Name: Saniya Galib Tamboli

Email: [saniyagtamboli2412@gmail.com](mailto:saniyagtamboli2412@gmail.com)

Batch: DWS\_B5\_25VID2550

User Id: 34749

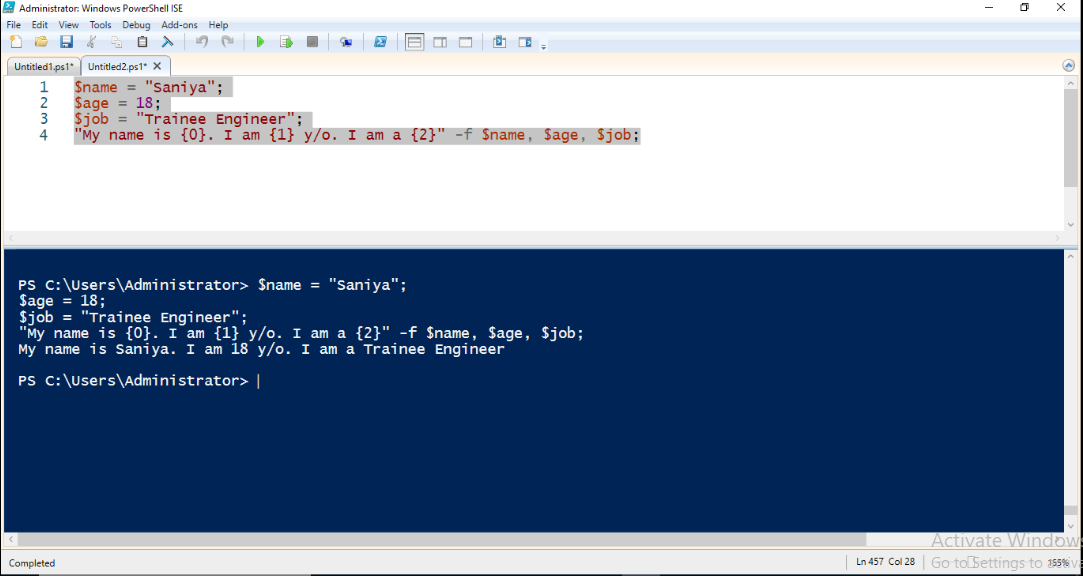
Date: 09/08/2025

**ASSIGNMENT**

PowerShell scripting constructs are fundamental elements that control the flow and logic of scripts. These constructs enable automation, decision-making, and repetition of tasks. Here's a breakdown of key scripting constructs in PowerShell:

* **Variables:** Used to store data values. They are defined using a $ prefix, e.g., $name = "John".
* **Arrays:** Ordered collections of items, accessed by index. Example: $fruits = @("apple", "banana", "orange").
* **Operators:** Symbols that perform operations on values:
* Arithmetic: +, -, \*, /, %
* Comparison: -eq, -ne, -gt, -lt, -ge, -le
* Logical: -and, -or, -not

1. **Printing Statements**

****Fig.1 Printing Statements

1. **Conditional Statements:**

if, elseif, else: Execute code blocks based on conditions.

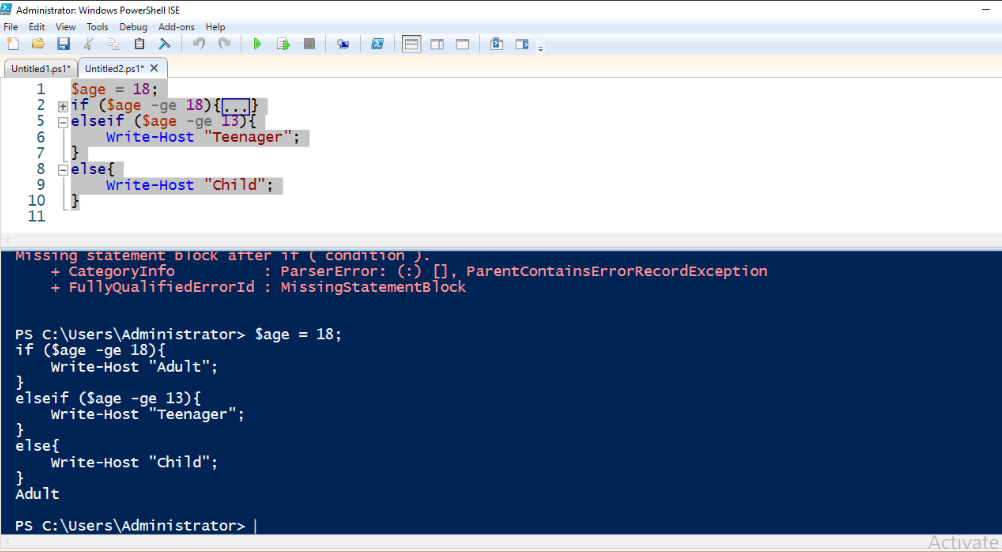
****

Fig.2 Conditional Statements

1. Switch: Efficiently handle multiple conditions.

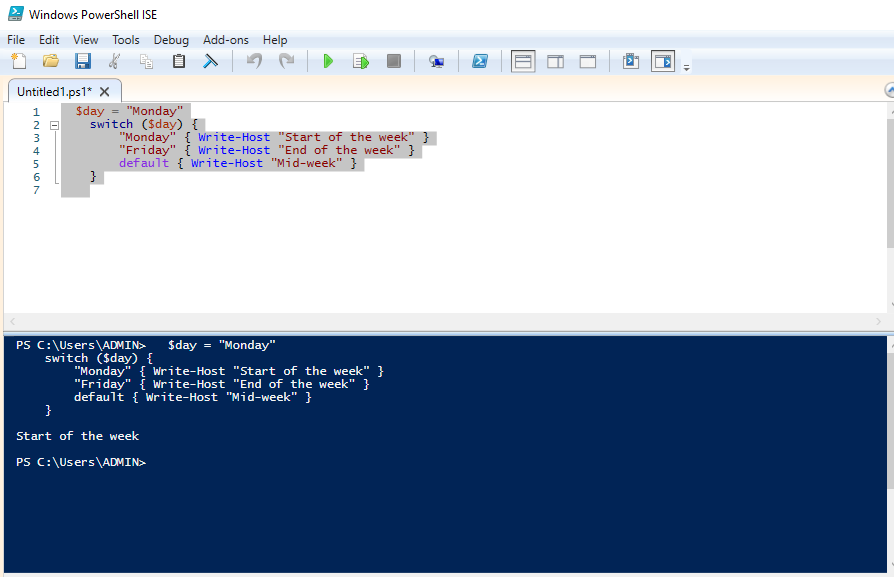
****

Fig.3. Switch Statement

**Modularization**

Modularization in PowerShell involves breaking down large scripts into smaller, reusable units called modules. These modules can contain functions, variables, and other PowerShell elements, promoting code organization, reusability, and maintainability.

**Types of Modules**

* **Script Modules (.psm1)**
* **Binary Modules (.dll)**
* **Manifest Modules (.psd1)**
* **Dynamic Modules**

**Creating a Module**

* **Create a .psm1 file**
* **Define functions**
* **(Optional) Create a module manifest**
* **Save the module**

**Using a Module**

* **Import the module.**
* **Use the module's functions**.

**Benefits of Modularization**

* **Code Reusability**
* **Organization**
* **Maintainability**
* **Collaboration**
* **Autoloading**