Assignment 1: Ensure the script checks if a specific tile (eg, myfile.bt) exists in the current directory. If easts, print "File exists", otherwise print "File not found"

Solution:

```
#!/bin/bash

# Specify the filename to check
filename="myfile.bt"

# Check if the file exists in the current directory
if [ -e "$filename" ]; then
   echo "File exists"
else
   echo "File not found"
fi
```

Assignment 2: Write a script that reads numbers from the user until they enter 0 The script should also print whether each number is odd or even

Solution:

```
#!/bin/bash
# This is a Bash script

# Set shell options
set -e  # Exit immediately if a command exits with a non-zero status
set -u  # Treat unset variables as an error when substituting
set -o pipefail  # Return non-zero status if any command in a pipeline fails

# Define variables or environment settings
MY_VAR="Hello, World!"

# Main script logic starts here
echo "$MY_VAR"
```

Assignment 3: Create a function that takes a filename as an argument and prints the number of lines in the file Call this function from your script with different filenames.
Solution :

```
#!/bin/bash

# Function to count lines in a file
count_lines() {
   local filename="$1"
   local num_lines=$(wc -l < "$filename")
   echo "Number of lines in $filename: $num_lines"
}

# Call the function with different filenames
count_lines "file1.txt"
count_lines "file2.txt"
count_lines "file3.txt"
```

Assignment 4: Witte a script that creates a directory named TestDir and inside it, creates ten files named File1 txt, File2 txt File10 bit. Each file should contain its filename as its content (eg. File1.txt contains "File1.txt)

```
#!/bin/bash

# Create the directory TestDir if it doesn't exist
mkdir -p TestDir

# Loop to create ten files
for ((i=1; i<=10; i++)); do
   filename="File${i}.txt"
   echo "$filename" > "TestDir/$filename"
   echo "Created $filename with content: $filename"
done

echo "Files created successfully in TestDir."
```

Assignment 5: Modify the script to handle errors, such as the directory already existing or lacking permissions to create files

Solution:

```
#!/bin/bash

# Function to create directory and files
```

```
create_files() {
   local dir_name="TestDir"
   local filename=""

   # Check if directory TestDir already exists
   if [ -d "$dir_name" ]; then
      echo "Error: Directory $dir_name already exists."
      return 1
   fi

   # Create the directory TestDir
   mkdir -p "$dir_name" || { echo "Error: Failed to create directory $dir_name."; return 1; }

   # Loop to create ten files
   for ((i=1; i<=10; i++)); do
      filename="File${i}.txt"
      echo "$filename" > "$dir_name/$filename" || { echo "Error: Failed to create file $filename."; return
1; }
      echo "Created $filename with content: $filename"
   done

   echo "Files created successfully in $dir_name."
}

# Call the function to create directory and files
create_files
```

Assignment 6: Given a sample log file, write a script using grep to extract all lines

containing "ERROR" Use awk to print the date, time, and error message of each extracted line

Data Processing with sed
Solution:

```
#!/bin/bash

# Use grep to extract lines containing "ERROR" from the log file
grep "ERROR" sample.log |
# Use awk to print date, time, and error message
awk '{print $1, $2, $3, $0}' |
# Use sed to format output (optional)
sed 's/\([0-9]\{4\}-[0-9]\{2\}-[0-9]\{2\}\)T\([0-9]\{2\}:[0-9]\{2\}:[0-9]\{2\}\)/Date: \1, Time: \2/g'
```

```
# Explanation:
# - grep "ERROR" sample.log: Extracts lines containing "ERROR" from sample.log
# - awk '{print $1, $2, $3, $0}': Prints date, time, and entire line
# - sed 's/\([0-9]\{4\}-[0-9]\{2\}-[0-9]\{2\}\)T\([0-9]\{2\}:[0-9]\{2\}:[0-9]\{2\}\)/Date: \1, Time: \2/g':
#   Formats the output to display "Date: <date>, Time: <time>" for each line
```

Assignment 7: Create a script that takes a text file and replaces all occurences of "old ted" with "new text". Use sed to perform this operation and output the result to a new file

Submission Guidelines:

1. Ensure that each answer is clear, concise, and reflects an understanding of the core concepts

2. Diagrams can be hand-drawn and scanned or created using any digital drawing tool Provide references for any exdermal sources used

Submit your work in a single PDF document by end of Module.
Solution:

```
#!/bin/bash

# Check if the user provided a filename as an argument
if [ $# -eq 0 ]; then
   echo "Error: Please provide a filename as an argument."
   exit 1
fi

# Check if the input file exists
if [ ! -f "$1" ]; then
   echo "Error: File '$1' not found."
   exit 1
fi

# Set input and output filenames
input_file="$1"
output_file="output.txt"

# Use sed to replace "old text" with "new text" and output to new file
sed 's/old text/new text/g' "$input_file" > "$output_file"

# Print success message
echo "Replacement completed. Output saved to $output_file"
```