**Submission Requirements**

* Submit the following through Brightspace:
  + Week 7 - Tool Development 4 (**this document**) with all required screenshots.
    - Plus the link to your GitHub repository where your PowerShell script was pushed: **https://github.com/hyamout/ISS212-Yamout**

**Screenshot of Full Script**

**A screen shot of a computer code

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**Screenshot of Script Output**

**A screenshot of a computer

Description automatically generated’**

**A screenshot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**Answer the following questions:**

1. **How does PowerShell handle large text-based log files, and what techniques improve efficiency when processing them?***(Hint: Consider how reading files into arrays vs. streaming them affects memory usage, and how filtering early in processing can optimize performance.)*

***PowerShell reads the file into memory using Get-Content, which loads the entire file into an array. While simple, this approach can become memory-intensive for very large files.***

***Streaming logs using Get-Content -ReadCount or Get-Content | ForEach-Object processes lines one at a time, reducing memory usage.***

***Filtering out invalid entries early (e.g., using regex) prevents unnecessary processing of irrelevant data and speeds up execution.***

1. **What are the advantages of using hash tables instead of arrays when working with structured log data in PowerShell?***(Hint: Think about how hash tables allow for quick lookups and counting occurrences of log levels, compared to iterating through an array repeatedly.)*

***Hash Tables: Allow quick lookups and direct counting of occurrences (e.g., incrementing log levels as keys).***

***Arrays: Require loops to search or count, which is slower when dealing with large datasets.***

***Hash tables are ideal for tasks like analyzing log levels because they simplify operations and improve performance.***

1. **Why is it important to validate and sanitize log file data before processing it in a script? What potential issues might arise if this step is skipped?***(Hint: Consider cases like missing data, incorrect formatting, or unexpected input that could break regex parsing or script logic.)*

**Validation ensures the log data has the correct format, preventing regex failures and script errors.**

**Skipping sanitization can lead to issues like:**

* **Processing invalid entries that distort results.**
* **Crashing the script due to unexpected input (e.g., missing fields or malformed lines).**
* **Wasting time and memory on unnecessary or bad data.**

**Think critically about efficiency, structure, and reliability when processing logs in PowerShell. The best scripts:**

* Use memory-efficient techniques (streaming, filtering early).
* Choose the right data structures (hash tables for counts, arrays for sequences).
* Validate input to prevent failures (handling missing or bad data properly).