**Shell script hands-on:**

1. Write a shell script that prints “Shell Scripting is Fun!” on the screen.

#!/bin/bash #

echo "Shell Scripting is Fun!"

1. Here's the modified shell script with a variable holding the message:

message="Shell Scripting is Fun!"

echo $message

1. Write a shell script to output a specified directory’s size.

#!/bin/bash

# This script outputs the size of the specified directory

# Check if a directory is provided as an argument

if [ -z "$1" ]; then

echo "Please specify a directory."

exit 1

fi

# Get the size of the specified directory

directory\_size=$(du -sh "$1" | cut -f1)

# Output the directory size

echo "The size of the directory '$1' is: $directory\_size

==================================

Steps to run the script:

chmod +x directory\_size.sh

./directory\_size.sh

======================

4. Write a shell script to show hardware information for Linux systems.

#!/bin/bash

# This script shows hardware information for a Linux system

echo "==================== CPU Information ===================="

lscpu

echo ""

echo "==================== Memory Information =================="

free -h

echo ""

echo "==================== Disk Information ===================="

lsblk

echo ""

echo "==================== Filesystem Usage ===================="

df -h

echo ""

echo "==================== PCI Devices ========================"

lspci

echo ""

echo "==================== USB Devices ========================"

lsusb

1. #Write a shell script to check if a number input from standard input is odd or even

#!/bin/bash

# This script checks if a number is odd or even

# Prompt the user for input

echo "Please enter a number:"

read number

# Check if the number is an integer

if ! [[ "$number" =~ ^-?[0-9]+$ ]]; then

echo "Error: Please enter a valid number."

exit 1

fi

# Check if the number is even or odd

if (( number % 2 == 0 )); then

echo "The number $number is even."

else

echo "The number $number is odd."

Fi

1. Write a shell script to check if a number input from standard input is odd or even

#!/bin/bash

# This script checks if a number is odd or even

# Prompt the user for input

echo "Please enter a number:"

read number

# Check if the number is an integer

if ! [[ "$number" =~ ^-?[0-9]+$ ]]; then

echo "Error: Please enter a valid number."

exit 1

fi

# Check if the number is even or odd

if (( number % 2 == 0 )); then

echo "The number $number is even."

else

echo "The number $number is odd."

Fi

1. Write a shell script to test if a number being entered is a Fibonacci number or not.

#!/bin/bash

# This script checks if a number is a Fibonacci number

# Ask the user to enter a number

echo "Enter a number:"

read number

# Check if the entered value is a valid number

if ! [[ "$number" =~ ^[0-9]+$ ]]; then

echo "Please enter a valid number."

exit 1

fi

# Initialize the first two Fibonacci numbers

a=0

b=1

# Check if the number is in the Fibonacci sequence

while [ $b -lt $number ]; do

# Generate the next Fibonacci number

fib=$((a + b))

a=$b

b=$fib

done

# Check if we found the number in the sequence

if [ $b -eq $number ]; then

echo "$number is a Fibonacci number."

else

echo "$number is NOT a Fibonacci number."

Fi

1. Write a script to zip a file.

#!/bin/bash

# This script zips a file

# Ask the user to enter the file name to be zipped

echo "Enter the file name (with path if needed) to zip:"

read file\_to\_zip

# Check if the file exists

if [ ! -f "$file\_to\_zip" ]; then

echo "Error: The file does not exist."

exit 1

fi

# Create a zip file with the same name as the input file, but with a .zip extension

zip "${file\_to\_zip}.zip" "$file\_to\_zip"

# Check if the zipping was successful

if [ $? -eq 0 ]; then

echo "The file has been zipped successfully: ${file\_to\_zip}.zip"

else

echo "There was an error zipping the file."

Fi

1. Write a script to unzip a file

#!/bin/bash

# This script unzips a file

# Ask the user to enter the zip file name

echo "Enter the zip file name (with path if needed) to unzip:"

read zip\_file

# Check if the file exists and is a zip file

if [ ! -f "$zip\_file" ]; then

echo "Error: The file does not exist."

exit 1

fi

if [[ "$zip\_file" != \*.zip ]]; then

echo "Error: The file is not a zip file."

exit 1

fi

# Unzip the file

unzip "$zip\_file"

# Check if the unzip was successful

if [ $? -eq 0 ]; then

echo "The file has been unzipped successfully."

else

echo "There was an error unzipping the file."

fi