

```

# FlowerShopManager.ps1
# A demo project that uses 10 common text cmdlets
# Subject: Managing a Flower Shop's files and data

Write-Output "=== FLOWER SHOP MANAGER DEMO ==="

# -----
# 1. Set-Content -> Create an inventory file
# -----
Set-Content -Path "Inventory.txt" -Value @"
Rose, 25
Tulip, 40
Orchid, 12
Daisy, 30
"@
Write-Output "`nCreated Inventory.txt with flower stock."

# -----
# 2. Get-Content -> Read the inventory
# -----
Write-Output "`nReading inventory from file:"
Get-Content -Path "Inventory.txt"

# -----
# 3. Add-Content -> Add new flowers to inventory
# -----
Add-Content -Path "Inventory.txt" -Value "Sunflower, 15"
Write-Output "`nAdded Sunflower to inventory."
Write-Output "Updated Inventory:"
Get-Content -Path "Inventory.txt"

# -----
# 4. Out-File -> Save a daily sales log
# -----
"Sold 5 Roses" | Out-File -FilePath "SalesLog.txt"
"Sold 3 Tulips" | Add-Content -Path "SalesLog.txt"
Write-Output "`nSalesLog.txt created:"
Get-Content -Path "SalesLog.txt"

# -----
# 5. Select-String -> Search for a flower in inventory
# -----
Write-Output "`nSearching for 'Tulip' in Inventory.txt:"
Select-String -Path "Inventory.txt" -Pattern "Tulip"

# -----
# 6. Format-Table -> Display inventory as a table
# -----
Write-Output "`nInventory as a table:"
Get-Content -Path "Inventory.txt" |

```

```

   ForEach-Object {
        $parts = $_ -split ", "
        [PSCustomObject]@{ Flower=$parts[0]; Stock=$parts[1] }
    } | Format-Table Flower, Stock

# -----
# 7. Format-List -> Show details of a single flower
# -----
Write-Output "`nShowing details of one flower (Rose):"
Get-Content -Path "Inventory.txt" |
    ForEach-Object {
        $parts = $_ -split ", "
        [PSCustomObject]@{ Flower=$parts[0]; Stock=$parts[1] }
    } | Where-Object { $_.Flower -eq "Rose" } | Format-List *

# -----
# 8. ConvertTo-Json -> Export inventory to JSON
# -----
$inventory = Get-Content -Path "Inventory.txt" | ForEach-Object {
    $parts = $_ -split ", "
    [PSCustomObject]@{ Flower=$parts[0]; Stock=[int]$parts[1] }
}
$json = $inventory | ConvertTo-Json
Set-Content -Path "Inventory.json" -Value $json
Write-Output "`nExported inventory to Inventory.json"

# -----
# 9. ConvertFrom-Json -> Import inventory back into objects
# -----
$importedInventory = Get-Content -Path "Inventory.json" | ConvertFrom-Json
Write-Output "`nImported inventory from JSON:"
$importedInventory | Format-Table Flower, Stock

# -----
# 10. Out-String -> Create a plain-text report
# -----
$report = $importedInventory | Format-Table Flower, Stock | Out-String
Set-Content -Path "InventoryReport.txt" -Value $report
Write-Output "`nCreated InventoryReport.txt (plain text table):"
Get-Content -Path "InventoryReport.txt"

Write-Output "`n=== DEMO COMPLETE ==="

```