

Name : Zemali Okba

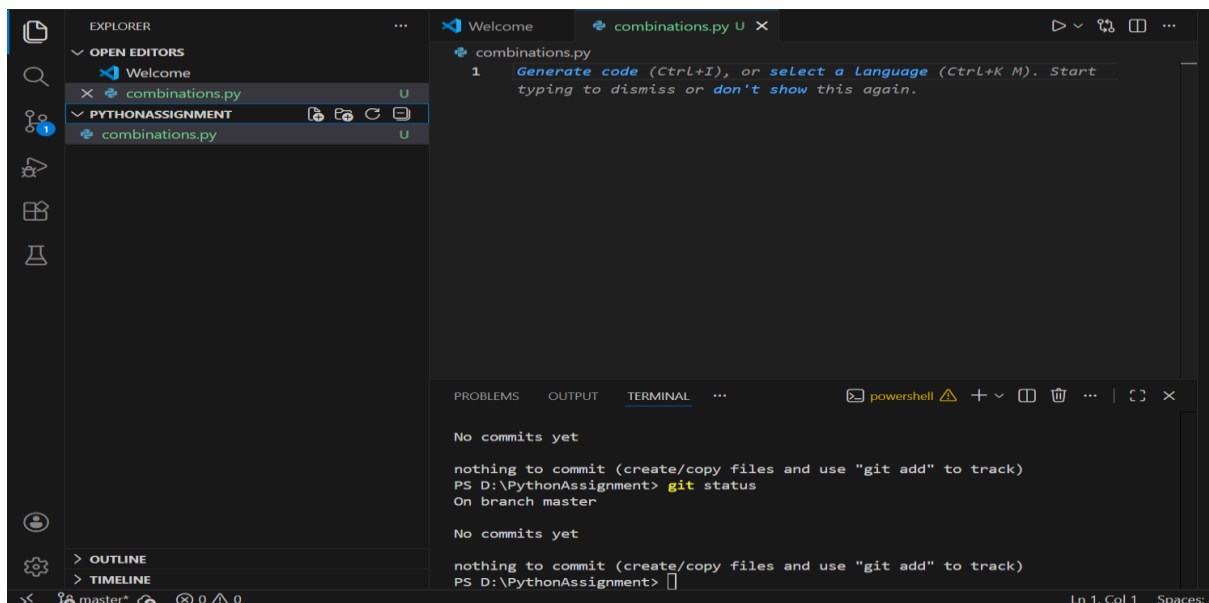
Exercise I: Modularity & VCS

1- Creation of the project directory

2- Creation of GitHub repository

<https://github.com/okba919/Python-Exercicse.git>

3- Git initialization



The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar displays the file structure with a folder named 'PYTHONASSIGNMENT' containing a file 'combinations.py'. The main editor window shows the 'combinations.py' file with a single line of code: `1 Generate code (Ctrl+I), or select a Language (Ctrl+K M). Start typing to dismiss or don't show this again.`. At the bottom, the Terminal panel is open, showing the output of the `git status` command in a PowerShell session. The output indicates that there are no commits yet and that the files are not tracked.

```
No commits yet

nothing to commit (create/copy files and use "git add" to track)
PS D:\PythonAssignment> git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
PS D:\PythonAssignment>
```

4- Create a file termed combinations.py and write a first trivial Python code, Add your file to the staging area, and commit using the message "Trivial comb"

The screenshot shows a Visual Studio Code editor with a file named `combinations.py` open. The code defines a function `comb(n, k)` that calculates combinations using a formula. The terminal shows the following commands and output:

```
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'DALMAC@DESKTOP-SV05VUR.(none)')
PS D:\PythonAssignment> git add combinations.py
fatal: pathspec 'combinations.py' did not match any files
PS D:\PythonAssignment> git add combinations.py
PS D:\PythonAssignment> git commit -m "Trivial comb"
[master (root-commit) 9433bc4] Trivial comb
1 file changed, 6 insertions(+)
create mode 100644 combinations.py
PS D:\PythonAssignment>
```

5- improve the code by the definition of an iterative function fact and add the file to the staging area, and commit using the message "With fact Iterative function"

The screenshot shows the same Visual Studio Code editor with the `combinations.py` file updated. The code now includes an iterative `fact(n)` function and updates the `comb(n, k)` function to use it. The terminal shows the following commands and output:

```
gment/combinations.py
Enter total number of elements n: 5
Enter number of elements to choose k: 2
Number of ways to choose 2 from 5 is: 10
PS D:\PythonAssignment> git add combinations.py
>>
PS D:\PythonAssignment> git commit -m "With fact Iterative function"
>>
[master 6f7bc79] With fact Iterative function
1 file changed, 11 insertions(+), 4 deletions(-)
PS D:\PythonAssignment>
```

6- Enhance code with a recursive factorial function and commit "With fact Recursive function".

The screenshot shows a Visual Studio Code editor with a file named `combinations.py` open. The code defines a recursive function `fact(n)` that calculates the factorial of `n`. It prompts the user to enter the total number of elements `n` and the number of elements to choose `k`. The program then calculates the number of ways to choose `k` elements from `n` elements using the formula $\frac{fact(n)}{(fact(k) * fact(n - k))}$.

```
1 def fact(n):
2     if n == 0 or n == 1:
3         return 1
4     else:
5         return n * fact(n-1)
6
7 n = int(input("Enter total number of elements n: "))
8 k = int(input("Enter number of elements to choose k: "))
9
10 result = fact(n) // (fact(k) * fact(n - k))
11 print(f"Number of ways to choose {k} from {n} is: {result}")
```

The terminal output shows the program being executed with `n=5` and `k=2`, resulting in 10 ways. It also shows the file being added to Git and committed with the message "With fact Recursive function".

```
gnment/combinations.py
Enter total number of elements n: 5
Enter number of elements to choose k: 2
Number of ways to choose 2 from 5 is: 10
PS D:\PythonAssignment> git add combinations.py
>>
PS D:\PythonAssignment> git commit -m "With fact Recursive function"
>>
[master 64e142c] With fact Recursive function
1 file changed, 6 insertions(+), 8 deletions(-)
PS D:\PythonAssignment>
```

7- Modify code to use `math.comb` and commit the message "Using the math library"

The screenshot shows the same Visual Studio Code editor with the modified `combinations.py` file. The code now imports the `math` module and uses `math.comb(n, k)` to calculate the number of combinations.

```
1 import math
2
3 n = int(input("Enter total number of elements n: "))
4 k = int(input("Enter number of elements to choose k: "))
5
6 result = math.comb(n, k)
7 print(f"Number of ways to choose {k} from {n} is: {result}")
```

The terminal output shows the program being executed with the same inputs, resulting in 10 ways. It also shows the file being added to Git and committed with the message "Using the math library".

```
gnment/combinations.py
Enter total number of elements n: 5
Enter number of elements to choose k: 2
Number of ways to choose 2 from 5 is: 10
PS D:\PythonAssignment> git add combinations.py
>>
PS D:\PythonAssignment> git commit -m "Using the math library"
>>
[master e723c25] Using the math library
1 file changed, 2 insertions(+), 6 deletions(-)
PS D:\PythonAssignment>
```

8- Push the solution to GitHub

The screenshot shows a VS Code editor with a file named `combinations.py` open. The file contains the following Python code:

```

1  import math
2
3  n = int(input("Enter total number of elements n: "))
4  k = int(input("Enter number of elements to choose k: "))
5
6  result = math.comb(n, k)
7  print(f"Number of ways to choose {k} from {n} is: {result}")

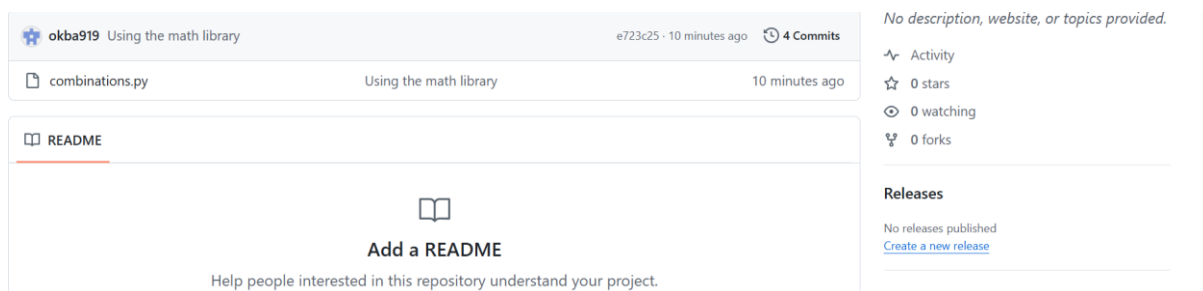
```

The terminal window at the bottom shows the following commands and output:

```

PS D:\PythonAssignment> git remote add origin https://github.com/okba919/Python-Exercise.git
PS D:\PythonAssignment> git push -u origin master
>>
info: please complete authentication in your browser...
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.40 KiB | 204.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.

```



Exercise II: OOP – Analysis Only

1- Identify a problem suitable for object-oriented design

A chemical reactor system in a chemical plant can be modeled using object-oriented design. Each reactor is an object with specific properties and behaviors that represent its characteristics and operations.

2- List examples of static properties (attributes) and behaviors (methods) of your objects

Static Properties (Attributes)	Behaviors (Methods / Dynamics)
Volume (liters)	start() – start the reactor
Temperature (°C)	heat() – increase temperature

Static Properties (Attributes)	Behaviors (Methods / Dynamics)
Pressure (bar)	mix() – mix the reactants
Material type	cool() – decrease temperature
Maximum capacity	stop() – stop the reactor

3- Elucidate inheritance and/or polymorphism cases

Inheritance Example:

- General object: Reactor
- Specialized objects inherit from Reactor:
 - BatchReactor (performs batch operations)
 - ContinuousReactor (performs continuous operations)

Polymorphism Example:

- The method start() is implemented for all reactor types.
- Each type behaves differently:
 - BatchReactor starts operation in batches.
 - ContinuousReactor starts operation continuously.