

GIT commands

GIT installation

- Ensure that GIT is installed on your system.

```
> sudo apt update  
> sudo apt install git vim
```

- You need to do few one time settings for the git. Assuming that your git user name is kd1_12345 and email is kd1_12345@gmail.com

```
> git config --global user.name "kd1_12345"  
> git config --global user.email "kd1_12345@gmail.com"  
> git config --global core.editor vim
```

Local Repository

- Open a new terminal. By default you are in your current directory. "pwd" command shows you the current/present working directory.

```
> pwd
```

- Create a directory "java-assign" in your home directory and go in that directory.

```
> mkdir java-assign  
> cd java-assign
```

- Create a new git repository in it.

```
> git init
```

- Use VS code to create a file. You can create any file e.g. Hello.java.

```
class Hello {  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```

- Add in staging area and commit it.

```
> git add Hello.java  
> git commit -m "hello world demo"
```

- Create a new file. You can write a program Addition.java for addition of two numbers.

```
class Addition {  
    public static void main(String[] args) {  
        int a = 22, b = 7, c;  
        c = a + b;  
        System.out.println("Addition: " + c);  
    }  
}
```

- Check git status and add it into staging area.

```
> git status
> git add .
> git status
```

- Commit the changes and check status again (working directory is now clean).

```
> git commit -m "addition assign"
> git status
```

- Modify the program to take input from user.

```
import java.util.Scanner;
class Addition {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a, b, c;
        System.out.print("Enter two numbers: ");
        a = sc.nextInt();
        b = sc.nextInt();
        c = a + b;
        System.out.println("Addition: " + c);
    }
}
```

- Add in staging area and commit the changes.

```
> git status
> git add .
> git status
```

```
> git commit -m "take user input"
> git status
```

- Create a new file (using VS code) in the directory by name ".gitignore". Here we can tell git, which files not to be added into git repository. Add following lines in that file.

```
*.class
*.jar
```

- Add in staging area and commit the changes.

```
> git status
> git add .
> git status
> git commit -m "gitignore added"
> git status
```

- Ensure that JDK is installed on your system.

```
> sudo apt install openjdk-11-jdk
```

- Compile the Java program and execute it.

```
> javac Addition.java

> java Addition
```

- Delete file Hello.java (using rm command). Add changes in staging area and commit the changes. Note in "git status" that Addition.class file is not added into git repository due to gitignore file.

```
> rm Hello.java  
  
> git status  
> git add .  
> git status  
> git commit -m "hello world deleted"  
> git status
```

Personal access token

- Create a new account on github, if not already created.
- Login into github.com and create personal access token.

Remote Repository

- Login into github.com and create a new private repository e.g. java-assign. Do not add gitignore, readme or license while creating repository.
- Push local directory changes into the created remote repository. Assuming that your git repository is 'https://github.com/kd1_12345/java-assign.git'.

```
git remote add origin https://github.com/kd1_12345/java-assign.git  
git branch -M main  
git push -u origin main
```

- Create a new file "Area.java" into local repository and upload into the remote repository.

```
import java.util.Scanner;  
class Addition {  
    public static void main(String[] args) {
```

```
Scanner sc = new Scanner(System.in);
int a, b, c;
System.out.print("Enter length and breadth of rectangle: ");
a = sc.nextInt();
b = sc.nextInt();
c = a * b;
System.out.println("Area: " + c);
}
}
```

```
> git status
> git add .
> git status
> git commit -m "area of rectangle"
> git status
```

```
> git push
```

- Refresh in browser to check if all changes are uploaded on git or not.
- Go to repository settings and invite your friend as collaborator.
- Ask him to accept the invite and clone your git repository on his system. He should use his git username and his token to clone your repository.

```
> git clone https://github.com/kd1_12345/java-assign.git
```

- Now you can create one program to calculate perimeter of the rectangle. Compile and run on your system and add into git repository. Push the changes to the repository.
- Ask your friend to get the latest changes. He can do it using git pull command.

```
> git pull
```

Important Note

- To submit your assignment your own login e.g. kd1_12345 and your personal access token. This process is describe above. You should invite your lab mentor as collaborator.
- You should use sunbeam-kdac user and given access token to download the daily data of the modules. On first day of the module use "git clone" and on subsequent days use "git pull" command.

```
> git clone
```

```
> git pull
```

- For security purpose, you should never give your personal access token to anyone for any reason.