**Assignment 1:**

Initialize a new Git repository in a directory of your choice. Add a simple text file to the repository and make the first commit.

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir (master)

**$ mkdir myproject**

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir (master)

**$ cd myproject**

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git init**

Initialized empty Git repository in C:/Program Files/Git/TestDir/myproject/.git/

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ touch index.html**

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ ls**

index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ vim index.html**

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git status**

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.html

nothing added to commit but untracked files present (use "git add" to track)

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git add index.html**

warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git status**

On branch master

No commits yet

Changes to be committed:

  (use "git rm --cached <file>..." to unstage)

        new file:   index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ touch hello.css**

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ vim hello.css**

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git add --all**

bash: gid: command not found

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git status**

On branch master

No commits yet

Changes to be committed:

  (use "git rm --cached <file>..." to unstage)

        new file:   index.html

Untracked files:

  (use "git add <file>..." to include in what will be committed)

        hello.css

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git add hello.css**

warning: in the working copy of 'hello.css', LF will be replaced by CRLF the next time Git touches it

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git add --all**

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git status**

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: hello.css

new file: index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git commit -m "First commit"**

[master (root-commit) 39d79a2] First commit

2 files changed, 20 insertions(+)

create mode 100644 hello.css

create mode 100644 index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ cat index.html**

```
<!DOCTYPE html>
<html>
<head>
<title>Sai Chandana</title>
</head>
<body>
    <h1> Hello </h1>
    <p> This is the first file in git Repo.</p>
</body>
</html>
```

**Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)**

**$ cat hello.css**

```
body {
    background-color: Lavender;
}
h1
{
```

```
    color: green;
    margin-left: 20px;
}
```

==========

**ASSIGNMENT 2:**

Branch Creation and Switching

Create a new branch named 'feature' and switch to it. Make changes in the 'feature' branch and commit them.

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git branch**

  feature

* master

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git checkout feature**

Switched to branch 'feature'

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (feature)

**$ ls**

hello.css  index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (feature)

**$ vim index.html**

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (feature)

**$ git add --all**

warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (feature)

**$ git status**

On branch feature

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: index.html

**$ cat index.html**

```
<!DOCTYPE html>
<html>
<head>
<title>Sai Chandana</title>
</head>

<body>
<h1> Hello </h1>
    <p> This is the first file in git Repe.</p>
    <p> I will write </p>
</body>
</html>
```

==========

**ASSIGNMENT 3:**

**Feature Branches and Hotfixes**

**Create a 'hotfix' branch to fix an issue in the main code. Merge the 'hotfix' branch into 'main' ensuring that the issue is resolved.**

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (feature)

**$ git branch hotfixer**

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (feature)

**$ git checout hotfixer**

git: 'checout' is not a git command. See 'git --help'.

The most similar command is

checkout

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (feature)

**$ git checkout hotfixer**

Switched to branch 'hotfixer'

M	index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (hotfixer)

**$ ls**

hello.css  index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (hotfixer)

**$ vim index.html**

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (hotfixer)

**$ git add --all**

warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (hotfixer)

**$ git status**

On branch hotfixer

Changes to be committed:

  (use "git restore --staged <file>..." to unstage)

	modified:   index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (hotfixer)

**$ git commit -m "Changes to fix from Hotfixer branch"**

[hotfixer 13fb4ec] Changes to fix from Hotfixer branch

1 file changed, 4 insertions(+), 2 deletions(-)

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (hotfixer)

**$ git checkout master**

Switched to branch 'master'

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ git merge hotfixer**

Updating 39d79a2..13fb4ec

Fast-forward

index.html | 6 ++++--

1 file changed, 4 insertions(+), 2 deletions(-)

Administrator@DESKTOP-TIC5DM4 MINGW64 /TestDir/myproject (master)

**$ cat index.html**

```
<!DOCTYPE html>
<html>
<head>
<title>Sai Chandana</title>
</head>

<body>
<h1> Hello </h1>
    <p> This is the first file in git Repe.</p>
    <p> I will write </p>
    <p> Assessment </p>
</body>
</html>
```

==========

## Day-2:

**Assignment 1: Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print "File exists", otherwise print "File not found".**

Administrator@DESKTOP-TIC5DM4 MINGW64 /desktop (master)

**$ touch checkfile.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 /desktop (master)

**$ vim checkfile.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 /desktop (master)

**$ touch myfile.txt**

Administrator@DESKTOP-TIC5DM4 MINGW64 /desktop (master)

**$ echo "Hello world!" > myfile.txt**

Administrator@DESKTOP-TIC5DM4 MINGW64 /desktop (master)

**$ chmod +x checkfile.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 /desktop (master)

**$ ./checkfile.sh**

File exists

Administrator@DESKTOP-TIC5DM4 MINGW64 /desktop (master)

**$ cat checkfile.sh**

```
#!/bin/bash
# Set the file name to check
FILENAME="myfile.txt"
#check if the file exists
if [ -f "$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.**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ touch Assignment1.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ vim Assignment1.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ vim Assignment1.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ ./Assignment1.sh**

Enter a number:

8

8 is even

Enter a number:

6

6 is even

Enter a number:

43

43 is odd

Enter a number:

44

44 is even

Enter a number:

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ .cat Assignment1.sh**

```
while true; do
  echo "Enter a number:"
  read number
  if [ $number -eq 0 ]; then
```

```
        echo "Existing"
        break
    fi
    if(($number %2 == 0)); then
        echo "$number is even"
    else
        echo "$number is odd"
    fi
```

==========

**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.**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ touch a2.txt**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ echo "Hello" >>a2.txt**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ echo "This is a test file." >>a2.txt**

Administrator@DESKTOP-TIC5DM4 MINGW64 / Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ vim Assignment3.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ chmod +x Assignment3.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ ./assignment3.sh**

number of lines are :

2 a2.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ cat a2.txt**

Hello

This is a test file.

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ cat Assignment3.sh**

```
filename=a2.txt
function1()
{
    echo "number of lines are : "
    wc -l $1
}
function1 $filename
```

==========

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

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ touch Assignment4.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ vim Assignment4.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ chmod +x Assignment4.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ ./Assignment4.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ ls TestDir**

file1.txt file10.txt file2.txt file3.txt file4.txt file5.txt file6.txt file7.txt file8.txt file9.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ cat Assignment4.sh**

```
#!/bin/bash
# create TestDir if it doesn't exist
mkdir -p "TestDir"
# change directory to TestDir
cd "TestDir"

# create 10 files with their names as content
for ((i=1; i<=10; i++));
do
    echo "file$i.txt" >file$i.txt
done
```

==========

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

**Add a debugging mode that prints additional information when enabled.**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ touch Assignment5.txt**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ vim Assignment5.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ cat Assignment5.sh**

```
#!/bin/bash
 if [ "$DEBUG"="true" ]; then
   set -x
fi
 if [ -d "TestDir" ]; then
      handleErrors "Directory already exists"
fi
 mkdir -p TestDir
cd TestDir
 for ((i=1; i<=10; i++)); do
    echo "File$i.txt" > "File$i.txt"
    if [ "$DEBUG"="true" ]; then
       set +x
fi
done
```

==========

**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**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ touch Assignment6.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ vim Assignment6.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ touch Sample.log**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ vim Sample.log**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ cat Sample.log**

2024-05-10 10:30:05 INFO: Application started

2024-05-10 10:30:10 ERROR: Database connection failed

2024-05-10 10:30:15 DEBUG: Processing request

2024-05-10 10:30:20 ERROR: Invalid input received

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ cat Assignment6.sh**

```
logFile="sample.log"
grep "ERROR" "$logFile" | awk '{print $1, $2, substr($0, index($0,$3))}' | sed 's/^[^ ] * //'
```

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ ./ Assignment6.sh**

2024-05-10 10:30:10 ERROR: Database connection failed

2024-05-10 10:30:20 ERROR: Invalid input received

==========

**Assignment 7: Create a script that takes a text file and replaces all occurrences of "old_text" with "new_text". Use sed to perform this operation and output the result to a new file.**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ touch Assignment7.txt**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ vim Assignment7.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ touch input.sh**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ vim input.txt**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ cat input.txt**

this is the text in input file

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ ./Assignment7.sh**

Usage: ./Assignment7.sh <given_file> <old_text> <new_text>

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ ./Assignment7.sh input.txt old_text new_text**

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

**$ cat input_modified.txt**

this is the text in input file