

**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 / (master)

```
$ git --version
```

```
git version 2.44.0.windows.1
```

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

```
$ mkdir demoproject1
```

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

```
$ cd demoproject1
```

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)

```
$ git init
```

```
Initialized empty Git repository in C:/Program Files/Git/demoproject1/.git/
```

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)

```
$ ls
```

```
index.html
```

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)

```
$ vim index.html
```

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)

```
$ cat index.html
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
<title>Page Title</title>
</head>
<body>
  <h1>My First Heading</h1>
</body>
</html>
```

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)

```
$ git init
```

Reinitialized existing Git repository in C:/Program Files/Git/demoproject1 /.git/

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (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 /demoproject1 (master)

```
$ git add -all. error: did you mean '--all` (with
two dashes)?
```

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)

```
$ 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 /demoproject1 (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 /demoproject1 (master)

\$ git config user.email "sivaveduruvada00@gmail.com"

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)

\$ git config user.name "siva"

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)

\$ git commit -m "initial commit"

[master (root-commit) b242225] initial commit

1 file changed, 9 insertions (+)

create mode 100644 index.html

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 /demoproject1 (master)
```

```
$ git branch feature
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)
```

```
$ git branch
```

```
feature
```

```
* master
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)
```

```
$ git checkout feature
```

```
Switched to branch 'feature'
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (feature)
```

```
$ ls
```

```
index.html
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (feature)
```

```
$ vim index.html
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (feature)
```

```
$ vim index.html
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (feature)
```

```
$ cat index.html
```

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <title>Page Title</title>
```

```
    </head>
```

```
    <body>
```

```
        <h1>My First Heading</h1>
```

```
        <h2>My second heading </h2>
```

```
</html>
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (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 /demoproject1 (feature)
```

```
$ git status
```

```
On branch feature
```

```
Changes to be committed:
```

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

```
modified:   index.html
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (feature)
```

```
$ git commit -m "feature branch" [feature  
bb5340f] feature branch  
1 file changed, 1 insertion(+)
```

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 /demoproject1 (feature)  
$ git checkout master  
Switched to branch 'master'
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)  
$ git branch hotfixer
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)  
$ git checkout hotfixer  
Switched to branch 'hotfixer'
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (hotfixer)  
$ ls  
index.html
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (hotfixer)  
$ vim index.html
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (hotfixer)  
$ cat index.html  
<!DOCTYPE html>  
<html>
```

```
<head>
  <title>Page Title</title>
</head>
<body>
  <h1>My First Heading</h1>
  <h2>second line</h2>
</body>
</html>
```

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (hotfixer)

```
$ git add --all
```

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (hotfixer)

```
$ git status
```

On branch hotfixer

Changes to be committed:

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

modified: index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (hotfixer)

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

[hotfixer 5ad629e] Changes to fix from Hotfixer branch

1 file changed, 1 insertion(+)

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (hotfixer)

```
$ git checkout master
```

Switched to branch 'master'

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 (master)

\$ git merge hotfixer

Updating b242225..5ad629e

Fast-forward

index.html | 1 +

1 file changed, 1 insertion (+)

Shell scripting:

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 / (master)

\$ touch myfile.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

\$ vim myfile.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

\$ touch sample.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

\$ touch sample.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

\$ vim sample.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

\$ vim myfile.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

\$./myfile.sh

file exists

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)


```
$ cat myfile.sh #!/bin/bash
```

```
if [ -f "sample.txt" ]
```

```
then
```

```
    echo "file exists"
```

```
else
```

```
    echo "file not found"
```

```
fi
```

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

```
$
```

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 myfile2.sh
```

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

```
$ vim myfile2.sh
```

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

```
$ ./myfile2.sh
```

```
./myfile2.sh: line 7: syntax error near unexpected token `fi'
```

```
./myfile2.sh: line 7: ` fi'
```

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

```
$ vim myfile2.sh
```

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

```
$ ./myfile2.sh
```

```
./myfile2.sh: line 8: syntax error near unexpected token `fi'
```

```
./myfile2.sh: line 8: ` fi'
```

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

```
$ vim myfile2.sh
```

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

```
$ ./myfile2.sh
```

```
./myfile2.sh: line 8: syntax error near unexpected token `fi'
```

```
./myfile2.sh: line 8: ` fi'
```

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

```
$ vim myfile2.sh
```

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

```
$ ./myfile2.sh enter
```

```
a number :
```

```
2      2 is even
```

```
enter a number :
```

```
3      3 is odd
```

```
enter a number :
```

```
0 exist
```

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

```
$ cat myfile2.sh #!/bin/bash
```

```
while true; do      echo
```

```
"enter a number : "  read
```

```
number      if [ $number -eq
```

```
0 ]; then      echo
```

```
"exist"      break
```

```
fi
```

```
        if [ $((number %2)) -eq 0 ]; then
echo "$number is even "
        else
            echo "$number is odd"
        fi
done
```

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

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 file1.txt
Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)
$ vim file1.txt
Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)
$ touch file2.txt
Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)
$ vim file12.txt
Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)
$ touch myfile3.sh
Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)
$ vim myfile3.sh
Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)
$ ./myfile3.sh
Number of lines in file1.txt : 1
```

Number of lines in file2.txt : 1

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

\$ cat file1.txt first

line

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)

\$ cat file2.txt

second line

Assignment 4: Write a script that creates a directory named Test Dir 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 /demoproject1

\$ touch fourth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1

\$ vim fourth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1

\$ cat fourth.sh

mkdir TestDir cd

TestDir

for((i=1;i<=10;i++))

do name="File\${i}"

echo "\$name">"\$name" done

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1

\$./fourth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1

\$ cd TestDir

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1 /TestDir

\$ ls

File1 File10 File2 File3 File4 File5 File6 File7 File8 File9Administrator@DESKTOP-TIC5DM4
MINGW64 /demoproject1 /TestDir

\$ cat file1

File1

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 /demoproject1

\$ touch fifth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1

\$ vim fifth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1

\$./fifth.sh

Error: Directory Already Exists

Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1

\$ cat fifth.sh if ["\$DEBUG" =

"true"]; then

set-x fi

errorHandler(){

echo "Error:\$1"

exit 1

}

if [-d "TestDir"]; then errorHandler

"Directory Already Exists"

fi

mkdir-p TestDir || errorHandler "Failed to create Directory" cs

TestDir || errorHandler "Failed to change Directory" for((

```
i=1;i<=10;i++ )); do          echo
"File$i.txt">"File$1.txt" | errorHandler "Failed to create
'File$i.txt'"
done
if [ "$DEBUG"="true" ]; then
    set + x
fi
```

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 /demoproject1
$ touch sixth.sh
Administrator@DESKTOP-TIC5DM4 MINGW64 / demoproject1
vim sixth.sh
Administrator@DESKTOP-TIC5DM4 MINGW64 / demoproject1
$ touch Sample.log
Administrator@DESKTOP-TIC5DM4 MINGW64 / demoproject1
$ vim Sample.log
Administrator@DESKTOP-TIC5DM4 MINGW64 / demoproject1
$ 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 / demoproject1
$ cat sixth.sh
```

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

```
Administrator@DESKTOP-TIC5DM4 MINGW64 / demoproject1
```

```
$ ./sixth.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 /demoproject1
```

```
$ touch seventh.sh
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 / demoproject1
```

```
$ vim seventh.sh
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 / demoproject1
```

```
$ touch input.txt
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1
```

```
$ vim input.txt
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /demoproject1
```

```
$ cat input.txt this is the text in input file
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 / demoproject1
```

```
$ cat seventh.sh if
```

```
[ "$#" -ne 3 ]; then
```

```
echo "Usage: $0
```

```
<givenFile>
```

```
<oldText>
```

```
<newText>"
```

```
exit 1 fi
```

```
givenFile="$1"
```

```
oldText="$2"
```

```
newText="$3"
```

```
finalFile="${givenFile%.txt}_modified.txt" sed
```

```
"s/$oldText/$newText/g" "$givenFile" > "$finalFile"
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 / demoproject1
```

```
$ ./seventh.sh input.txt oldText newText
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 / demoproject1
```

```
$ cat input_modified.txt
```

```
this is the text in input file
```