```
Experimentation.txt  README.md  Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main|MERGING)
$ cd Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main|MERGING)
$ ls
Intro  Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main|MERGING)
$ git commit
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
U       Experimentation.txt
U       Task-1.txt
U       Task-1/Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main|MERGING)
$ git commit -m "This is an updated format"
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
U       Experimentation.txt
U       Task-1.txt
U       Task-1/Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main|MERGING)
$ ls
Intro  Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main|MERGING)
$ cd ..

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main|MERGING)
$ git-merge
bash: git-merge: command not found

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main|MERGING)
$ --no-ff
bash: --no-ff: command not found

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main|MERGING)
$ ^C

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main|MERGING)
$ gh pr merge pull-request
bash: gh: command not found

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main|MERGING)
$ git pr merge pull-request
git: 'pr' is not a git command. See 'git --help'.

The most similar command is
        p4
```

```
Type
    -t, --type <>          value is given this type
    --bool                 value is "true" or "false"
    --int                  value is decimal number
    --bool-or-int          value is --bool or --int
    --bool-or-str          value is --bool or string
    --path                 value is a path (file or directory name)
    --expiry-date          value is an expiry date

Other
    -z, --null             terminate values with NUL byte
    --name-only            show variable names only
    --includes             respect include directives on lookup
    --show-origin          show origin of config (file, standard input, blob, command line)
    --show-scope           show scope of config (worktree, local, global, system, command)
    --default <value>      with --get, use default value when missing entry


Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main)
$ git config --list --show-origin
file:C:/Program Files/Git/etc/gitconfig diff.astextplain.textconv=astextplain
file:C:/Program Files/Git/etc/gitconfig filter.lfs.clean=git-lfs clean -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.smudge=git-lfs smudge -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.process=git-lfs filter-process
file:C:/Program Files/Git/etc/gitconfig filter.lfs.required=true
file:C:/Program Files/Git/etc/gitconfig http.sslbackend=openssl
file:C:/Program Files/Git/etc/gitconfig http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
file:C:/Program Files/Git/etc/gitconfig core.autocrlf=true
file:C:/Program Files/Git/etc/gitconfig core.fscache=true
file:C:/Program Files/Git/etc/gitconfig core.symlinks=false
file:C:/Program Files/Git/etc/gitconfig pull.rebase=false
file:C:/Program Files/Git/etc/gitconfig credential.helper=manager-core
file:C:/Program Files/Git/etc/gitconfig credential.https://dev.azure.com.usehttppath=true
file:C:/Program Files/Git/etc/gitconfig init.defaultbranch=master
file:C:/Users/Sri Nirmal Kumaar/.gitconfig        user.name=Sri Nirmal Kumaar
file:C:/Users/Sri Nirmal Kumaar/.gitconfig        user.email=sri_nirmal@protonmail.com
file:.git/config          core.repositoryformatversion=0
file:.git/config          core.filemode=false
file:.git/config          core.bare=false
file:.git/config          core.logallrefupdates=true
file:.git/config          core.symlinks=false
file:.git/config          core.ignorecase=true
file:.git/config          remote.origin.url=https://github.com/NirmalMJ/Cognizance.git
file:.git/config          remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
file:.git/config          branch.main.remote=origin
file:.git/config          branch.main.merge=refs/heads/main


Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main)
$
```

```
'ganesh endsem.bak'
'ganesh endsem.dwg'
'gears of war 4 password.txt'
 hosts
 plot.log
 test.flp
 untitled.flp

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents
$ cd Git_Code/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--super-prefix=<path>] [--config-env=<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
    clone     Clone a repository into a new directory
    init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
    add       Add file contents to the index
    mv        Move or rename a file, a directory, or a symlink
    restore   Restore working tree files
    rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
    bisect    Use binary search to find the commit that introduced a bug
    diff      Show changes between commits, commit and working tree, etc
    grep      Print lines matching a pattern
    log       Show commit logs
    show      Show various types of objects
    status    Show the working tree status

grow, mark and tweak your common history
    branch    List, create, or delete branches
    commit    Record changes to the repository
    merge     Join two or more development histories together
    rebase    Reapply commits on top of another base tip
    reset     Reset current HEAD to the specified state
    switch    Switch branches
    tag       Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
    fetch     Download objects and refs from another repository
    pull      Fetch from and integrate with another repository or a local branch
    push      Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ mkdir Cognizance
```

```
 hosts
 plot.log
 test.flp
 untitled.flp

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents
$ cd Git_Code/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--super-prefix=<path>] [--config-env=<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
    clone     Clone a repository into a new directory
    init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
    add       Add file contents to the index
    mv        Move or rename a file, a directory, or a symlink
    restore   Restore working tree files
    rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
    bisect    Use binary search to find the commit that introduced a bug
    diff      Show changes between commits, commit and working tree, etc
    grep      Print lines matching a pattern
    log       Show commit logs
    show      Show various types of objects
    status    Show the working tree status

grow, mark and tweak your common history
    branch    List, create, or delete branches
    commit    Record changes to the repository
    merge     Join two or more development histories together
    rebase    Reapply commits on top of another base tip
    reset     Reset current HEAD to the specified state
    switch    Switch branches
    tag       Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
    fetch     Download objects and refs from another repository
    pull      Fetch from and integrate with another repository or a local branch
    push      Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ mkdir Cognizance

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ |
```

 untitled.flp

```
Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents
$ cd Git_Code/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--super-prefix=<path>] [--config-env=<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
    clone     Clone a repository into a new directory
    init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
    add       Add file contents to the index
    mv        Move or rename a file, a directory, or a symlink
    restore   Restore working tree files
    rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
    bisect    Use binary search to find the commit that introduced a bug
    diff      Show changes between commits, commit and working tree, etc
    grep      Print lines matching a pattern
    log       Show commit logs
    show      Show various types of objects
    status    Show the working tree status

grow, mark and tweak your common history
    branch    List, create, or delete branches
    commit    Record changes to the repository
    merge     Join two or more development histories together
    rebase    Reapply commits on top of another base tip
    reset     Reset current HEAD to the specified state
    switch    Switch branches
    tag       Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
    fetch     Download objects and refs from another repository
    pull      Fetch from and integrate with another repository or a local branch
    push      Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ mkdir Cognizance

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ cd Cognizance/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance
$
```

```
$ vi trail.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Task-1 (master)
$ cd ..

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ rm Task-1/
rm: cannot remove 'Task-1/': Is a directory

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ del Task-1/
bash: del: command not found

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ delete Task-1/
bash: delete: command not found

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ git clone git@github.com:NirmalMJ/Cognizance.git
Cloning into 'Cognizance'...
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

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

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ git clone https://github.com/NirmalMJ/Cognizance.git
Cloning into 'Cognizance'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 4), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (20/20), 5.07 KiB | 5.07 MiB/s, done.
Resolving deltas: 100% (4/4), done.

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ ls
Cognizance/  Task-1/  test.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ rm -r Task-1/
rm: remove write-protected regular file 'Task-1/.git/objects/23/7b12cfc198e620a583e1372c8a1ccfabad5316'? y
rm: remove write-protected regular file 'Task-1/.git/objects/51/b978b5b4f0e97b998d8b9d65a9ce7e000899cb'? yes
rm: remove write-protected regular file 'Task-1/.git/objects/65/b34ac73093fbcff81d9c0ddd7d466f2e2361d9'? 1
rm: cannot remove 'Task-1/.git/objects/65': Directory not empty
rm: remove write-protected regular file 'Task-1/.git/objects/89/13a6ec711324e9a0028eef5cb523b8346860ad'? 0
rm: cannot remove 'Task-1/.git/objects/89': Directory not empty
rm: remove write-protected regular file 'Task-1/.git/objects/b7/6a8be750dd89c1c0d14bfdc5a512a0beb90e61'? 1
rm: cannot remove 'Task-1/.git/objects/b7': Directory not empty
rm: remove write-protected regular file 'Task-1/.git/objects/bf/e2701dff2af4090ed881d3d08180081f096cf6'?

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ rmdir Task-1/
rmdir: failed to remove 'Task-1/': Directory not empty

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$
```

```
Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code (master)
$ ls
Task-1/  test.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code (master)
$ git config --list --show-
error: ambiguous option: show- (could be --show-origin or --show-scope)
usage: git config [<options>]

Config file location
    --global              use global config file
    --system              use system config file
    --local               use repository config file
    --worktree            use per-worktree config file
    -f, --file <file>     use given config file
    --blob <blob-id>      read config from given blob object

Action
    --get                 get value: name [value-pattern]
    --get-all             get all values: key [value-pattern]
    --get-regexp          get values for regexp: name-regex [value-pattern]
    --get-urlmatch        get value specific for the URL: section[.var] URL
    --replace-all         replace all matching variables: name value [value-pattern]
    --add                 add a new variable: name value
    --unset               remove a variable: name [value-pattern]
    --unset-all           remove all matches: name [value-pattern]
    --rename-section      rename section: old-name new-name
    --remove-section      remove a section: name
    -l, --list            list all
    --fixed-value         use string equality when comparing values to 'value-pattern'
    -e, --edit            open an editor
    --get-color           find the color configured: slot [default]
    --get-colorbool       find the color setting: slot [stdout-is-tty]

Type
    -t, --type <>         value is given this type
    --bool                value is "true" or "false"
    --int                 value is decimal number
    --bool-or-int         value is --bool or --int
    --bool-or-str         value is --bool or string
    --path                value is a path (file or directory name)
    --expiry-date         value is an expiry date

Other
    -z, --null            terminate values with NUL byte
    --name-only           show variable names only
    --includes            respect include directives on lookup
    --show-origin         show origin of config (file, standard input, blob, command line)
    --show-scope          show scope of config (worktree, local, global, system, command)
    --default <value>     with --get, use default value when missing entry


Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code (master)
$ ls
Task-1/  test.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code (master)
```

```
      --expiry-date          value is an expiry date

Other
    -z, --null             terminate values with NUL byte
    --name-only            show variable names only
    --includes             respect include directives on lookup
    --show-origin          show origin of config (file, standard input, blob, command line)
    --show-scope           show scope of config (worktree, local, global, system, command)
    --default <value>      with --get, use default value when missing entry

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code (master)
$ ls
Task-1/  test.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code (master)
$ cd Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Task-1 (master)
$ git config --list --show-origin
file:C:/Program Files/Git/etc/gitconfig diff.astextplain.textconv=astextplain
file:C:/Program Files/Git/etc/gitconfig filter.lfs.clean=git-lfs clean -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.smudge=git-lfs smudge -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.process=git-lfs filter-process
file:C:/Program Files/Git/etc/gitconfig filter.lfs.required=true
file:C:/Program Files/Git/etc/gitconfig http.sslbackend=openssl
file:C:/Program Files/Git/etc/gitconfig http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
file:C:/Program Files/Git/etc/gitconfig core.autocrlf=true
file:C:/Program Files/Git/etc/gitconfig core.fscache=true
file:C:/Program Files/Git/etc/gitconfig core.symlinks=false
file:C:/Program Files/Git/etc/gitconfig pull.rebase=false
file:C:/Program Files/Git/etc/gitconfig credential.helper=manager-core
file:C:/Program Files/Git/etc/gitconfig credential.https://dev.azure.com.usehttppath=true
file:C:/Program Files/Git/etc/gitconfig init.defaultbranch=master
file:C:/Users/Sri Nirmal Kumaar/.gitconfig        user.name=Sri Nirmal Kumaar
file:C:/Users/Sri Nirmal Kumaar/.gitconfig        user.email=sri_nirmal@protonmail.com
file:.git/config           core.repositoryformatversion=0
file:.git/config           core.filemode=false
file:.git/config           core.bare=false
file:.git/config           core.logallrefupdates=true
file:.git/config           core.symlinks=false
file:.git/config           core.ignorecase=true

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Task-1 (master)
$ cd ..

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code (master)
$ ls
Task-1/  test.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code (master)
$ git clone ggit@github.com:NirmalMJ/Cognizance.git
Cloning into 'Cognizance'...
The authenticity of host 'github.com (13.234.210.38)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDAOzPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
```

```
 MINGW64:/c/Users/Sri Nirmal Kumaar/Documents/Git_Code/Cognizance

$ git clone git@github.com:NirmalMJ/Cognizance.git
Cloning into 'Cognizance'...
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

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

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ git clone https://github.com/NirmalMJ/Cognizance.git
Cloning into 'Cognizance'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 4), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (20/20), 5.07 KiB | 5.07 MiB/s, done.
Resolving deltas: 100% (4/4), done.

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ ls
Cognizance/  Task-1/  test.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ rm -r Task-1/
rm: remove write-protected regular file 'Task-1/.git/objects/23/7b12cfc198e620a583e1372c8a1ccfabad5316'? y
rm: remove write-protected regular file 'Task-1/.git/objects/51/b978b5b4f0e97b998d8b9d65a9ce7e000899cb'? yes
rm: remove write-protected regular file 'Task-1/.git/objects/65/b34ac73093fbcff81d9c0ddd7d466f2e2361d9'? 1
rm: cannot remove 'Task-1/.git/objects/65': Directory not empty
rm: remove write-protected regular file 'Task-1/.git/objects/89/13a6ec711324e9a0028eef5cb523b8346860ad'? 0
rm: cannot remove 'Task-1/.git/objects/89': Directory not empty
rm: remove write-protected regular file 'Task-1/.git/objects/b7/6a8be750dd89c1c0d14bfdc5a512a0beb90e61'? 1
rm: cannot remove 'Task-1/.git/objects/b7': Directory not empty
rm: remove write-protected regular file 'Task-1/.git/objects/bf/e2701dff2af4090ed881d3d08180081f096cf6'?

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ rmdir Task-1/
rmdir: failed to remove 'Task-1/': Directory not empty

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ ls
Cognizance/  test.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ cd Cognizance/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ ls
README.md  Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ mkdir Task-1

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ ls
README.md  Task-1/  Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ |
```

```
MINGW64:/c/Users/Sri Nirmal Kumaar/Documents/Git_Code/Cognizance/Task-1
rm: remove write-protected regular file 'Task-1/.git/objects/bf/e2701dff2af4090ed881d3d08180081f096cf6'?

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ rmdir Task-1/
rmdir: failed to remove 'Task-1/': Directory not empty

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ ls
Cognizance/  test.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code
$ cd Cognizance/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ ls
README.md  Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ mkdir Task-1

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ ls
README.md  Task-1/  Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ open Task-1.txt
bash: open: command not found

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ gio open Task-1.txt
bash: gio: command not found

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ geo open Task-1.txt
bash: geo: command not found

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ nano Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ mv Task-1.txt ./Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ ls
README.md  Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ cd Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main)
$ ls
Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main)
$ nano Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main)
$
```

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

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ git add /
fatal: C:/Program Files/Git/: 'C:/Program Files/Git/' is outside repository at 'C:/Users/Sri Nirmal Kumaar/Documents/Git_Code/Cognizance'

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
        add

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ git add .

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ ls
README.md  Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ ls -a
./   ../   .git/  README.md  Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ git add Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ ls
README.md  Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (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)
        renamed:    Task-1.txt -> Task-1/Task-1.txt


Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (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)
        renamed:    Task-1.txt -> Task-1/Task-1.txt

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:   Task-1/Task-1.txt


Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$
```

```
 MINGW64:/c/Users/Sri Nirmal Kumaar/Documents/Git_Code/Cognizance

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ ls -a
./   ../   .git/   README.md   Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ git add Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ ls
README.md   Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (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)
        renamed:    Task-1.txt -> Task-1/Task-1.txt


Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (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)
        renamed:    Task-1.txt -> Task-1/Task-1.txt

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:   Task-1/Task-1.txt


Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ git add *.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ git add .

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ git commit -m "I have added a new fun line"
[main bb68a84] I have added a new fun line
 1 file changed, 3 insertions(+)
 rename Task-1.txt => Task-1/Task-1.txt (67%)

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ |
```

```
$ ^C

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main|MERGING)
$ gh pr merge pull-request
bash: gh: command not found

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main|MERGING)
$ git pr merge pull-request
git: 'pr' is not a git command. See 'git --help'.

The most similar command is
        p4

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main|MERGING)
$ git p4 merge pull-request
fatal: 'p4' appears to be a git command, but we were not
able to execute it. Maybe git-p4 is broken?

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main|MERGING)
$ git add .

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main|MERGING)
$ git commit -m "This is an updated thing"
[main 4b91286] This is an updated thing

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 839 bytes | 839.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/NirmalMJ/Cognizance.git
   1abb281..4b91286  main -> main

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ ls
Experimentation.txt  README.md  Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance (main)
$ cd Task-1/

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main)
$ ls
Intro  Task-1.txt

Sri Nirmal Kumaar@LAPTOP-BLVU2TFE MINGW64 ~/Documents/Git_Code/Cognizance/Task-1 (main)
$
```