Conditions and Logical Operators

- The AND operator returns true if and only if both expressions (conditions) are true.
- The OR operator returns false if and only if both expressions (conditions) are false. If either expression (condition) is true, then the OR operator returns true.
- The NOT operator simply flips the result of an expression. If an expression (condition) is true, it returns false, and if the expression (condition) is false, it returns true. If not true, a different block of statements is executed and then goes to the next instruction after the two blocks



Examples of the AND operator: Given that `x = 15`, `y = 8`, `z = 2`,

`(x > y) AND (x > z)` returns `true`; both conditions are true

`(x > y) AND (z > x)` returns `false`; one condition is false

`(x < y) AND (y > z)` returns `false`; one condition is false

`(x < y) AND (z > x)` returns `false`; both conditions are false

Loops

- A block of statements executed repeatedly.
- While loop – Starts with the word “while” and a test condition. If the condition is true, the loop begins and repeats until the condition is no longer true.
- For loop – Repeats a block of instructions a specific number of times; a shorthand way to write a while loop

Python program that counts by 5s

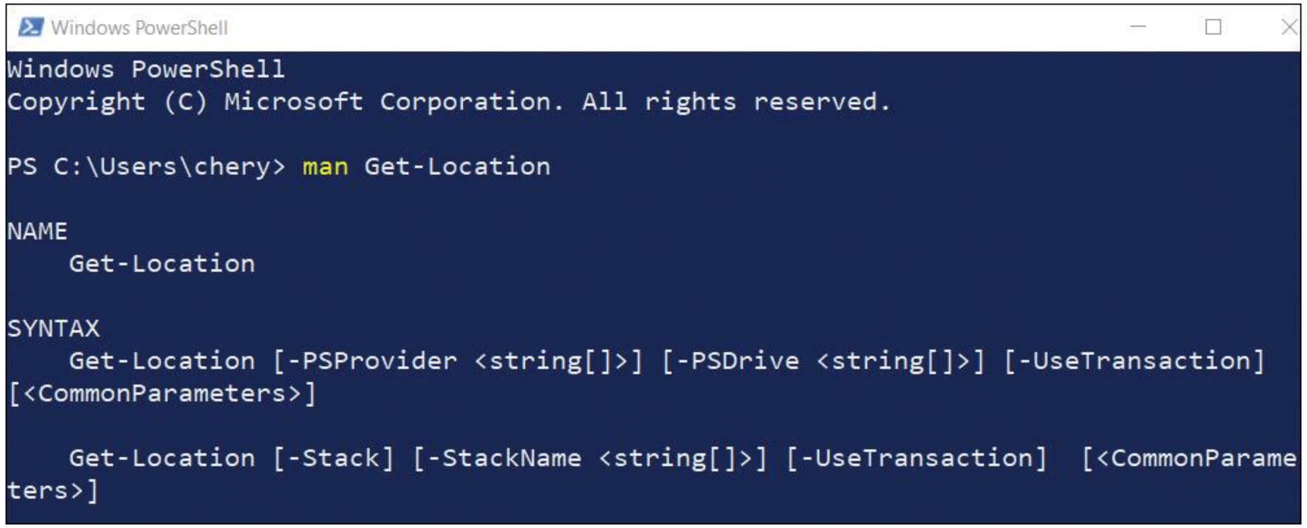
```
x = 5
count = 1
while (count < 11, end = "  "):
    print(count * x)
    count++
```

Java program that counts by 5s

```
var x = 5;
    for (count = 1; count < 11; count++)    {
        console.log(count * x + "  ");
    }
```

PowerShell

- Microsoft's open source cross-platform
- Uses cmdlets
  - Get-Location – gets current directory
  - Move-item – moves a file to a different location
  - New-item – creates a new file



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\chery> man Get-Location

NAME
    Get-Location

SYNTAX
    Get-Location [-PSProvider <string[]>] [-PSDrive <string[]>] [-UseTransaction]
[<CommonParameters>]

    Get-Location [-Stack] [-StackName <string[]>] [-UseTransaction]  [<CommonParame
ters>]
```

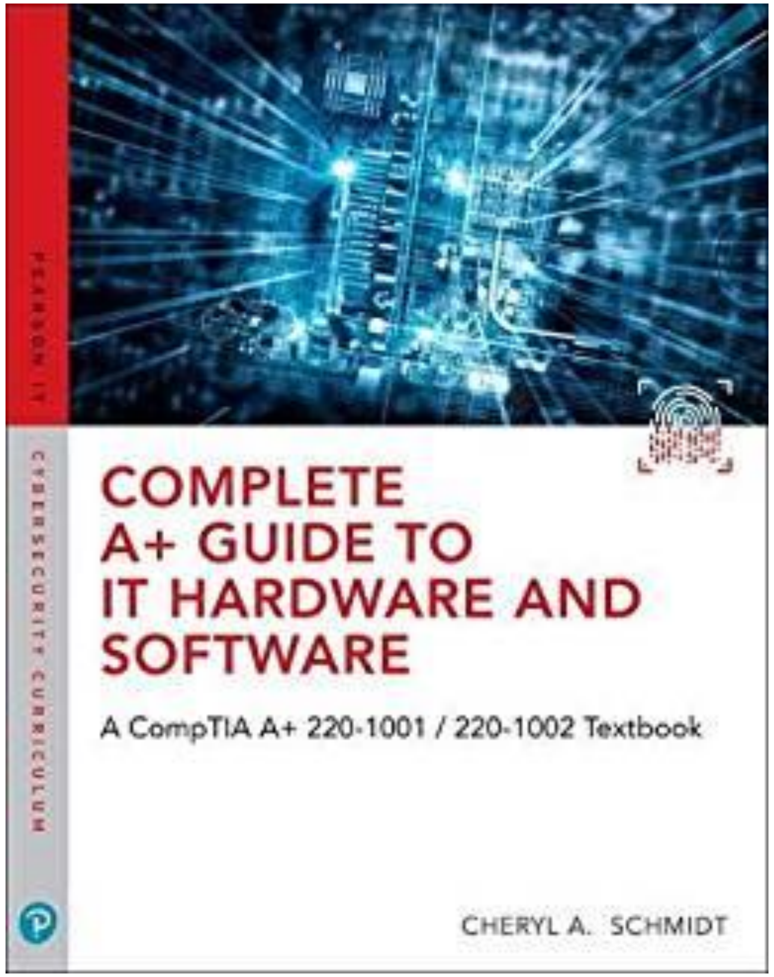
Computer Terms

Refer to the glossary terms at the end of the textbook chapter. Review Chapter 15 and become familiar with the terms.

This PPT deck was developed

to support instruction of

The Complete CompTIA A+ Guide to IT Hardware and Software 8th Ed.


PEARSON IT
CYBERSECURITY CURRICULUM
COMPLETE A+ GUIDE TO IT HARDWARE AND SOFTWARE
A CompTIA A+ 220-1001 / 220-1002 Textbook
CHERYL A. SCHMIDT