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.

Solution:

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /
$ cd myproject

Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ git init
bash: ff: command not found

Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ git init
Reinitialized existing Git repository in C:/Program Files/Git/myproject/.git/

Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ touch index.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ vim index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ ls
README.md  blueStyle.css  bueStyle.css  index.html  index.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ git status
On branch hello-world-images
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.txt

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

Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ git add index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ git status
On branch hello-world-images
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.txt
Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ git add all
fatal: pathspec 'all' did not match any files

Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ git add --all
```

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

Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ git status
On branch hello-world-images
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   anantha.css
        new file:   index.txt


Administrator@DESKTOP-TIC5DM4 MINGW64 /myproject (hello-world-images)
$ git commit -m "First commit"
[hello-world-images eebeab9] First commit
 2 files changed, 9 insertions(+)
 create mode 100644 anantha.css
 create mode 100644 index.txt
```

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.

Solution:

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

Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)
$ cd mysecoundproject

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (master)
$ touch index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (master)
$ vim index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (master)
$ git add index.html
warning: in the working copy of 'mysecoundproject/index.html', LF will be
replaced by CRLF the next time Git touches it

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (master)
$ git init
Initialized empty Git repository in C:/Program
Files/Git/mysecoundproject/.git/
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (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 /mysecoundproject (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 /mysecoundproject (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 /mysecoundproject (master)
$ git commit -m "secound assignment commit"
[master (root-commit) 185bb75] secound assignment commit
 1 file changed, 15 insertions(+)
 create mode 100644 index.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

Solution:

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (master)
$ git branch newbranch

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

```
* master
  newbranch

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (master)
$ git checkout newbranch
Switched to branch 'newbranch'

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ ls
index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ vim index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ 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 /mysecoundproject (newbranch)
$ git status
On branch newbranch
nothing to commit, working tree clean

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ vim index.html

[1]+  Stopped                 vim index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ vim index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ git add index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ git status
'On branch newbranch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .index.html.swp

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

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ git add index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ git status
On branch newbranch
```

```
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .index.html.swp

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

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ vim index.html

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ 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 /mysecoundproject (newbranch)
$ git status
On branch newbranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .index.html.swp


Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ git commit -m "cnages from new branch"
[newbranch a73b8e4] cnages from new branch
 1 file changed, 3 insertions(+), 1 deletion(-)

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (newbranch)
$ git checkout master
Switched to branch 'master'

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (master)
$ git marge newbranch
git: 'marge' is not a git command. See 'git --help'.

The most similar command is
        merge

Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (master)
$ cd index.html
bash: cd: index.html: Not a directory
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /mysecoundproject (master)
$ cat index.html
<!DOCTYPE html>
<html>
        <head>

                <title>HELLO WORLD</title>
                <link rel="stylesheet" herf="reddy.css">
        </head>
        <body>
                <h1>HELLO WORLD HOW ARE YOU</h1>
                <p>this is the first file in my new git repo.</p>
                <p>A new in the file!</p>
        </body>
</html>
```

Shell Scripting with Bash

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".

Solution:

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

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ ls -l
total 3
-rw-r--r-- 1 Administrator 197121 314 May  9 13:18 "'"
-rw-r--r-- 1 Administrator 197121   0 May  9 14:38  Filename.sh
-rw-r--r-- 1 Administrator 197121 314 May  9 13:20  ananth.sh
-rw-r--r-- 1 Administrator 197121   0 May  9 13:11  reddy.s
-rwxr-xr-x 1 Administrator 197121 378 May  9 13:59  reddy.sh*
-rw-r--r-- 1 Administrator 197121   0 May  9 12:43  testfile

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ touch fileExit.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ vim fileExit.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ ./fileExit.sh
fileExit.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ vim fileExit.sh
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ vim fileExit.sh
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ ./fileExit.sh

File is exist


Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ cat fileExit.sh
#! / bin / bash
#Using [expression] syntax and in place
# of fileExit.sh
if [ -f "fileExit.sh"];
then
#If file exist the it will be printed
echo "File is exist"
else
#it is not exist then it will be printed
echo "File is not exit"
fi


Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ git add fileExit.sh
warning: in the working copy of 'fileExit.sh', LF will be replaced by CRLF
the next time Git touches it

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Filename.sh
        new file:   ananth.sh
        new file:   fileExit.sh
        new file:   reddy.sh
```

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:

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ touch evenodd.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ vim evenodd.sh
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ ./evenodd.sh

----even or odd in the shell script---
Enter your number:10
RESULT: expr: non-integer argument
10 is even

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ cat evenodd.sh

#How to find even number is even or odd is shell script

clear
echo "----even or odd in the shell script---"
echo -n "Enter your number:"
read n
echo -n "RESULT: "
if [ `expr $n % 2` ==0]
then
        echo "$n is even"
else
        echo "$n is odd"
fi

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Filename.sh
        new file:   ananth.sh
        new file:   evenodd.sh
```

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:

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


Administrator@DESKTOP-TIC5DM4 MINGW64 / (master)
$ cd shellcript

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ git init
Initialized empty Git repository in C:/Program Files/Git/shellCript/.git/

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ ls -l
total 0


Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ touch R1.txt R2.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ vim R1.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ vim R2.txt
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ touch function.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ vim function.sh


Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ cat R1.txt
```

```
this is the first line
this is the secoud line
this is the third line


Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ cat R2.txt
i am learning linux this is the first day
this is secound day
this is third day
this is fourth day


Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ ./function.sh
The Number of Lines Are:
4 R1.txt
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellcript (master)
$ cat function.sh
#! / bin / bash
filename="R1.txt"

func1(){
        echo "The Number of Lines Are:"
        Wc -l "$1"
}

func1 "$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").

Solution:

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

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ touch assignmnt4.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ vim assignmnt4.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ chmod +x assignmnt4.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ ls TestDir
File1.txt File2.txt File3.txt File4.txt File5.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript (master)
$ cat assignmnt4.sh
#/bin/bash
#create TestDir if it does not exit
mkdir -p TestDir

cd TestDir

for ((i=1; i<=5; 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.

Solution:

```
 #! / bin / bash
if [ "$DEBUG" = "true"]; then
set -x
fi#Function Creation
handleErrors () {
echo "Error: $1"
exit 1
}
if [ -d "TestDir”]; then
handleErrors "Directory Already Exists"
fi
mkdir -p TestDir || handleErrors "Failed to create Directory"
cd TestDir || handleErrors "Failed to change Directory."
for ((i=1; i<=10; i++)); do
echo "File$i.txt" > "File$i.txt" || handleErrors "Failed to create file 'File$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

Solution:

```
# Define the filename

logfile="sample.log"

# Use grep to extract lines containing "ERROR", then use awk to print date, time, and error message


grep "ERROR" "$logfile" | awk '{print $1, $2, substr ($0, index ($0, $3))}' |

# Use sed for additional data processing if needed
sed 's/<pattern>/<replacement>/g
```

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.

Solution:

```
#! / bin/ bash
# Check if correct number of arguments provided
if [ "$#" -ne 3]; then
echo "Usage: $0 <input_file> <old_text> <new_text>"
exit 1
fi
input_file="$1"
old_text="$2"
new_text="$3"
# Check if input file exists
if [! -f "$input_file"]; thenecho "Error: Input file '$input_file' does not exist."
exit 1
fi
# Use sed to replace old_text with new_text and output to a new file
sed "s/$old_text/$new_text/g" "$input_file" > "${input_file} updated"
echo "Replacement complete. Updated file: ${input_file} updated"
```