## **Variable and Comparison Operators Workshop**

**Work through the below tasks to practice what has been covered in today’s workshop. Rather than focusing on getting the correct answer, it’s the process of getting to the answer that’s important; ask for help but do not copy others, it’s more important to think about these tasks then getting the answers right.**

**Task 1**

Assign a variable with each datatypes covered in the previous workshop

*The datatypes previous covered were “Boolean”, “String”, “Int”*

Notes:

$BoolanVar = $True

$StringVar = "This is a string"

$IntVar = 42

**Task 2**

List all variables currently loaded in to memory.

*The noun within the cmdlet is “Variable”*

Notes:

Get-Variable

**Task 3**

Multiple two Int variables together

*PowerShell can perform basic maths operators, such as “+”, “-“ and “\*”*

Notes:

$IntVar1 = 4

$IntVar2 = 10

$IntVar1 \* $IntVar2

**Task 4**

First declare two Int variables. Then divided the first variable by the second and assign the result to a variable named $VariableResult

*The easiest way to do this is within the Integrated Scripting Environment (ISE) using multiple lines of code*

Notes:

$IntVar3 = 10

$IntVar4 = 2

$VariableResult = $IntVar3 / $IntVar4

**Task 5**

Typecast a Variable as a “String” and assign it a value of 5

*Remember to normal brackets [] rather than curly brackets {} when typecasting a variable*

Notes:

[String]$TypecastVar = 5

**Task 6**

Using the greater than comparison operator produce code for which the result will be the Boolean value of “True”

*To use the greater than comparison operator use “-gt”*

Notes:

10 -gt 2

**Task 7**

Using the greater than comparison operator produce code for which the result will be the Boolean value of “False”

Notes:

10 -gt 20

**Task 8**

Using the greater than comparison operator produce code for which the result will not return no value or datatype.

Notes:

10, 20, 30 -gt 50

**Task 9**

Using the greater than comparison operator produce code for which the result will be the Int value of “32”

Notes:

10, 20, 32 -gt 31

**Task 10**

Using the less than comparison operator produce code for which the result will be the Boolean value of “True”

*To use the greater than comparison operator use “-lt”*

Notes:

10 -lt 20

**Task 11**

Using the less than comparison operator produce code for which the result will be the Boolean value of “False”

Notes:

20 -lt 1

**Task 12**

Using the less than comparison operator produce code for which the result will not return no value or datatype.

Notes:

10, 20, 30 -lt 1

**Task 13**

Using the less than comparison operator produce code for which the result will be the int value of “342”

Notes:

342, 500, 1000 -lt 400

**Task 14**

Using the like comparison operator produce code for which the result will be the Boolean value of “True”

*To use the greater than comparison operator use “-like”*

Notes:

"Hello" -like "\*ello"

**Task 15**

Using the like comparison operator produce code for which the result will be the Boolean value of “False”

Notes:

"Boo" -like "\*Chocolate"

**Task 16**

Using the like comparison operator produce code for which the result will not return no value or datatype.

Notes:

"Boo", "Chocolate", "Unicorn" -like "\*Potato"

**Task 17**

Using the like comparison operator produce code for which the result will be the String values of both “The unique gravity within Skyrim when you are hit with a troll’s club is not a bug, but rather an expected feature” and “Butterflies are unique and wonderful creatures”

Notes:

"The unique gravity within Skyrim when you are hit with a troll's club is not a bug, but rather an unexpected feature",

"Butterflies are unique and wonderful creatures" -like "\*unique\*"

**Task 18**

Using the equal to comparison operator produce code for which the result will be the Boolean value of “True”

*To use the equal to comparison operator use “-eq”*

Notes:

1 -eq 1

**Task 19**

Using the equal to comparison operator produce code for which the result will be the Boolean value of “False”

Notes:

1 -eq 2

**Task 20**

Using the equal to comparison operator produce code for which the result will not return no value or datatype.

Notes:

1,2,3 -eq 4

**Task 21**

Using the equal to comparison operator produce code for which the result will be the Int value of “456”

Notes:

216, 400, "Boo", 456 -eq 456