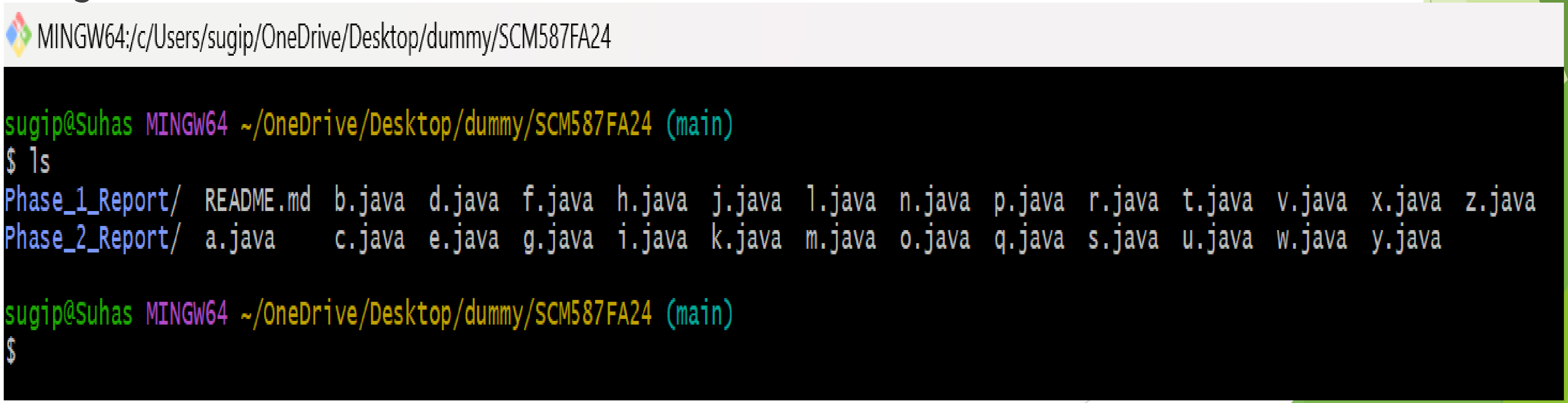
Git

An Overview

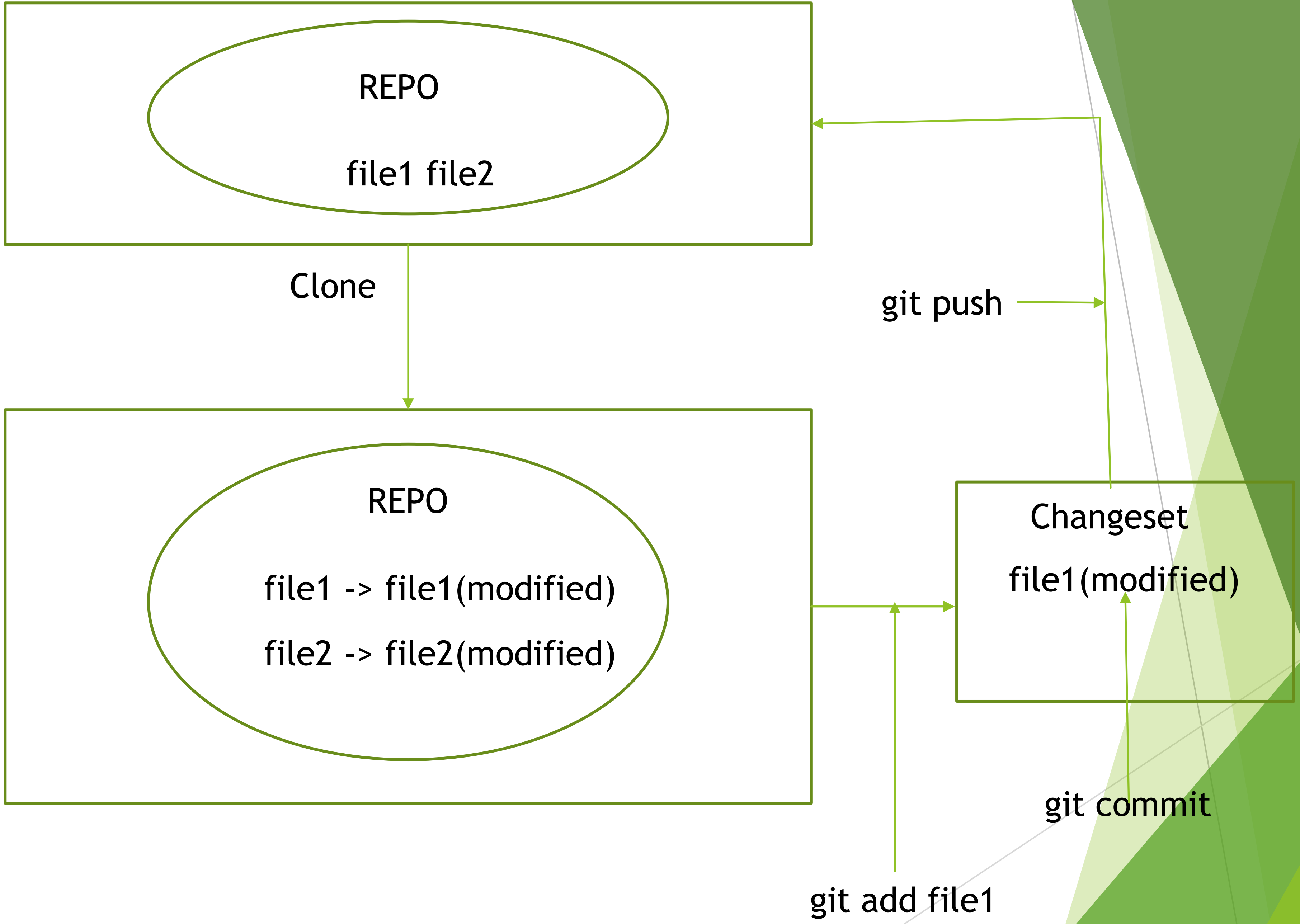
- We have seen how GitHub works
- This tutorial gives you an introduction to the GIT commands that you will be working with.
- The Git Tool helps us modify and update files in a repository.
- You will need to familiarize yourself with the Git commands. You can install GitBash from this LINK (Command line prompt to run the GIT commands)
- The GIT commands explained in this tutorial will guide you through your assignments.



MINGW64:/c/Users/sugip/OneDrive/Desktop/dummy/SCM587FA24

```
sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ ls
Phase_1_Report/  README.md  b.java  d.java  f.java  h.java  j.java  l.java  n.java  p.java  r.java  t.java  v.java  x.java  z.java
Phase_2_Report/  a.java     c.java  e.java  g.java  i.java  k.java  m.java  o.java  q.java  s.java  u.java  w.java  y.java

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$
```

Owner - Bob
GitHub Server



REPO

file1 file2

Clone

git push

REPO

file1 -> file1(modified)

file2 -> file2(modified)

Bob’s
Laptop

Changeset

file1(modified)

git commit

git add file1

Owner - Bob
GitHub Server

REPO

file1 file2

1 2 3 4 master

1 2 branch1

git push

Bob's
Laptop

REPO

file1 -> file1(modified)

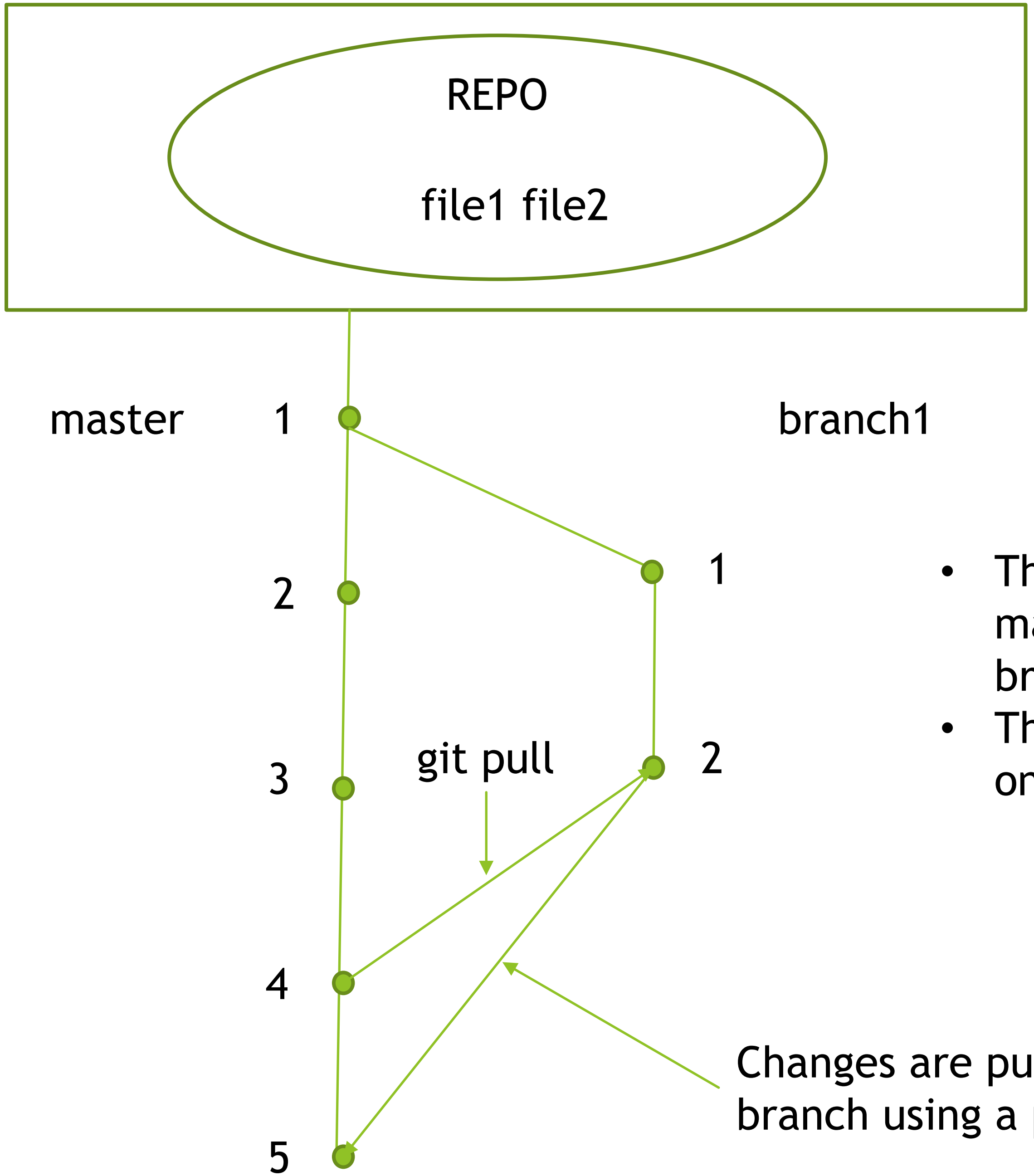
file2 -> file2(modified)

Changeset

file1(modified)

file2(modified)

git add .
git commit



REPO

file1 file2

master

1

2

3

4

5

branch1

1

2

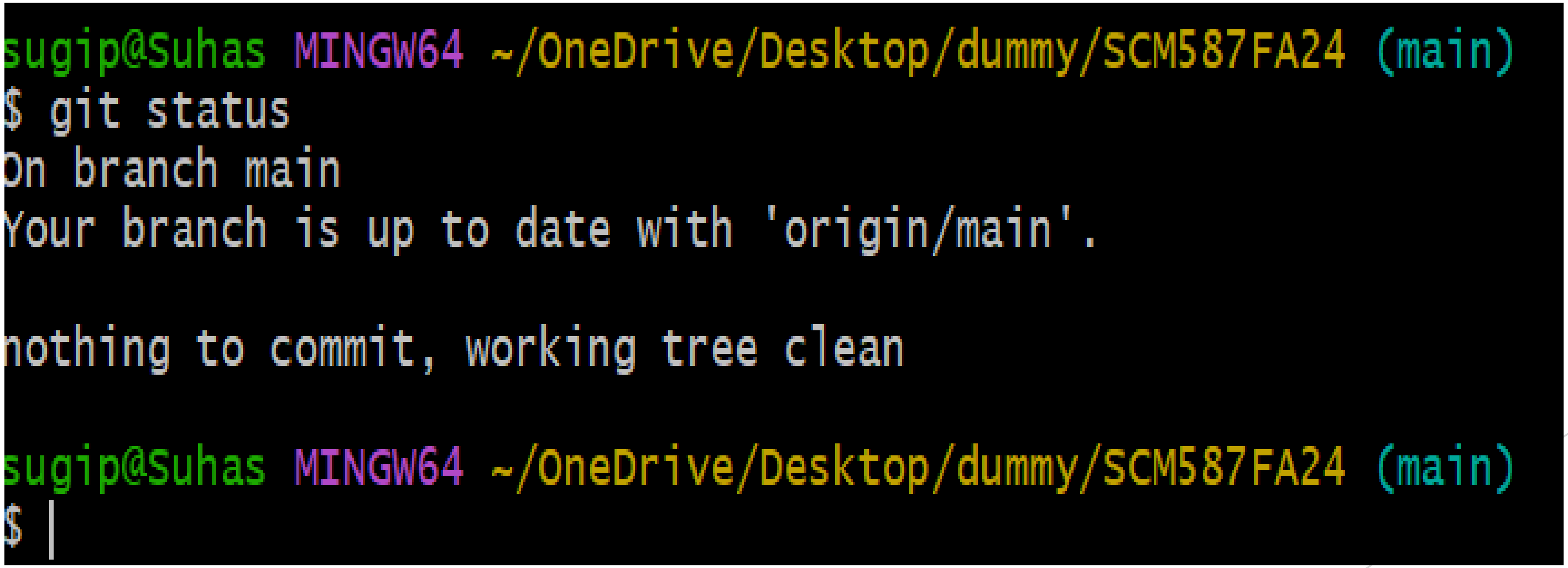
git pull

- The latest version from the master branch is pulled into the branch1
- Then the changes are merged onto the master branch

Changes are pulled onto the master branch using a pull request

git status

- The status command shows you the current status of the repository.
- It shows the branch that you are working in.
- It identifies the changes that are yet to be committed.
- The following screenshot from the Git Shell shows the output of the git status.



```
sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ |
```

git add

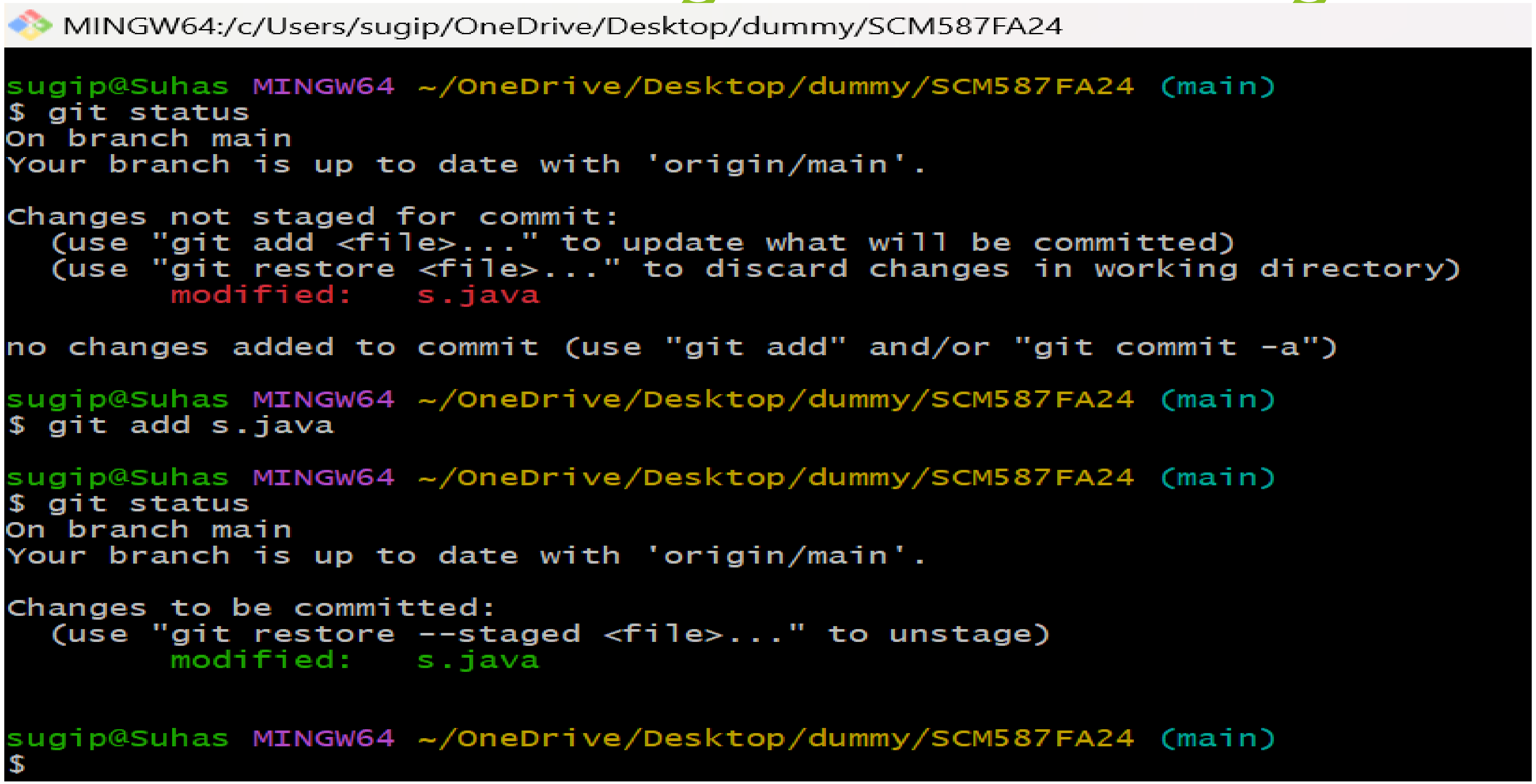
- Git add command adds any modified file or new file to the list of files to be committed to a branch.
- So any modified file you want to commit to a branch must first be added.

  - <u>Syntax</u>
    - **git add *filename***   ***- to add a single file***
    - **git add .**   ***- to add all files in the current directory***

git commit

- Now you have a repository that is cloned, and you want to modify a file and commit it.
- Once the file is modified and ready to be committed, in the Git shell type the following commands.
  - **[bash]$ git add *filename***
  - **[bash]$ git commit -m "*Commit message here*"**

Screenshot of Adding and Committing



MINGW64:/c/Users/sugip/OneDrive/Desktop/dummy/SCM587FA24

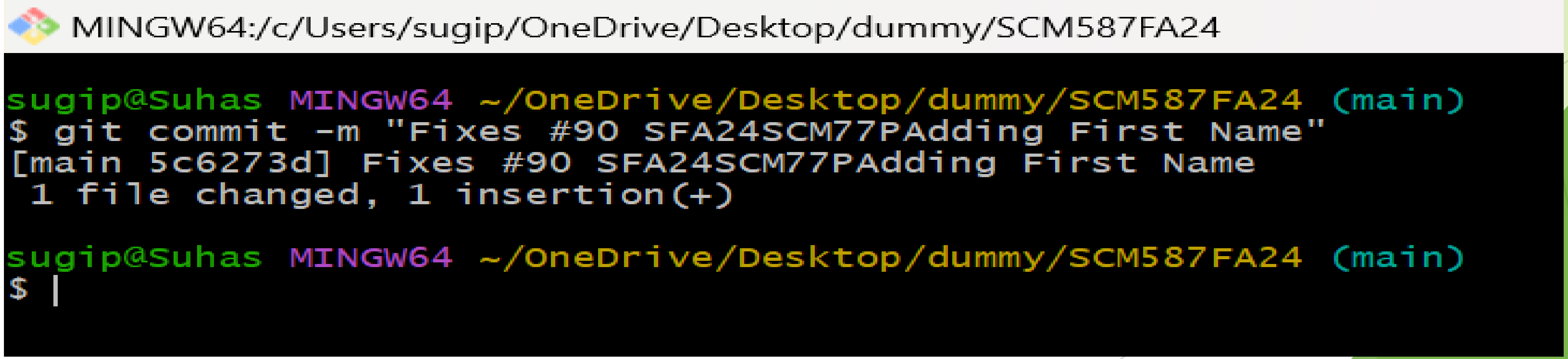
```
sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   s.java

no changes added to commit (use "git add" and/or "git commit -a")

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git add s.java

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   s.java


sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$
```

Fixes option

- We have seen in GitHub how to create issues.
- Once you have created issue that issue will open till you manually close it.
- This can be done through Git.
- Giving the “Fixes #<Issue number> during commit automatically closes the issue on GitHub.

Syntax

- **git commit -m “Fixes #*<Issue Number> Your message here*”**

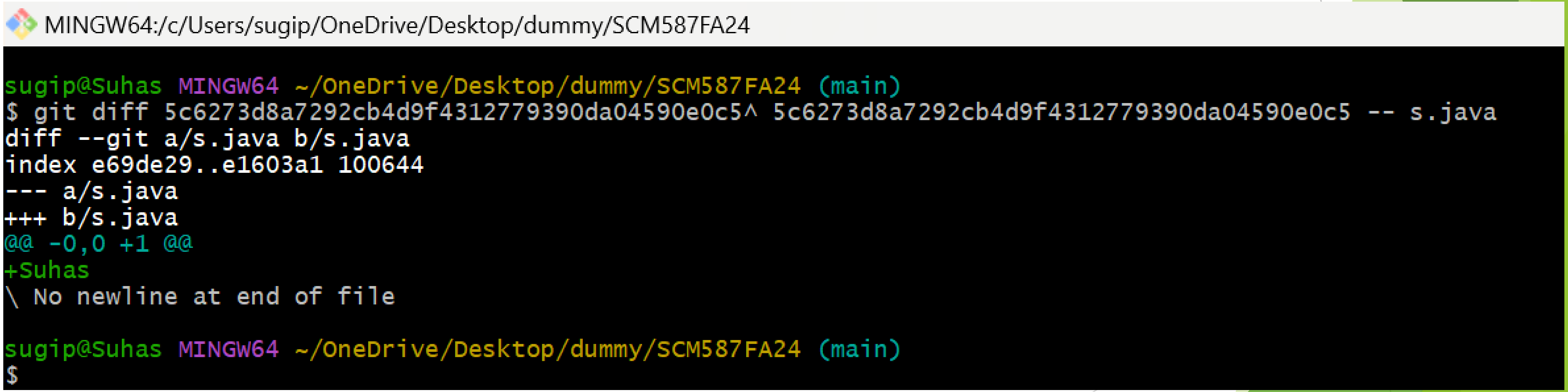


MINGW64:/c/Users/sugip/OneDrive/Desktop/dummy/SCM587FA24

```
sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git commit -m "Fixes #90 SFA24SCM77PAdding First Name"
[main 5c6273d] Fixes #90 SFA24SCM77PAdding First Name
 1 file changed, 1 insertion(+)

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ |
```

git diff

- The diff command shows the changes that have been made in a file or a directory from the last commit.
  - <u>Syntax</u>
    - git diff - shows diff for the whole root directory
    - git diff *filename* - shows diff for the file mentioned

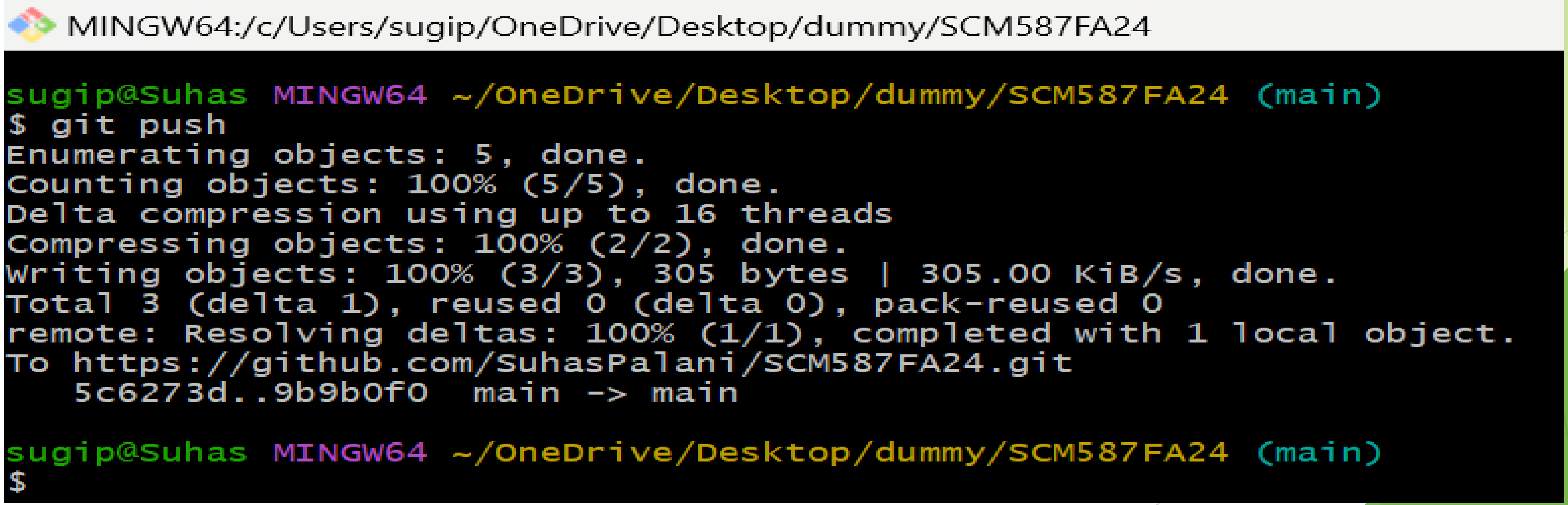


MINGW64:/c/Users/sugip/OneDrive/Desktop/dummy/SCM587FA24

```
sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git diff 5c6273d8a7292cb4d9f4312779390da04590e0c5^ 5c6273d8a7292cb4d9f4312779390da04590e0c5 -- s.java
diff --git a/s.java b/s.java
index e69de29..e1603a1 100644
--- a/s.java
+++ b/s.java
@@ -0,0 +1 @@
+Suhas
\ No newline at end of file

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$
```

git push

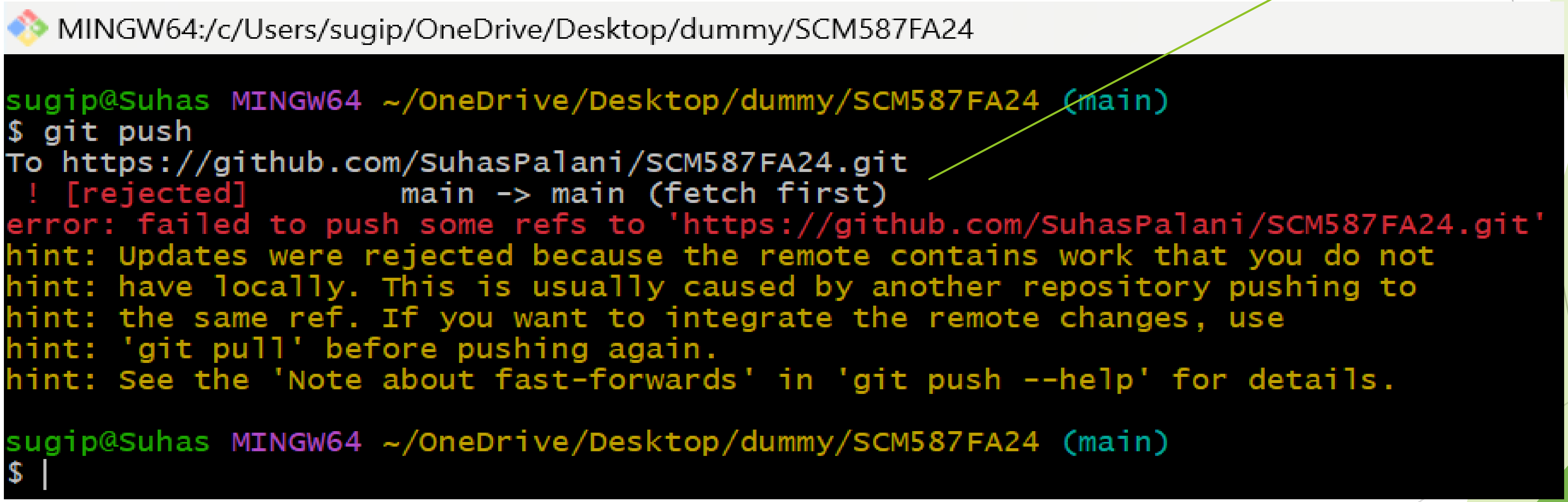
- Now that we have committed the changes that we have made we want the changes to be reflected in the centralized repository
- This will enable other users to see these changes as well.
- Git push command pushes the changes that we have committed to the centralized repository.
  - Syntax
    - **git push**



MINGW64:/c/Users/sugip/OneDrive/Desktop/dummy/SCM587FA24

```
sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 305 bytes | 305.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/SuhasPalani/SCM587FA24.git
   5c6273d..9b9b0f0  main -> main

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$
```

What happens when another user commits?



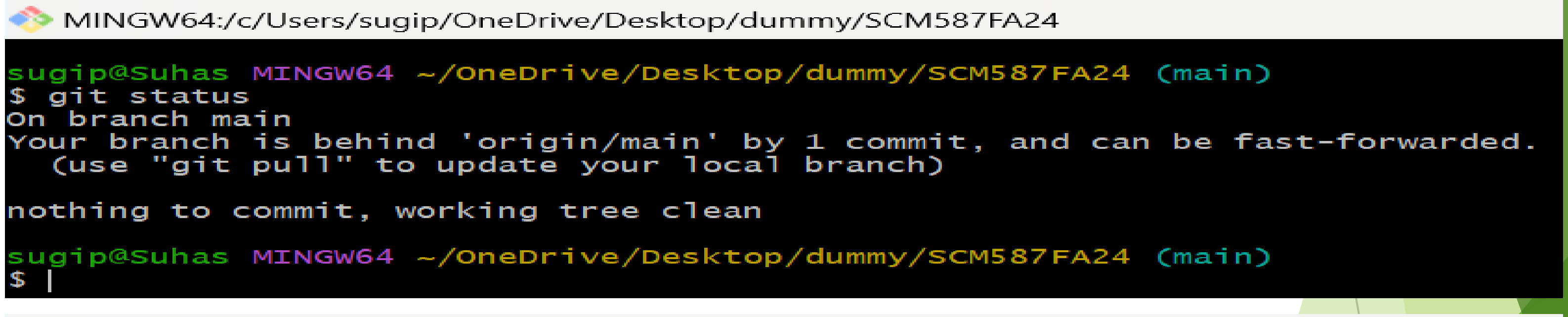
MINGW64:/c/Users/sugip/OneDrive/Desktop/dummy/SCM587FA24

```
sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git push
To https://github.com/SuhasPalani/SCM587FA24.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/SuhasPalani/SCM587FA24.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ |
```

Push to repository getting rejected

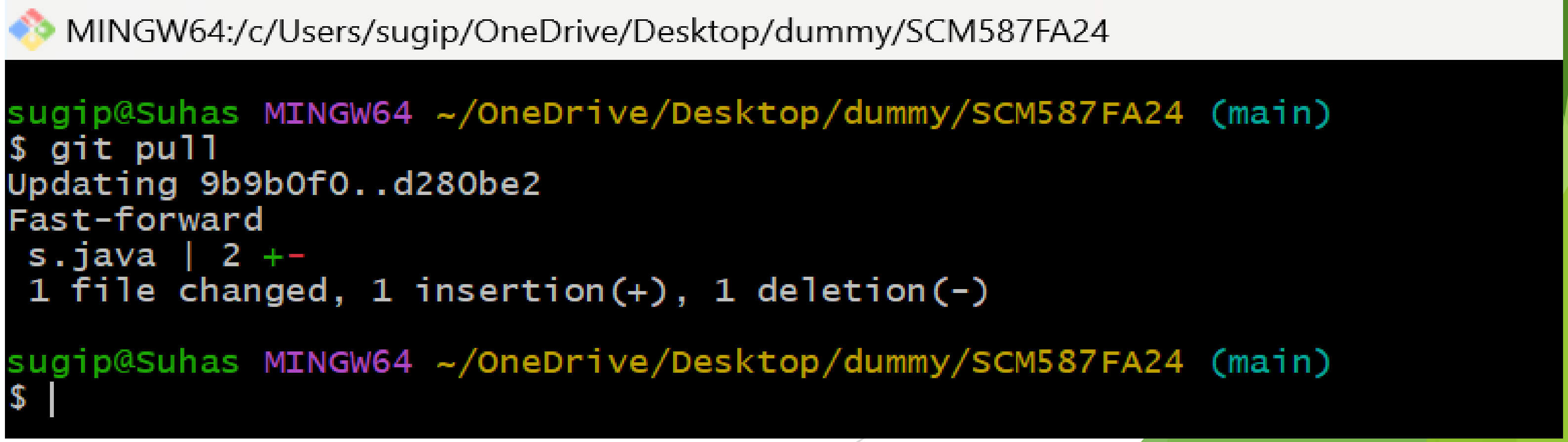
git pull

- In the previous screenshot the git push failed because in the main repository newer changes are available which is not present in our repository.
- This means that if we upload our changes without including the changes in the repository, it will overwrite those changes.
- That is why git rejected the push command. Git pull commands updates the current repository with the newer changes available in the repository.
  - Syntax
    - git pull



MINGW64:/c/Users/sugip/OneDrive/Desktop/dummy/SCM587FA24

```
sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working tree clean

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ |
```



MINGW64:/c/Users/sugip/OneDrive/Desktop/dummy/SCM587FA24

```
sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git pull
Updating 9b9b0f0..d280be2
Fast-forward
 s.java | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ |
```

git branch

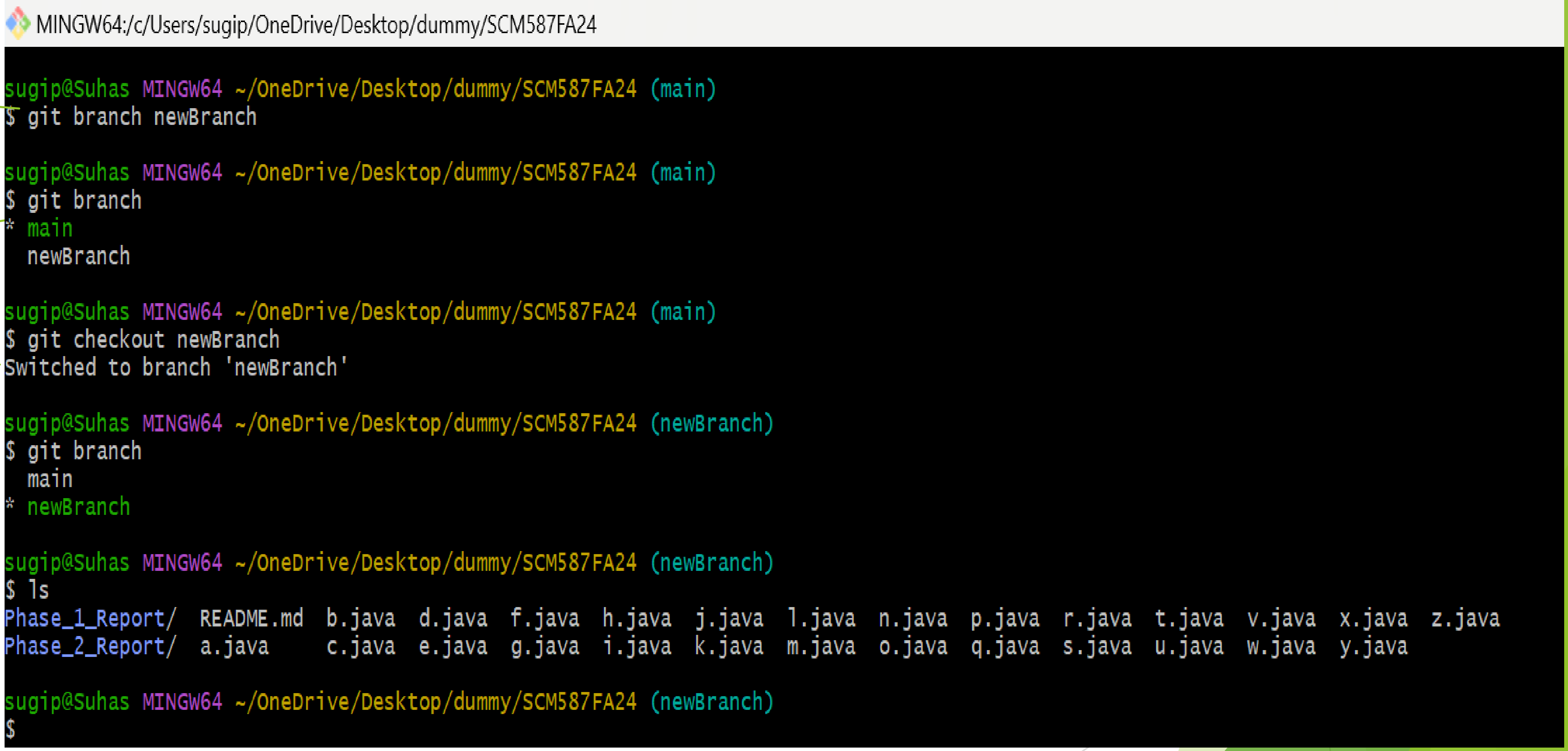
- This command creates a new branch from the main branch.
- The branch name is specified, and git creates a new branch with the name.

  - **Syntax**
    - **git branch *branchname* <*commit number*>**

git checkout

- This command is used to revert the changes to a previous commit.
  - **Syntax**
    - **git checkout *commitnumber***
- Checkout command can also help us switch to a different branch from the current branch.
  - **Syntax**
    - **git checkout -b *branchname commitnumber***

Creating and checking out a branch



```
sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git branch newBranch

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git branch
* main
  newBranch

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ git checkout newBranch
Switched to branch 'newBranch'

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (newBranch)
$ git branch
  main
* newBranch

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (newBranch)
$ ls
Phase_1_Report/  README.md  b.java  d.java  f.java  h.java  j.java  l.java  n.java  p.java  r.java  t.java  v.java  x.java  z.java
Phase_2_Report/  a.java     c.java  e.java  g.java  i.java  k.java  m.java  o.java  q.java  s.java  u.java  w.java  y.java

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (newBranch)
$
```

Creating branch newBranch

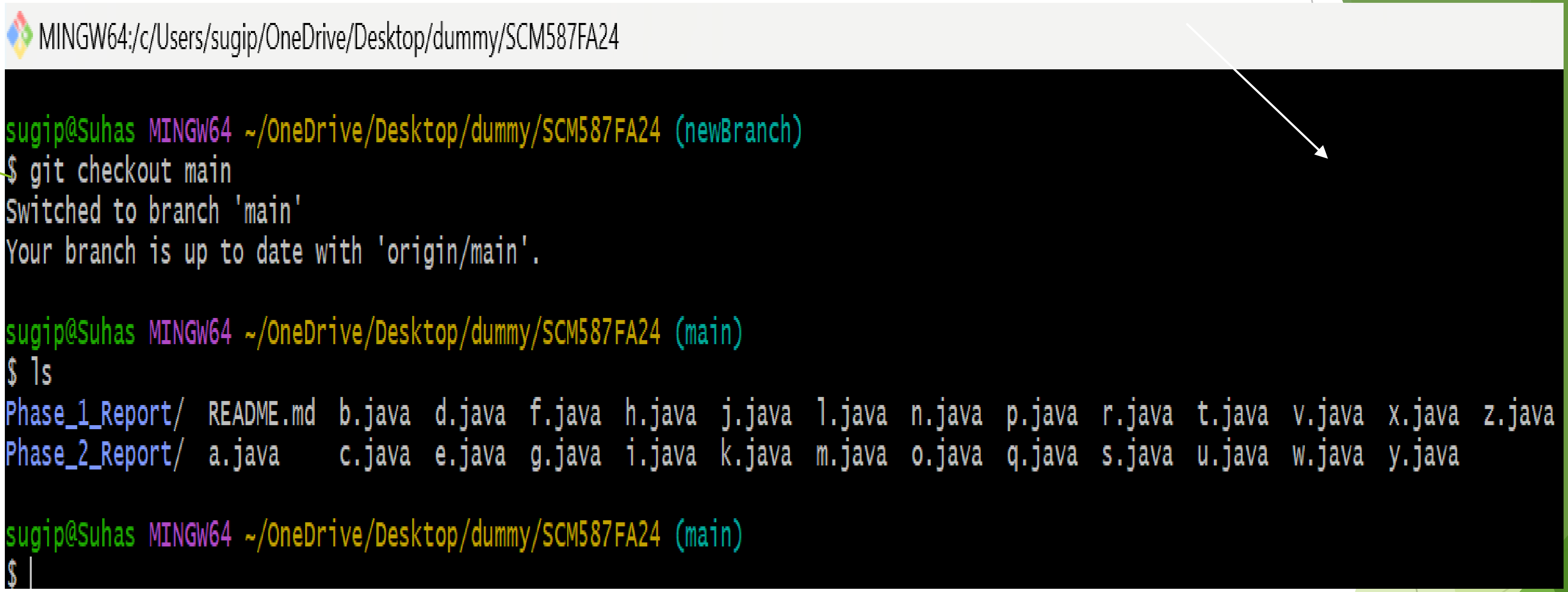
Check for branches

Checking out newBranch

Listing out all files in the newBranch

Creating and checking out a branch

Checking out main branch



```
sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (newBranch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ ls
Phase_1_Report/  README.md  b.java  d.java  f.java  h.java  j.java  l.java  n.java  p.java  r.java  t.java  v.java  x.java  z.java
Phase_2_Report/  a.java     c.java  e.java  g.java  i.java  k.java  m.java  o.java  q.java  s.java  u.java  w.java  y.java

sugip@Suhas MINGW64 ~/OneDrive/Desktop/dummy/SCM587FA24 (main)
$ |
```

Questions?