Git and Github

- Difference between Git and GitHub

➔ Git is a version control tool (software) to track the changes in the source code.

➔ GitHub is a web-based cloud service to host your source code(Git repositories). It is a centralized system.

➔ Git doesn't require GitHub but GitHub requires Git.

1.) The history of a project is stored in git repositories, and github is an online platform which allows us to host these repositories which can be viewed by others. And is initialised using the command: git init

2.)

"ls -a" ➔ shows hidden folders(like .git etc.)

(ls: list command gives the list of all items present in a directory)

"ls .git" ➔ shows the items inside the git repository

3.) To know the status of the git repository, the command is: git status

```
HP@LAPTOP-MJEV37CF MINGW64 /c/DSA (Rohit)
$ git status
On branch Rohit

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .vscode/
        a.exe
        cgpa.cpp
        helloworld/
        p.cpp
        p.exe
        sum_of_numbers.cpp
        test.cpp
        types_of_triangles.cpp

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

4.) The untracked files(remember the wedding example) can be brought onto the stage using the command: git add filename. Eg: git add test.cpp

```
HP@LAPTOP-MJEV37CF MINGW64 /c/DSA (Rohit)
$ git add test.cpp

HP@LAPTOP-MJEV37CF MINGW64 /c/DSA (Rohit)
$ git status
On branch Rohit

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   test.cpp

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .vscode/
        a.exe
        cgpa.cpp
        helloworld/
        p.cpp
        p.exe
        sum_of_numbers.cpp
        types_of_triangles.cpp
```

5.) After bringing the required files onto the stage, we need to click a photo of the file(wedding example) or committing a file using the command: git commit -m "your message".
(-m indicates a message to be printed).

```
HP@LAPTOP-MJEV37CF MINGW64 /c/DSA (Rohit)
$ git commit -m "all files were added"
[Rohit (root-commit) c263d4a] all files were added
 16 files changed, 457 insertions(+)
 create mode 100644 .vscode/c_cpp_properties.json
 create mode 100644 .vscode/launch.json
 create mode 100644 .vscode/tasks.json
 create mode 100644 a.exe
 create mode 100644 cgpa.cpp
 create mode 100644 helloworld/.vscode/c_cpp_properties.json
 create mode 100644 helloworld/.vscode/launch.json
 create mode 100644 helloworld/.vscode/tasks.json
 create mode 100644 helloworld/a.exe
 create mode 100644 helloworld/test.cpp
 create mode 100644 helloworld/test.exe
 create mode 100644 p.cpp
 create mode 100644 p.exe
 create mode 100644 sum_of_numbers.cpp
 create mode 100644 test.cpp
 create mode 100644 types_of_triangles.cpp

HP@LAPTOP-MJEV37CF MINGW64 /c/DSA (Rohit)
$ git status
On branch Rohit
nothing to commit, working tree clean
```

6.) To unstage a file, the command used is: git restore –staged filename.
   EG: git restore –staged test.cpp.

```
HP@LAPTOP-MJEV37CF MINGW64 /c/DSA (Rohit)
$ git restore --staged test.cpp

HP@LAPTOP-MJEV37CF MINGW64 /c/DSA (Rohit)
$ git status
On branch Rohit
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:   test.cpp

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

7.) To view the overall history of the project, we use the command: git log.

```
HP@LAPTOP-MJEV37CF MINGW64 /c/DSA (Rohit)
$ git log
commit 868ca8c2a6b344a42cc32919b9d2f25599b63329 (HEAD -> Rohit)
Author: Rohit <rohitkumarvskp18@gmail.com>
Date:   Mon Jun 20 16:15:33 2022 +0530

    hell

commit c263d4aebd7078065c1ea61f7586e76789c27a40
Author: Rohit <rohitkumarvskp18@gmail.com>
Date:   Mon Jun 20 15:56:37 2022 +0530

    all files were added
```

It shows all the commits we have made.