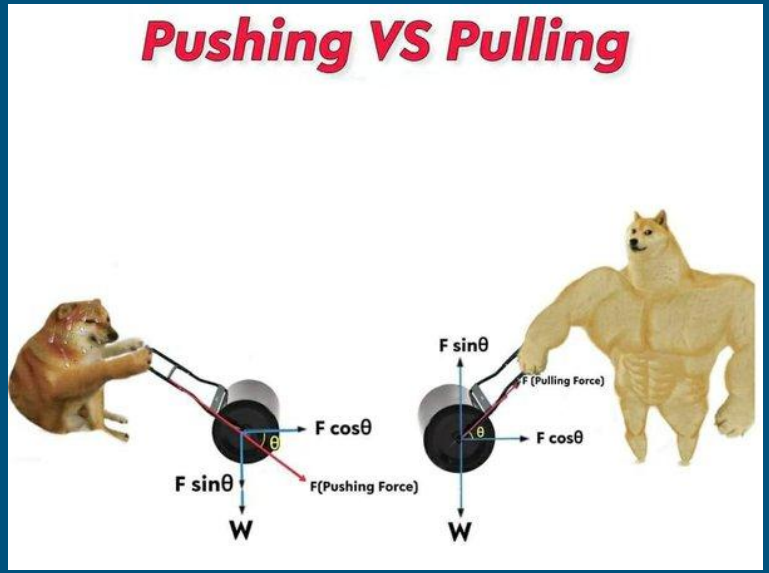
# Git Merge and Classes

Manas Jyoti Das, PhD
Computer Science

# To pull or to push

- When another team member updates the global repo, your local repo gets outdated. To make it updated you need to `pull`
- `git pull` (make sure of the branch you are at before pulling)
- If there is inconsistency between local and global repo's content, you will not be able to push your code. You have to pull first to make it consistent with the global repo. Then only you can push.



# Why we are using git?

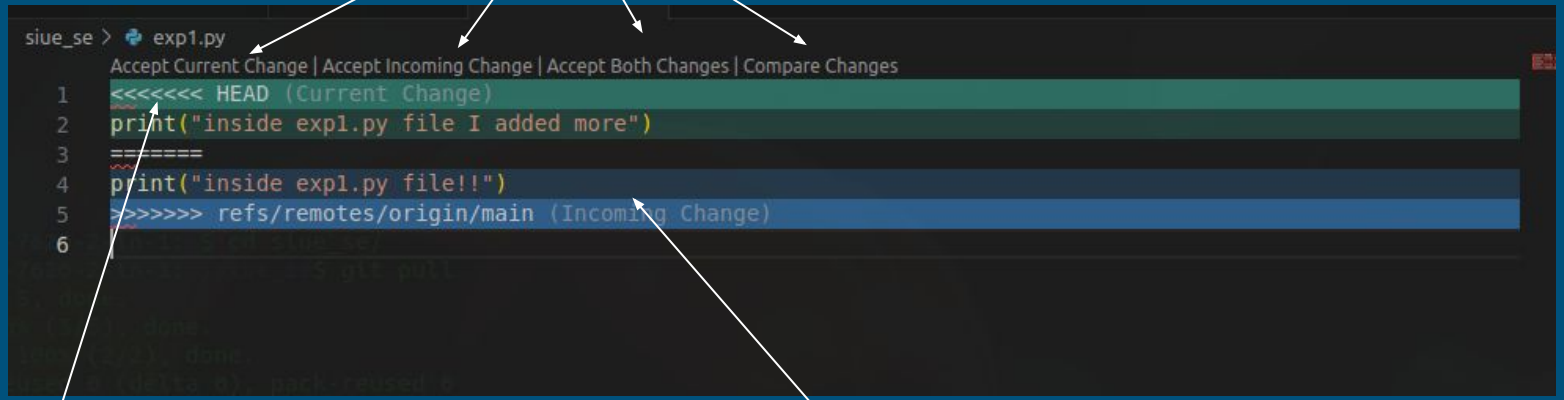
- For collaboration, that means other people will also push code
- When someone else pushed code to the repo what will happen?
- Your local repository is outdated now
- Easy to pull if the changes are in different lines of the same file
- Example: Person-1 change lines:34-39 in file `abc.py` and person-2 changes lines:59-56 (same file different lines)
- `git pull`

# Conflict

- When trying to pull, it may happen that few files might have code changes that conflict with your code changes. Because the changes are on the same line for both the local and global repo
- Example: Person-1 changes lines:23-27 in `abc.py` and lines:5-6 in `xyz.py`. Person-2 changes lines: 25-28 in `abc.py` and lines:9-10 in `xyz.py`
- There will be conflict for file abc.py due to overlap. Line 25, 26, 27.
- `git merge`
- Use vscode to resolve it
- To resolve the issue talk with your collaborator, since resolving and pushing on your own may again break your collaborator's code.

# Resolving conflict in VS code after `git merge`

Vscode options to resolve conflict




Changes done by you

changes done by other user(s)

# Function vs Classes

- What is a function?
  - You can pass some data to it, it will do some task on the data and send you back the result
  - In python you can pass a function to a function
  - Function are mainly focused on doing one task or action
- Classes (OOPs)
  - How you are structuring the information
  - You compose groups of variables into objects
  - E.g. car class: color, price, HP, etc.
  - This object can be part of other object too
  - E.g car can be an object of 4  wheeler vehicles, etc.
  - Basically it represent the state of the application rather than assigning a specific task to it
  - Methods are not functions they usually modify the state of the object

```
def f1(x):
        return x+1

def f2(f,x):
        return f(x)

print(f2(f1,10))
```

# FUNctions

- Action based, usually the output of one function is fed into another function
- E.g: your project. [ just an example do not take it literally]
- One function to call the URLs one by one
- Pass this URL to another function that is going to download the data
- Another function: input will be file name containing all the downloaded info. output will be a clean file with only the article in it. Without the HTML tags and other junk information
- Things are done in sequence, data flow is important.

# FUNctions

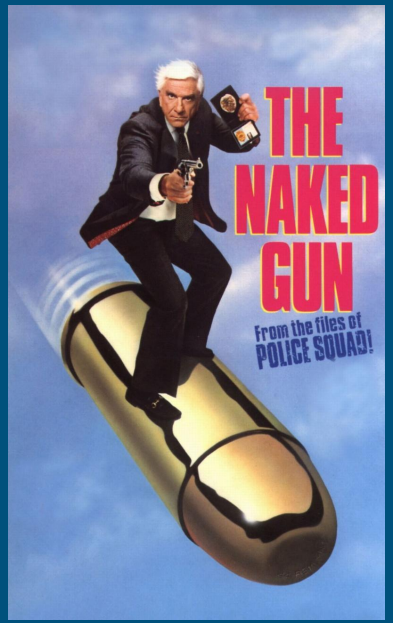
- If you want to convert your project into a OOPs, create a God class
- Make all the functions as method of this class
- Functions are easier to test, since most of them do a single task
- Easy to do Unit testing on it
- Remember: Many a times the function may be modifying the global state
- If you are interested in the structure of the data use OOPs
- Another e.g. class teacher: courses_teaching, Student_list, services.
- Teachers represents a state, we can create objects so we can have many objects, e.g. many teachers
- If done with functions it will be more problematic since teachers are state based (depending on courses, students , etc)

# OOPs

- Used mostly to deal with real world scenarios
- E.g.  Vehicle -> 2/4 wheeler, commercial -> car, truck,..->color, HP, etc
- E.g. Game environment->Characters in game, structures -> health, speed, etc

# Data focused vs Behaviour focused

- Data focused classes are used to structure data into an object (it can have methods)
- Behaviour class means grouping methods that is going to perform similar tasks (it can have data or attributes)
- Usually not strict, can mix them too. It is all about your design choice, there is no `golden bullet` to do it.



# Data focused class

- Emphasis: Primarily on storing and managing data.
- Methods: Relatively fewer or simpler, mainly for accessing, manipulating, or validating the data. They might directly interact with the attributes or simply expose them for external use.
- Example: A "Product" class in an e-commerce application might have attributes like name, price, description, and stock level. Its methods might involve updating information, checking availability, or calculating discounts.

# Method focused class

- Emphasis: Primarily on defining and implementing operations or behaviors.
- Methods: Numerous and complex, representing distinct actions or transformations that the object can perform. They might heavily interact with attributes but focus on the logic and steps involved in the operation.
- Example:
  - A "Calculator" (not again!!) class would have methods for addition, subtraction, multiplication, and division. It might have attributes like current operand or memory, but the focus is on the methods and their functionality.
  - Data Converter: Converts data between different formats (e.g., CSV to JSON, XML to text, etc).
  - Game Character: In a game, might have methods for moving, attacking, interacting with objects

# Thank You

Really!! see you all in the next class