Advanced Scripting   
Other Operators

Last Updated: 4/19/2024 6:01 PM Version 1  
Document Prepared for: CYBER360 Student

# Name Click here to enter name ID Click here to enter id

# Instructions

Answer all questions directly in this document. You will save and upload this completed document as your homework submission.

# Overview

This exercise presents a few practice drills involving PowerShell operators.

# Setup

## Requirements

* PowerShell

# Task 1—Property and Method Operators

## Steps

1. The . “dot” operator is used to access an object’s instance properties and methods. (This is sometimes described as *passing a message* to an object.) Methods always require parentheses () after the method name.
   1. Put an object into a variable  
      $gem='Diamond'
   2. Access a property  
      $gem.Length
      1. How many characters are in $gem? Click or tap here to enter text.
   3. Call a method with no parameters  
      $gem.ToUpper()
      1. What was returned? Click or tap here to enter text.
   4. Call a method with a parameter  
      $gem.Substring(3)
      1. What was returned? Click or tap here to enter text.
   5. When calling a method with multiple parameters use a **,** (comma) to separate parameter values. Call a method with multiple parameters  
      $gem.Substring(3,2)
      1. What was returned? Click or tap here to enter text.
2. The **::** operator is used to access *static* properties and methods of a class. Static members are used *without* an instance of an object.
   1. Call a static method of the string type  
      [string]::IsNullOrEmpty($gem)
      1. What was returned? Click or tap here to enter text.
   2. Access a static property  
      [datetime]::now
      1. What was returned? Click or tap here to enter text.

# Task 2—Grouping and Subexpressions

## Steps

1. Parenthesis are used to group expressions, or force order of operations.
2. Group expression. This example forces a multiply operation to complete *before* a method call:  
   (1 \* 2.5).gettype()
   1. What is returned? Click or tap here to enter text.
3. Group the results of a cmdlet  
   (Get-Process).count
   1. What is returned? Click or tap here to enter text.
4. **$()** groups/interpolates statements. (May contain loops, statements separated by semicolon **;** )  
   "Hi $($n=read-host 'What is your name?';$n.toupper())"
   1. What is returned? Click or tap here to enter text.
5. **@()** is the same as **$()** except it *always* returns an array.  
   $a=$($n=read-host 'What is your name?';$n.toupper())  
   $b=@($n=read-host 'What is your name?';$n.toupper())
   1. What type is the value of **$a**? Click or tap here to enter text.
   2. What type is the value of **$b**? Click or tap here to enter text.

# Deliverable

Upload this document with completed answers to I-Learn Canvas.