## **Control Structures, Functions and Standards: Activity 1**

### **What is an if/Else statement?**

An if statement is a logical control structure that only executes if a condition is true. For example:

$Day = $True

If ($Day –eq $True) {

Write-Host “Hello World”

}

Would write “Hello World” to the console.

An extension of this is the if/else statement, where the else code block is executed if the if condition provided is false. For example:

$Day = $False

If ($Day –eq $True) {

Write-Host “Hello World”

} else {

Write-Host “Goodnight World”

}

Would write “Goodnight World” to the console.

On the sheet overhead, there are four semi-complete examples of if an if else statements. See if you can complete these. Note that each of the comparison operators we covered in the previous workshop (-gt, -lt, -like, -eq) should be used.

1. Amend the below so that “You are the dancing queen” is written to the console.

$Age = 23

If ( 17) {

Write-Host "You are the dancing Queen!"

} else {

Write-Host "I can't relate to Abba songs anymore :("

}

1. Amend the below so that “Please buy more licenses” is written to the console.

$Licenses = 4000

if ( ) {

Write-Host "Please buy more licenses"

}

1. Amend the below so that “Quack!” is written to the console.

$Name = "Duck"

if ( ) {

Write-Host "Quack!"

}

1. Amend the below so that “It’s just the one swan actually” is written to the console.

$Swans = 1

Write-Host "No luck catching them swans then?"

if ( ) {

Write-Host "It's just the one swan actually"

} else {

Write-Host "I prefer Shaun of the dead"

}

### **What is a while statement?**

A while statement will execute a block of code once, check the given statement and if true run again. It’s vitally important to ensure your while statements can close, otherwise they’ll run forever.

For example:

While (1 –eq 1) {

Write-Host “Hello”

}

Will first check if 1 is equal to 1. As this condition is true it’ll then write “Hello” to the console. It’ll then check the statement again, 1 is equal to 1 so the condition is true and it’ll write “Hello” again. 1 will never not be equal to 1, so the while statement will be infinitely.

Whereas;

$Beer = 99

While ($Beer –gt 0) {

$CurrentBeer = $Beer

$NewBeer = $Beer – 1

Write-Host “$CurrentBeer bottles of beer on the wall, $CurrentBeer bottles of beer.”

Write-Host “Take one down and pass it around, $NewBeer bottles of beer on the wall”

$Beer = $Beer -1

}

Will continue until $Beer is not greater than 0. As the end of the while statement takes 1 away from this variable, the while statement will eventually close. You can test this on your machine!

On the sheet overhead, there are 3 semi-complete examples of while statements. See if you can complete these and identify ones which are infinitely loops.

1. Amend the below so that 5, 4, 3, 2, 1 are written to the console on separate lines.

$Counter = 5

While ( ) {

Write-Host $Counter

$Counter

}

1. Amend the below so that “Gobble gobble ram” is written to the console.

$Run = $True

While ( ) {

Write-Host "Gobble gobble ram"

}

1. Amend the below so that “All your files are belong to us” is written to the console.

$RansomwareFiles = 10

While ( ) {

Write-Host "No ransomware"

} else {

Write-Host "All your files are belong to us"

}

### **What is a foreach statement?**

A foreach statement will enumerate – which is just a fancy word for going through things one by one – through a given array.

It will start on the first item, store it in a temporary variable and then execute the given code block. Once this is complete it will then continue on to the second item, and continue the process until all items have been enumerated through.

For example:

$NarniaBooks = "The Lion, the Watch and the Wardrobe", "Prince Caspian: The Return to Narnia", "The Voyage of the Dawn Treader", "The Silver Chair", "The Horse and His Boy", "The Magician's Nephew", "The Last Battle"

foreach ($Book in $NarniaBooks) {

Write-Host "C.S. Lewis wrote $Book"

}

Would write:

C.S. Lewis wrote The Lion, the Watch and the Wardrobe

C.S. Lewis wrote Prince Caspian: The Return to Narnia

C.S. Lewis wrote The Voyage of the Dawn Treader

C.S. Lewis wrote The Silver Chair

C.S. Lewis wrote The Horse and His Boy

C.S. Lewis wrote The Magician's Nephew

C.S. Lewis wrote The Last Battle

As the $Book variable equates to whatever item in the array is currently being enumerated.

Another example would be:

$Stuff = "Bob", "Steve", "Thing"

foreach ($Thingy in $Stuff) {

Write-Host "This is an iteration"

}

Which would return the following to the console:

This is an iteration

This is an iteration

This is an iteration

As the foreach loop executes 3 times; once for each item in the array.

On the sheet overhead, there are 3 semi-complete examples. See if you can complete them.

1. Amend the below so that “I like twix, I like Snickers, I like Mars Bar and I like Bounty” are all returned to the console on separate lines.

$Chocolates = "Twix", "Snickers", "Mars Bar", "Bounty"

foreach ($ThingyMcThing in $Chocolates) {

Write-Host "I like "

}

1. Amend the below to return all services and their current running state. E.G. Service1, Running

$Services = Get-Service

foreach ($OtherThing in $Services) {

$ServiceName =

$ServiceStatus =

Write-Host "$ServiceName, $ServiceStatus"

}

1. The below is an example of a nested control structure; the if statement is inside the foreach loop. Amend the below so that if the current enumerated item is equal to “Matt Smith” the message written to console includes “apples rubbish, I hate apples”

$Doctors = "William Hartnell", "Patrick Troughton", "Jon Pertwee", "Tom Baker", "Peter Davison", "Colin Baker", "Sylvester McCoy", "Paul MCGann", "Christopher Eccelston", "David Tenant", "Matt Smith", "Peter Capaldi", "Jodie Whitttaker"

foreach ($Doc in $Doctors) {

$MSG = " says "

if ( ) {

$MSG = $MSG + "apples rubbish, I hate apples."

} else {

$MSG = $MSG + "hello!"

}

Write-Host $MSG

}