

Source Code Management

Course Code: CSE 2015

Slot: L3-L4



Name: Varun

SEN No.: A86605224316

Faculty: Dr Monit Kapoor

INDEX

S. No.	Lab Session Title	Page No.
1	Git Fundamentals	3
2	Installing Git on Windows	4
3	Basic CLI Commands	5
4	Vim Text Editor	8
5	Git Commands	10
6	SCM Project (Creating and Managing Repositories)	17

Lab Session 1: Git Fundamentals

Computer

A **computer** is any device capable of performing calculations, whether they are logical or mathematical.

Program/Code

A **program** (or **code**) is a set of instructions, often organized as an algorithm, that directs a computer to perform a specific task.

Need for Managing Source Code

Modern applications, such as Spotify, consist of multiple programs working together on both the frontend and backend to deliver smooth user experience.

Regular updates are essential for:

- **Fixing Bugs:** Quickly resolving errors that may occur.
- **Improving UI/UX:** Enhancing the user interface and overall experience.
- **Optimizing Performance:** Addressing and refining issues for better performance.

For programmers, effective management of source code is crucial because:

- It ensures that all files remain in context throughout the lifecycle of the program.
- It facilitates collaboration, allowing multiple developers to work together on a shared codebase.

Tools for Source Code Management

1. Git:

A version control system that runs locally on your computer. Git helps track changes and manage versions of your project.

2. GitHub:

A global, cloud-based platform that hosts Git repositories, enabling developers to share, collaborate, and contribute to projects from anywhere in the world.

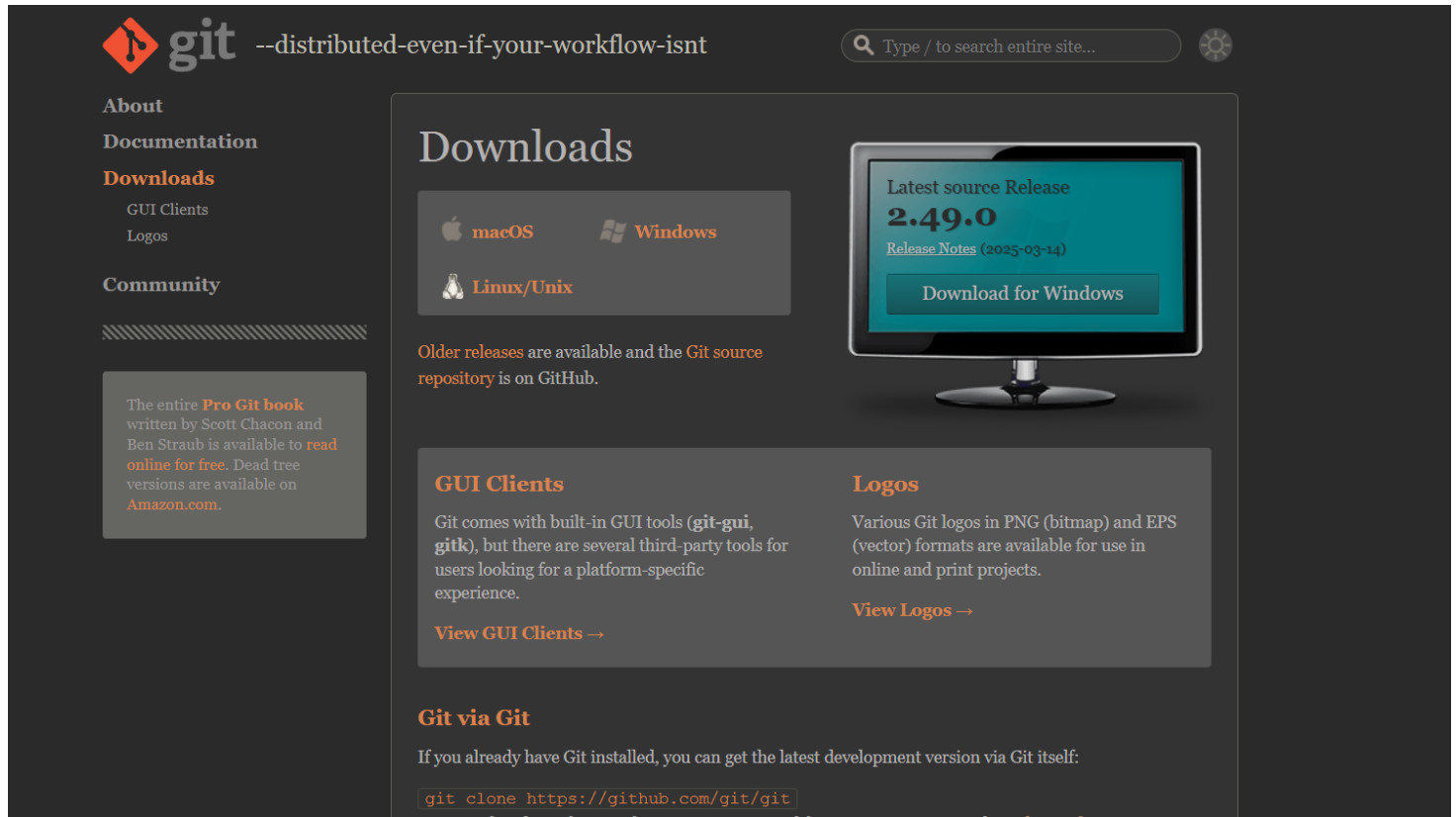
Version

A **version** in version control represents a snapshot of your project at a specific moment in time. This snapshot allows you to review, revert, or compare changes made throughout the development process.

Lab Practical 1

1. Installing Git in Windows

Step 1: Visit section 1.5 of pro git document and navigate to Windows section



Step 2: Verify Git Installation:

```
MINGW64:/c/Users/VARUN

AzureAD+VARUN@Varun MINGW64 ~
$ git --version
git version 2.49.0.windows.1
```

2. Basic CLI Commands

1) Command: pwd

Description: Prints the directory the user is working in.

```
AzureAD+VARUN@Varun MINGW64 ~  
$ pwd  
/c/Users/VARUN
```

2) Command: ls

Description: Lists all files and directories in the current directory.

```
AzureAD+VARUN@Varun MINGW64 ~  
$ ls  
AppData/      Documents/    Music/        NetHood@      Recent@       Videos/  
'Application Data'@ Downloads/    'My Documents'@ 'New folder'/' Saved Games'/' ex8/  
Contacts/     Favorites/    NTUSER.DAT   OneDrive/     Searches/     ntuser.dat.LOG1  
Cookies@      IntelGraphicsProfiles/ NTUSER.DAT{e9569593-fe94-11ef-8360-8bf8ec139016}.TM.blf Pavan/ SendTo@      ntuser.dat.LOG2  
CrossDevice/  Links/       NTUSER.DAT{e9569593-fe94-11ef-8360-8bf8ec139016}.TMContainer00000000000000000001.regtrans-ms Pictures/ 'Start Menu'@ ntuser.ini  
Desktop/     'Local Settings'@ NTUSER.DAT{e9569593-fe94-11ef-8360-8bf8ec139016}.TMContainer00000000000000000002.regtrans-ms PrintHood@ Templates@ varun.m/
```

3. Command: date

Description: shows the current date and time in a standard format

```
AzureAD+VARUN@Varun MINGW64 ~  
$ date  
Mon Jun  2 11:20:03 IST 2025
```

4. Command: clear

Description: The `clear` command in the CLI is used to clear all the current text and output displayed in the terminal window.

```
AzureAD+VARUN@Varun MINGW64 ~  
$ date  
Mon Jun  2 11:20:03 IST 2025  
  
AzureAD+VARUN@Varun MINGW64 ~  
$ time  
  
real    0m0.001s  
user    0m0.000s  
sys     0m0.000s  
  
AzureAD+VARUN@Varun MINGW64 ~  
$ |
```

5. Command: time

Description: The `time` command in the CLI is used to measure the execution time of a command or program.

```
AzureAD+VARUN@Varun MINGW64 ~  
$ time  
  
real    0m0.001s  
user    0m0.000s  
sys     0m0.000s
```

6. Command: cd 'Directory'

Description: Changes the current working directory to the desired directory.

```
AzureAD+VARUN@Varun MINGW64 ~  
$ cd documents  
  
AzureAD+VARUN@Varun MINGW64 ~/documents  
$
```

7. Command: cd ..

Description: Goes back to the previous directory.

```
AzureAD+VARUN@Varun MINGW64 ~
$ cd documents

AzureAD+VARUN@Varun MINGW64 ~/documents
$ cd ..

AzureAD+VARUN@Varun MINGW64 ~
$
```

8. Command: mkdir

Description: To create a new directory.

```
AzureAD+VARUN@Varun MINGW64 ~
$ cd documents

AzureAD+VARUN@Varun MINGW64 ~/documents
$ mkdir Trial

AzureAD+VARUN@Varun MINGW64 ~/documents
$ ls
AirDroid/           LetsView/           Trial/               'kushal form.docx'
AirDroidCast/       'My Music'@         aa.cpp
'Custom Office Templates'/'My Pictures'@     desktop.ini
'Downloads - Shortcut.Ink'* 'My Videos'@      kanak/
```

9. Command: rmdir

Description: To delete a directory

```
AzureAD+VARUN@Varun MINGW64 ~/documents
$ rmdir trial

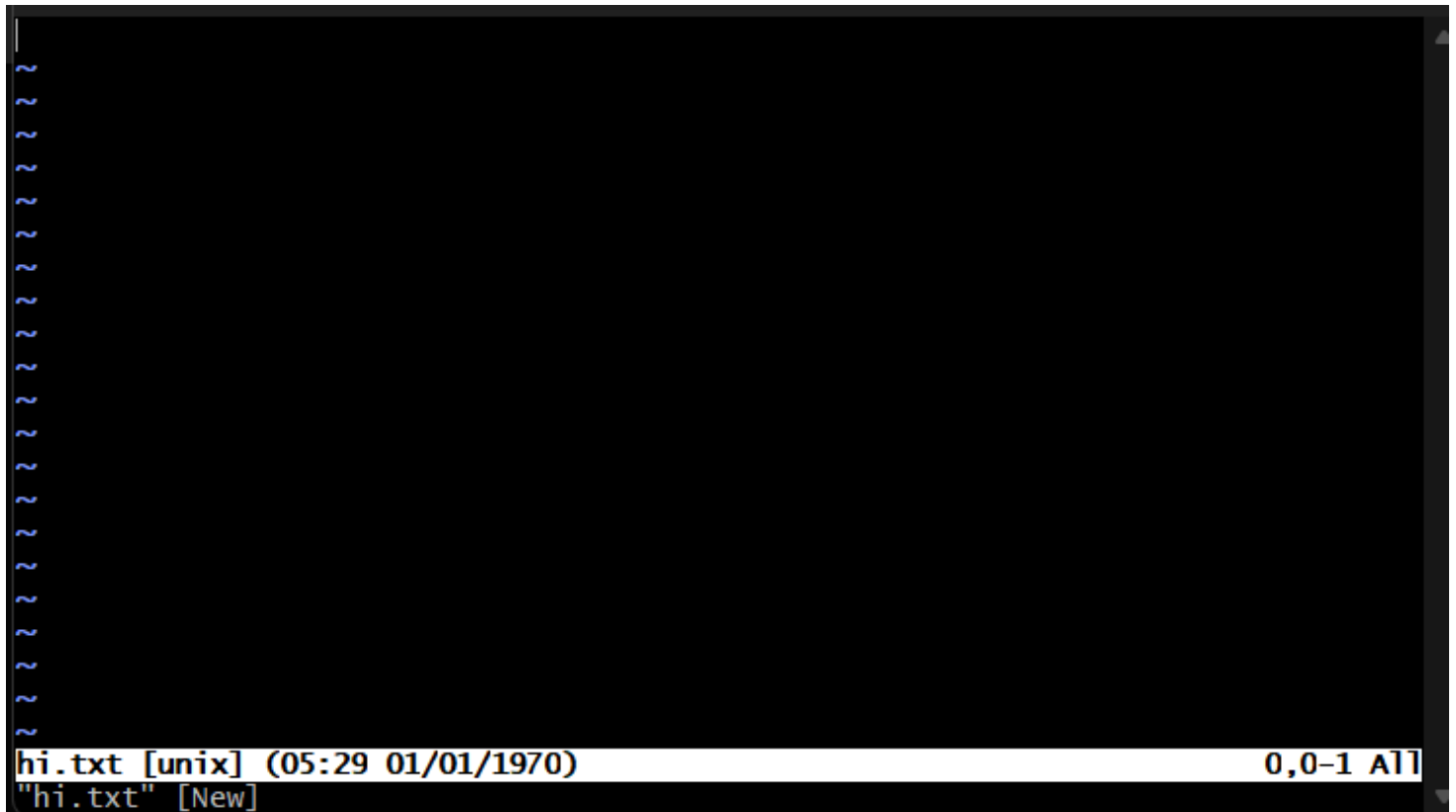
AzureAD+VARUN@Varun MINGW64 ~/documents
$ ls
AirDroid/           LetsView/           desktop.ini
AirDroidCast/       'My Music'@         kanak/
'Custom Office Templates'/'My Pictures'@     'kushal form.docx'
Documents/          'My Videos'@
'Downloads - Shortcut.Ink'* aa.cpp
```

3. Vim Text Editor

1) Command: vi hi.txt

Description: Opens (or creates) the file `hi.txt` in the Vim text editor.

```
adity@DESKTOP-71ENK80 MINGW64 ~/Documents
$ vi hi.txt
```



2) Command: i (Insert Mode)

Description: Enters insert mode in Vim to allow text input.




```
AzureAD+VARUN@Varun MINGW64 ~/documents
$ git init
Initialized empty Git repository in C:/Users/VARUN/Documents/.git/

AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$
```

3. Command: git status

Description: Displays the current status of the working directory and staging area.

```
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ git status
warning: could not open directory 'My Music/': Permission denied
warning: could not open directory 'My Pictures/': Permission denied
warning: could not open directory 'My Videos/': Permission denied
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Downloads - Shortcut.lnk
    aa.cpp
    desktop.ini
    hi.txt
    kanak/
    kushal form.docx

nothing added to commit but untracked files present (use "git add" to track)
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ |
```

4. Command: git add Test.c

Description: Add hi.txt to the staging area in preparation for a commit.

```
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ git add hi.txt
warning: in the working copy of 'hi.txt', LF will be replaced by CRLF the next time Git touches it
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$
```

5. Command: git commit -m "add file one"

Description: Commits the stage changes with the message "add file one".

```
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ git commit -m "add file one"
[master (root-commit) b096e9e] add file one
Committer: VARUN <varun11@s.amity.edu>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 4 insertions(+)
create mode 100644 hi.txt

AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ |
```

6. Command: git log

Description: Display the commit history of the repository.

```
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ git log
commit b096e9e342be4fc9bfd3f995b0088c9e8888aec8 (HEAD -> master)
Author: VARUN <varun11@s.amity.edu>
Date:   Mon Jun 2 11:32:09 2025 +0530

    add file one
```

7. Command: git clone

Description: To obtain a copy of an existing Git repository.

```
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ git clone https://github.com/AsmSafone/MusicPlayer
Cloning into 'MusicPlayer'...
remote: Enumerating objects: 266, done.
remote: Counting objects: 100% (118/118), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 266 (delta 98), reused 83 (delta 83), pack-reused 148 (from 2)
Receiving objects: 100% (266/266), 1.84 MiB | 343.00 KiB/s, done.
Resolving deltas: 100% (146/146), done.
```

8. Command: git log --oneline

Description: For generating shorter commit ID.

```
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ git log --oneline
b096e9e (HEAD -> master) add file one
```

9. Command: git diff

Description: To compare two files.

+

```
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ git diff b46a555 521a0d3
diff --git a/Trial.c b/Trial.c
index 7a64e83..e6ef922 100644
--- a/Trial.c
+++ b/Trial.c
@@ -8,5 +8,4 @@ int main()
     printf("Addition: %d", a+b);
     printf("Subtraction: %d", a-b);
     printf("Multiplication: %d", a*b);
-    printf("Division: %d", a/b);
 }
```

10. Command: git remote add "Variable"

Description: To connect with the Users GitHub account.

```
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ git remote add origin https://github.com/Varundharne/Labfile.git
```

11. Command: git remote

Description: To check the status of the repositories connected with the Users account.

```
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ git remote
origin
```

12. Command: git push -u "Variable" master

Description: To push all the files to the Users account.

```
AzureAD+VARUN@Varun MINGW64 ~/documents (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 237 bytes | 237.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Varundharne/Labfile.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

13. Command: git merge "File_Name" -m "comment"

Description: To merge a branch with main branch.

```
AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git merge adding_sort
Updating 612a79b..594216b
Fast-forward
 hello2.cpp | 246 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 246 insertions(+)
 create mode 100644 hello2.cpp
```


SCM Project

The project is to make a repository in GitHub, make 3 branches and merge it with the main branch and access all 4 team-mate's repositories, fork it, clone it, make some changes and merge them.

First, make your own repositories and make 3 branches and add files and merge with the main branch.

1) Go to the directory on your computer

```
AzureAD+VARUN@Varun MINGW64 ~  
$ cd 'C:\Users\VARUN\Source'
```

2) Git init the folder

```
AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source  
$ git init  
Initialized empty Git repository in C:/Users/VARUN/source/.git/
```

3) Make three branches

```
AzureAD+VARUN@Varun MINGW64 ~/Source (master)  
$ git branch master  
AzureAD+VARUN@Varun MINGW64 ~/Source (master)  
$ git branch Update_Student_Record  
AzureAD+VARUN@Varun MINGW64 ~/Source (master)  
$ git branch adding_sort
```

4) Checkout to the first branch

```
AzureAD+VARUN@Varun MINGW64 ~/Source (master)  
$ git checkout master  
M      hello.cpp  
Already on 'master'  
Your branch is up to date with 'origin/master'.
```

5) Create a html file, add the file and commit it

```
AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source
$ vi hello.cpp

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source
$ git init
Initialized empty Git repository in C:/Users/VARUN/source/.git/

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git add .
warning: in the working copy of 'hello.cpp', LF will be replaced by CRLF the nex

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git config user.email "varun11@s.amity.edu"

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git commit -m "First Commit"
[master (root-commit) 3d54b1e] First Commit
1 file changed, 192 insertions(+)
create mode 100644 hello.cpp
```

6) Reopen the file, make changes, add the file again and commit

```
AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git add hello.cpp
```

7) Checkout to the second branch, do the steps again for adding and committing a file 3 times.

```

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git checkout Update_Student_Record
M      hello.cpp
Switched to branch 'Update_Student_Record'

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (Update_Student_Record)
$ git fetch origin

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (Update_Student_Record)
$ git merge origin/master
git: 'merge' is not a git command. See 'git --help'.

The most similar command is
    merge

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (Update_Student_Record)
$ git merge master\
>
git: 'marge' is not a git command. See 'git --help'.

The most similar command is
    merge

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (Update_Student_Record)
$ git merge master
Updating 612a79b..594216b
Fast-forward
 hello2.cpp | 246 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 246 insertions(+)
 create mode 100644 hello2.cpp

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (Update_Student_Record)
$ git push Update_Student_Record
fatal: 'Update_Student_Record' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (Update_Student_Record)
$ git push origin Update_Student_Record
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'Update_Student_Record' on GitHub by visiting:
remote:   https://github.com/Varundharne/Sourcecode/pull/new/Update_Student_Record
remote:
To https://github.com/Varundharne/Sourcecode.git
 * [new branch]      Update_Student_Record -> Update_Student_Record

```

8) Checkout to the third branch after committing 3 changes in the previous branch

```
AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git checkout adding_sort
M       hello.cpp
A       hello2.cpp
Switched to branch 'adding_sort'

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (adding_sort)
$ git pull origin master
From https://github.com/Varundharne/Sourcecode
* branch      master       -> FETCH_HEAD
Already up to date.

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (adding_sort)
$ git commit -m "Adding sort feature"
[adding_sort 594216b] Adding sort feature
1 file changed, 246 insertions(+)
create mode 100644 hello2.cpp

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (adding_sort)
$ git pull origin master
From https://github.com/Varundharne/Sourcecode
* branch      master       -> FETCH_HEAD
Already up to date.

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (adding_sort)
$ git merge adding_sort
Already up to date.

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (adding_sort)
$ git checkout master
M       hello.cpp
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git push origin master
Everything up-to-date

AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

9) Checking out to the main branch

```
AzureAD+VARUN@Varun MINGW64 ~/Source (adding_sort)
$ git checkout master
M       hello.cpp
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

10) Merging the first branch (Update_Student_Record) to the master branch

```
AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git merge Update_Student_Record
Updating 3d54b1e..612a79b
Fast-forward
 hello1.cpp | 223 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 223 insertions(+)
 create mode 100644 hello1.cpp
```

11) Same way merging the rest branches with the main branch as well

```
AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git merge adding_sort
Updating 612a79b..594216b
Fast-forward
 hello2.cpp | 246 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 246 insertions(+)
 create mode 100644 hello2.cpp
```

12) Using “git log –oneline” to check all the commits

```
AzureAD+VARUN@Varun MINGW64 ~/Source (master)
$ git log --oneline
594216b (HEAD -> master, origin/master, origin/adding_sort, origin/Update_Student_Record, origin/HEAD, adding_sort, Update_Student_Record) Adding sort feature
612a79b Adding Update Feature
3d54b1e First Commit
```

13) Using git remote to add GitHub repository

```
AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git remote add origin https://github.com/Varundharne/Sourcecode.git
```

14) Pushing all the commits and file in the GitHub repository

```
AzureAD+VARUN@Varun MINGW64 /c/Users/varun/source (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1.47 KiB | 1.47 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Varundharne/Sourcecode.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```