

Assignment 3

Total = 25pts

```
In [ ]: # create an array to hold your name and student number,
# use array destructuring to extract the data into variables
# and then print out the values
$student = "Sam", "w0471137"
Write-Host "Name - " $student[0] "ID - "$student[1]
```

Name - Sam ID - w0471137

Name: Jane Doe, StudentID: w123456

Part A - Small Bits (15pts)

For each item below, determine the appropriate PowerShell code to generate the desired output.

```
In [ ]: # create an array of all five assignments in this course
# (e.g. "Assignment 1", "Assignment 2", etc.)
# then loop through the array and print the names of all the
# assignments, except the current one.

# 3pts
$Assigns = "Assign1", "Assign2", "Assign3", "Assign4", "Assign5"
$Assigns = $Assigns[0,1+3..($Assigns.Length-1)]
for ($i=0; $i -le 5; $i++) {
    Write-Host (IEX "`$Assigns[$i]")
}
```

Assign1
Assign2
Assign4
Assign5

```
In [ ]: # create an array of your courses this semester.
# make each element another array (i.e. a multidimensional array)
# that includes both the course code and the course name
# e.g. 'PROG1700' and 'Logic and Programming I'
# then print out the first course name in the list and the last course code

# 3pts
$Program = "PROG1700"
$program2 = "Logic and Program"
$Windows = "OSYS1200"
$windows2 = "Intro to windows"
```

```

$Web = "WEBD1000"
$web2 = "Website develop"
$Data = "DBAS1007"
$data2 = "Data Fundament"
$program2
$Data

```

Logic and Program
DBAS1007

```

In [ ]: # create a hash table that contains a list of common
        # texting slang and their matching words/phrases
        # e.g. lol = laugh out loud
        # then use the new hash table to decode the following text

        # 3 pts

        # use this text
        $gibberish = "idk imho fwiw ur skillz r l33t l8r"

        # add your code here
        $Words = @{
            'idk' = 'I don't know'
            'imho' = 'in my humble opinion'
            'fwiw' = "for what it's worth"
            'ur' = 'your'
            'r' = 'are'
            'l33t' = 'elite'
            'l8r' = 'later'
        }

        $wordsArray = $gibberish -split '\s+'
        $decodedWords = @()

        foreach ($word in $wordsArray) {
            if ($Words.ContainsKey($word)) {
                $decodedWords += $Words[$word]
            } else {
                $decodedWords += $word
            }
        }

        $decodedText = $decodedWords -join ' '
        Write-Host "Decoded: $decodedText"

```

Decoded: I don't know in my humble opinion for what it's worth your skillz are elite later

```

In [ ]: # create a program that takes a lowercase word
        # then makes every odd letter upper case and every even letter lower case
        # print out the resulting word
        # e.g. elite => ElItE

        # 3 pts

        # use this word

```

```
$word = "hacker"

# add your code here
$result = ""
for ($i = 0; $i -lt $word.Length; $i++) {
    if ($i % 2 -eq 0) {
        $result += [System.Char]::ToUpper($word[$i])
    } else {
        $result += [System.Char]::ToLower($word[$i])
    }
}
$result
```

HaCkEr

```
In [ ]: # create a program that produces an acronym from a series of words
# e.g. the output should be 'NSCC'

# 3 pts

# use this text
$text = "Nova Scotia Community College"

# add your code here
$words = $text.Split(" ")
$acronym = ""
foreach ($word in $words){
    $acronym += $word[0]
}
$acronym.toupper()
```

NSCC

Part B - Big Bytes! (10pts)

The following are more challenging questions. Be patient when tackling these!

```
In [ ]: # create a program that converts a date in the format "mm/dd/yy"
# to a date in the format "dd-mmm-yyyy" using the provided string.
# e.g. 10/25/19 => 25-Oct-2019
# Note: you can assume all dates are >= 2000

# 3 pts

$date = "10/25/19" # sample date to convert

# use the months from this string to create the date
$months = "Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec"

# put your code here
$parts = $date.Split('/')
$month = $parts[0]
```

```

$day = $parts[1]
$year = $parts[2]
$monthName = $months.Split(', ')[($month - 1)]
$convertedDate = "$day-$monthName-20$year"
$convertedDate

```

25-Oct-2019

```

In [ ]: # using the following hash table of assignment grades,
# programmatically determine the lowest grade in the list.
# (there are lots of different ways to do this!)
# once you have determined the lowest grade,
# create a two-element array containing the name and grade
# of that assignment and then display the contents of the array

# 4 pts

$assignments = @{
    'Assignment 1' = 45;
    'Assignment 2' = 65;
    'Assignment 3' = 12;
    'Assignment 4' = 78;
    'Assignment 5' = 52
}

# put your code here
$assignments = @{
    'Assignment 1' = 45;
    'Assignment 2' = 65;
    'Assignment 3' = 12;
    'Assignment 4' = 78;
    'Assignment 5' = 52
}

foreach ($a in $assignments.GetEnumerator()) {

}

$assignments.GetEnumerator() | Sort-Object -Property Value

$Lowest = @('Assignment 3, '12')
Write-Host $Lowest

```

Name	Value
-----	-----
Assignment 3	12
Assignment 1	45
Assignment 5	52
Assignment 2	65
Assignment 4	78
Assignment 3, 12	

```

In [ ]: # create a hash table (a.k.a dictionary) that maps each letter of the alphabet to a
# e.g. a = x, b = q, c = p, ...
# then use that hash table to encrypt the following message
# the resulting encrypted message should be all capital letters and contain no
# punctuation or spaces.

```

```

# 3 pts

$message = 'This is the great and powerful Oz.'

# add your code here
$alphabetbutnot = @{
    'A' = 'J'
    'B' = 'V'
    'C' = 'E'
    'D' = 'R'
    'E' = 'D'
    'F' = 'U'
    'G' = 'Q'
    'H' = 'O'
    'I' = 'B'
    'J' = 'N'
    'K' = 'I'
    'L' = 'Y'
    'M' = 'C'
    'N' = 'W'
    'O' = 'H'
    'P' = 'G'
    'Q' = 'P'
    'R' = 'T'
    'S' = 'F'
    'T' = 'K'
    'U' = 'S'
    'V' = 'M'
    'W' = 'Z'
    'X' = 'L'
    'Y' = 'X'
    'Z' = 'A'
}

$encryptedMessage = ""

foreach ($char in $message.ToCharArray()) {
    $char = $char.ToString().ToLower()
    if ($alphabetbutnot.ContainsKey($char)) {
        $encryptedMessage += $alphabetbutnot[$char].ToString().ToUpper()
    } elseif ($char -eq ' ') {
    } else {
        $encryptedMessage += $char
    }
}

$encryptedMessage

```

KOBFBFKODQTDJKJWRGHZDTUSYHA.

In []: