# GIT

* Use git bash (install).
* Use Gitbash here.
* The touch command creates files.
* Initialize using git init command.
* Check status: git status.
* Add all: ‘git add .’.
* Git commit: Run the command git commit -m "Your message about the commit."
* Branching checkout commands etc.
* Code that github gives:

git remote add origin https://github.com/MuProHere/Python-Prep.git

git branch -M main

git push -u origin main

* Push/pull, clone, connect, env/masterclass/subclass.
* Check tutorial again.

A useful guide: [An Intro to Git and GitHub for Beginners (Tutorial) (hubspot.com)](https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners)

# Object Oriented Programming

Review OOP PY FILES, here’s one of the codes:

class game:

def \_\_init\_\_(ouch, a,b):

ouch.level1 = a

ouch.level2 = b

def \_\_add\_\_(gamer1, gamer2):

obj = game(gamer1.level1 + gamer2.level1,gamer1.level2 + gamer2.level2)

return obj

def \_\_str\_\_(a):

return "LEVEL 1 SCORE: " + str(a.level1) + "\nLEVEL 2 SCORE: " + str(a.level2)

gmr1 = game(100,200)

gmr2 = game(400,100)

#combine

ttl = gmr1+gmr2

print(ttl)

# Virtual Environments

* Meant to limit python use to a newly created environment, which can be modified and used without affecting the original installation of python.
* **To create:** use [python.exe (concerned version) directory] -m venv [name of env]
* **To utilize:** execute activate.bat in Scripts folder.
* Then use pip to install libraries.
* **Create a dependency** (readable file: txt etc.), then use pip install -r [dependency name+extension] -> u can specify version of library in the dependency file using == (==1.0.0).

# Exception Handling

* This is the stuff:

try:

except *nothing or exception name - which can be found by executing the program*:

#There can be multiple exceptions

finally:

**-IMPORTANT ->**

except Exception as e:

# Context Managers – “with”

with expression as target\_var:

expression must return an object that implements the **context management protocol** consists of two [special methods](https://realpython.com/python-classes/#special-methods-and-protocols):

1. [.\_\_enter\_\_()](https://docs.python.org/3/library/stdtypes.html#contextmanager.__enter__) is called by the with statement to enter the runtime context.
2. [.\_\_exit\_\_()](https://docs.python.org/3/library/stdtypes.html#contextmanager.__exit__) is called when the execution leaves the with code block.

e.g:

with open(“asd.csv”) as myfile:

myfile.write(“asd”)

myfile = open(“asd.csv”)

with myfile:

myfile.write(“sadas”)