

Objectives

Familiar with Git commands like git init, git status, git add, git commit, git push, and git pull.

In this hands-on lab, you will learn how to

- Setup your machine with Git Configuration
- Integrate notepad++.exe to Git and make it a default editor
- Add a file to source code repository

Prerequisites

- Install Git Bash client in your machine

Step 1: Setup your machine with Git Configuration

To create a new repository, signup with GitLab and register your credentials

Login to GitLab and create a “GitDemo” project

1. To check if Git client is installed properly: Open Git bash shell and execute

```
mkanm@Kanmani MINGW64 ~  
$ git version  
git version 2.50.0.windows.1
```

If output shows Git with its version information that indicates, that Git Client installs properly.

2. To configure user level configuration of user ID and email ID execute

```
mkanm@Kanmani MINGW64 ~  
$ git config --global user.name "KANMANI MURUGHAIYAN"  
  
mkanm@Kanmani MINGW64 ~  
$ git config --global user.email "kanmanicse2022@gmail.com"
```

3. To check if the configuration is properly set, execute the following command.

```
mkanm@Kanmani MINGW64 ~  
$ git config --list  
diff.astextplain.textconv=astextplain  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
http.sslbackend=schannel  
core.autocrlf=true  
core.fscache=true  
core.symlinks=false  
pull.rebase=false  
credential.helper=manager  
credential.https://dev.azure.com.usehttppath=true  
init.defaultbranch=master  
user.name=KANMANI MURUGHAIYAN  
user.email=kanmanicse2022@gmail.com  
core.editor=notepad++.exe -multiInst -nosession
```

Step 2: Integrate notepad++.exe to Git and make it a default editor

1. To check, if notepad++.exe execute from Git bash

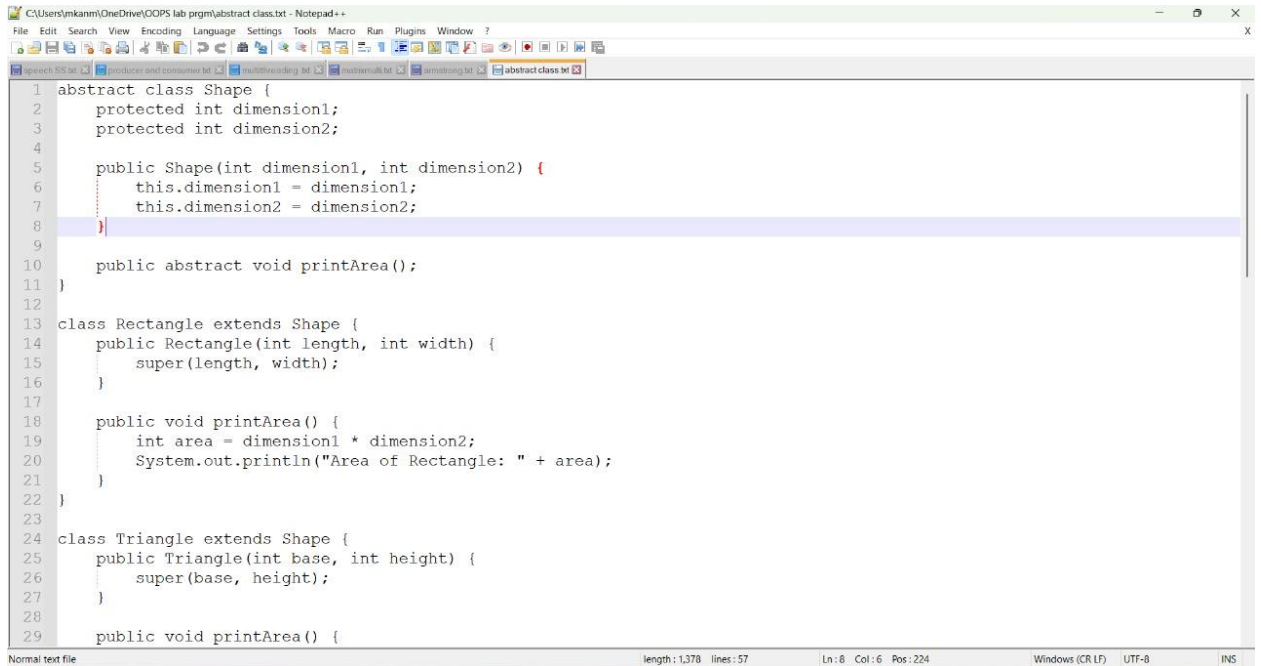
```
mkanm@Kanmani MINGW64 ~  
$ alias np='notepad++'  
  
mkanm@Kanmani MINGW64 ~  
$ git config --global core.editor "notepad++ -multiInst -nosession"
```

If Git bash could not able to recognize notepad++ command that implies notepad++.exe is not added to the environment path variable.

To add path of notepad++.exe to environment variable, go to control panel -> System -> Advanced System settings. Go to Advanced tab -> Environment variables -> Add path of notepad++.exe to the path user variable by clicking on "Edit"

2. Exit Git bash shell, open bash shell and execute

```
$ notepad++
```



```
1 abstract class Shape {
2     protected int dimension1;
3     protected int dimension2;
4
5     public Shape(int dimension1, int dimension2) {
6         this.dimension1 = dimension1;
7         this.dimension2 = dimension2;
8     }
9
10    public abstract void printArea();
11 }
12
13 class Rectangle extends Shape {
14     public Rectangle(int length, int width) {
15         super(length, width);
16     }
17
18     public void printArea() {
19         int area = dimension1 * dimension2;
20         System.out.println("Area of Rectangle: " + area);
21     }
22 }
23
24 class Triangle extends Shape {
25     public Triangle(int base, int height) {
26         super(base, height);
27     }
28
29     public void printArea() {
```

Now, notepad++ will open from Git bash shell

3. To create an alias command for notepad++.exe, execute

```
$ notepad++.exe bash -profile
```

4. To configure the editor, execute the command

```
mkanm@kanmani MINGW64 ~
$ git config --global core.editor "notepad++ -multiInst -nosession"
```

5. To verify if notepad++ is the default editor, execute the command

```
$ git config --global -e
hint: Waiting for your editor to close the file... _
```

Here '-e' option implies editor

It will show the entire global configuration as shown below,

```
mkanm@Kanmani MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=KANMANI MURUGHAIYAN
user.email=kanmanicse2022@gmail.com
core.editor=notepad++.exe -multiInst -nosession
```

Step 3: Add a file to source code repository

1. Open Git bash shell and create a new project “**GitDemo**” by executing the command

```
mkanm@Kanmani MINGW64 ~
$ mkdir GitDemo
cd GitDemo
git init
mkdir: cannot create directory 'GitDemo': File exists
Reinitialized existing Git repository in C:/Users/mkanm/GitDemo/.git/
```

2. Git bash initializes the “**GitDemo**” repository. To verify, execute the command

It will display all the hidden files in the Git “working directory”.

```
mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ ls -al
total 21
drwxr-xr-x 1 mkanm 197609  0 Aug  9 16:25 ./
drwxr-xr-x 1 mkanm 197609  0 Aug  9 16:30 ../
drwxr-xr-x 1 mkanm 197609  0 Aug  9 16:33 .git/
```

3. To create a file “**welcome.txt**” and add content to the file, execute the command

```
mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ echo "Welcome to Git Demo" > welcome.txt
```

4. To verify if the file "welcome.txt" is created, execute

```
$ ls -al
total 9
drwxr-xr-x 1 494096 1049089  0 Jan 13 12:02 ./
drwxr-xr-x 1 494096 1049089  0 Jan 13 11:54 ../
drwxr-xr-x 1 494096 1049089  0 Jan 13 12:01 .git/
-rw-r--r-- 1 494096 1049089 31 Jan 13 12:02 welcome.txt
```

5. To verify the content, execute the command

```
mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ cat welcome.txt
Welcome to Git Demo
```

6. Check the status by executing

```
mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git status
On branch master

No commits yet

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

Now the file "**welcome.txt**" is available in Git "working directory"

7. To make the file to be tracked by Git repository, execute the command

```
mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git add welcome.txt
warning: in the working copy of 'welcome.txt', LF will be replaced by CRLF the n
ext time Git touches it
```

8. To add multi line comments, we are opening default editor to comment. Execute the command

```
mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git commit
```

Notepad++ editor will open and to add multi-line comment with default editor

9. To check if local and "Working Directory" git repository are same, execute git status

```
mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git status
On branch master
nothing to commit, working tree clean
```

welcome.txt is added to the local repository.

10. Signup with GitLab and create a remote repository “**GitDemo**”

11. To pull the remote repository, execute

`git pull origin master`

12. To push the local to remote repository, execute

`git push origin master`



```
mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ cd ~/GitDemo ~
bash: cd: too many arguments

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ pwd
/c/Users/mkanm/GitDemo

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ ^[[200~git remote add origin https://gitlab.com/kanmanicse2022/GitDemo.git
bash: $'\E[200~git': command not found

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git remote add origin https://gitlab.com/kanmanicse2022/GitDemo.git

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git remote -v
origin  https://gitlab.com/kanmanicse2022/GitDemo.git (fetch)
origin  https://gitlab.com/kanmanicse2022/GitDemo.git (push)

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git push -u origin master
```

KANMANI MURUGHAIYAN
SUPERSET ID: 6407636