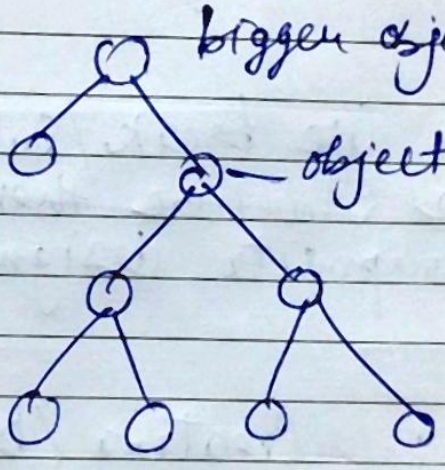
Page No.
Date / /

COMPOSITE DESIGN PATTERN

Use Case
↳ Design File System
↳ Design Calculator.

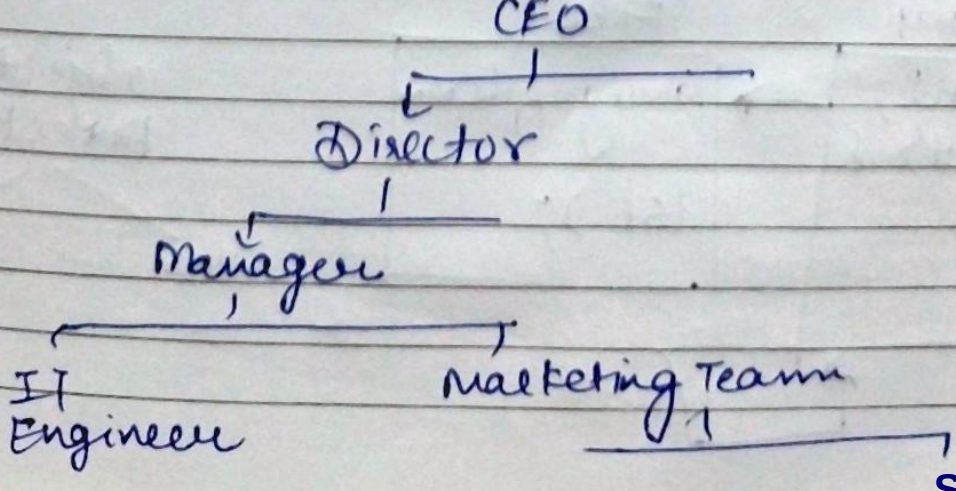
Object inside object

Example:



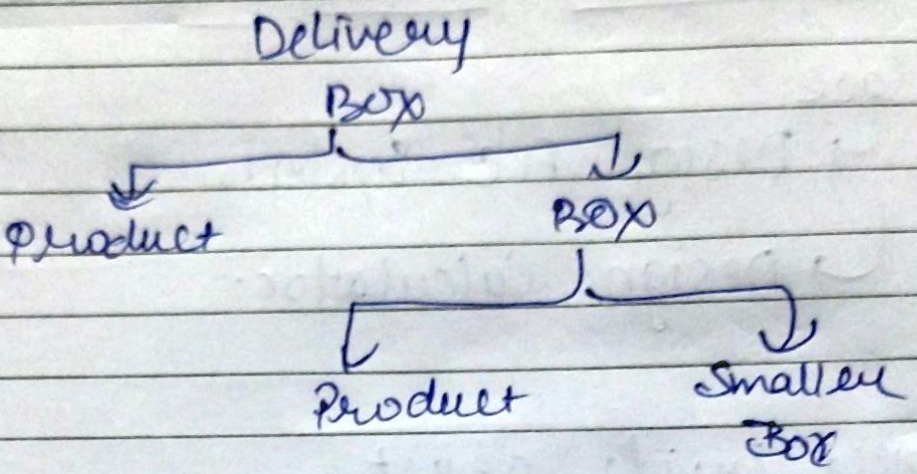
Whichever problems can take tree like structure then we say it's object inside object.

Example: Employee



Page No.
Date / /

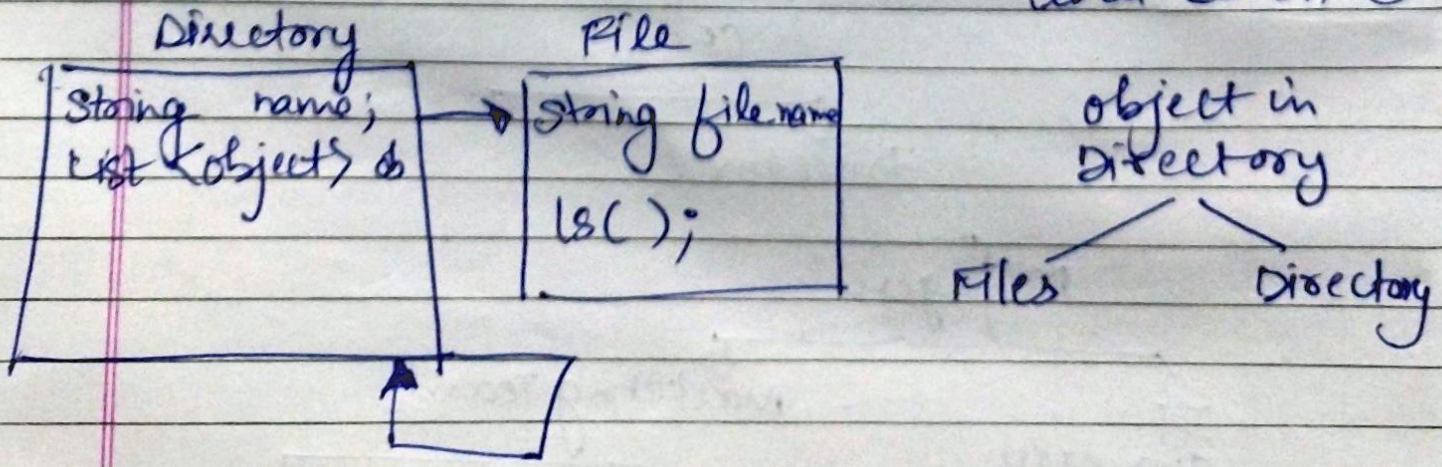
Example:



Whenever we break/solve the problem in Tree like Structure then there we can use composite design Pattern.

Problem: Creating / Designing File System

File System can have
↳ File
↳ Directory.
↳ Files
↳ Sub-directory
and so on



```
Public class Directory
{
    String directory_name;
    List<Object> list;

    Public void Directory(String)
    {
    }

    Public void ls()
    {
        for(object obj : objectlist)
        {
            if(obj instanceof File)
            {
                ((File) obj).ls();
            }
            else if(obj instanceof Directory)
            {
                ((File).obj).ls();
            }
        }
    }
```

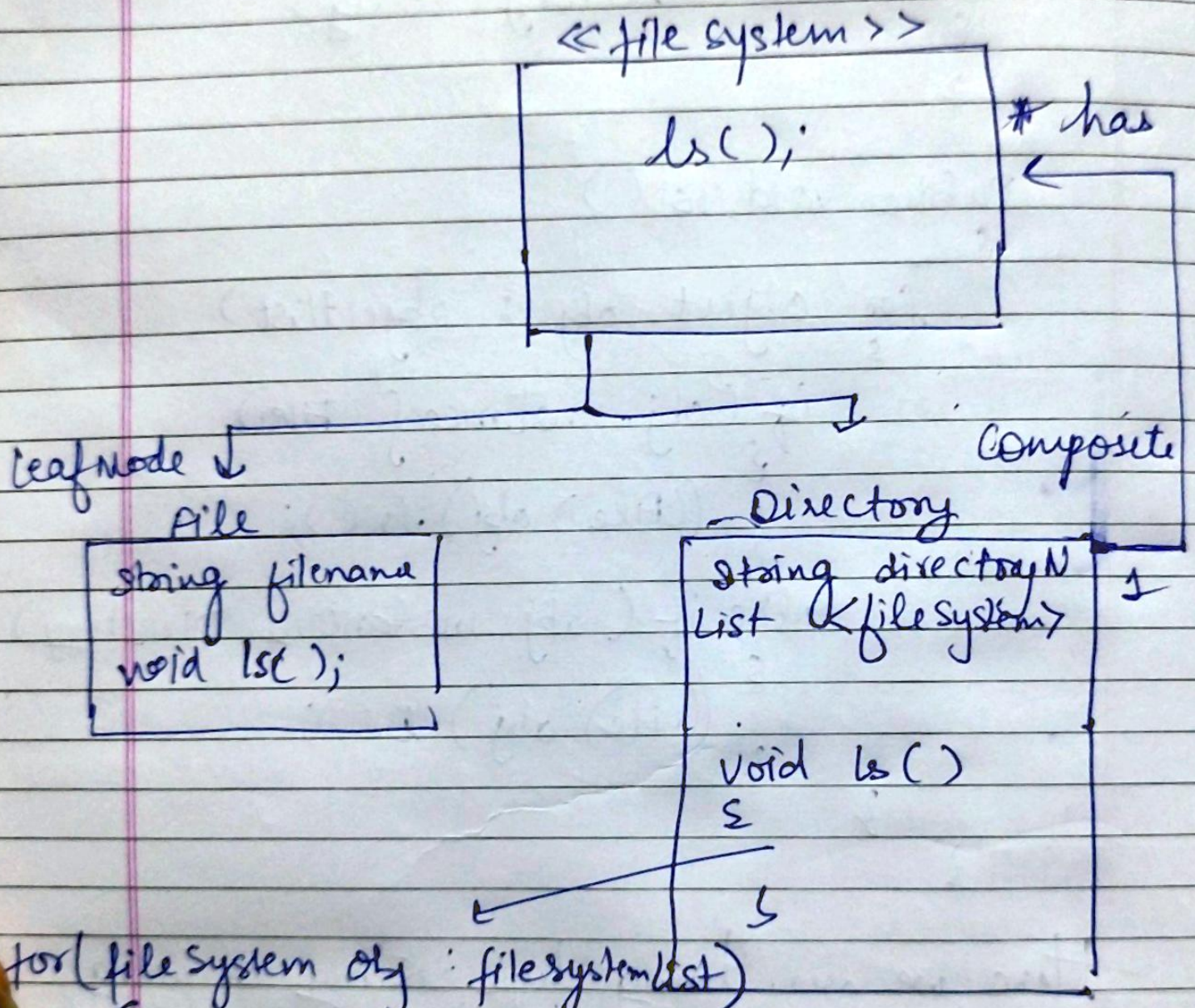
Here we are doing type casting or using if else condition to determine ~~obj~~ ls according to object type.

Page No.
Date / /

Solution:

Composite design pattern

Create an Interface

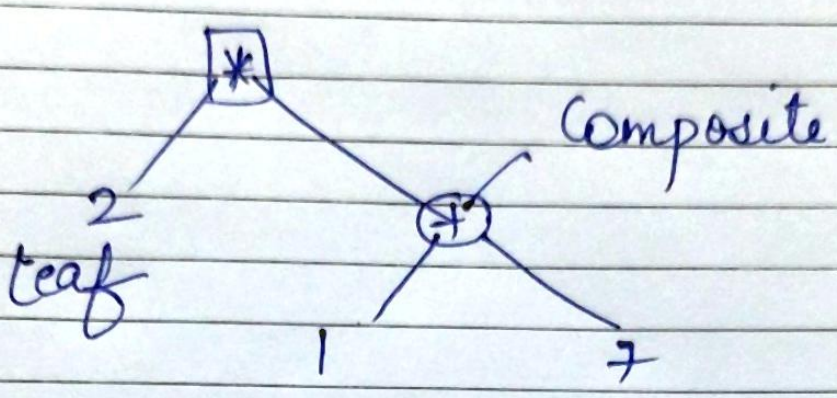


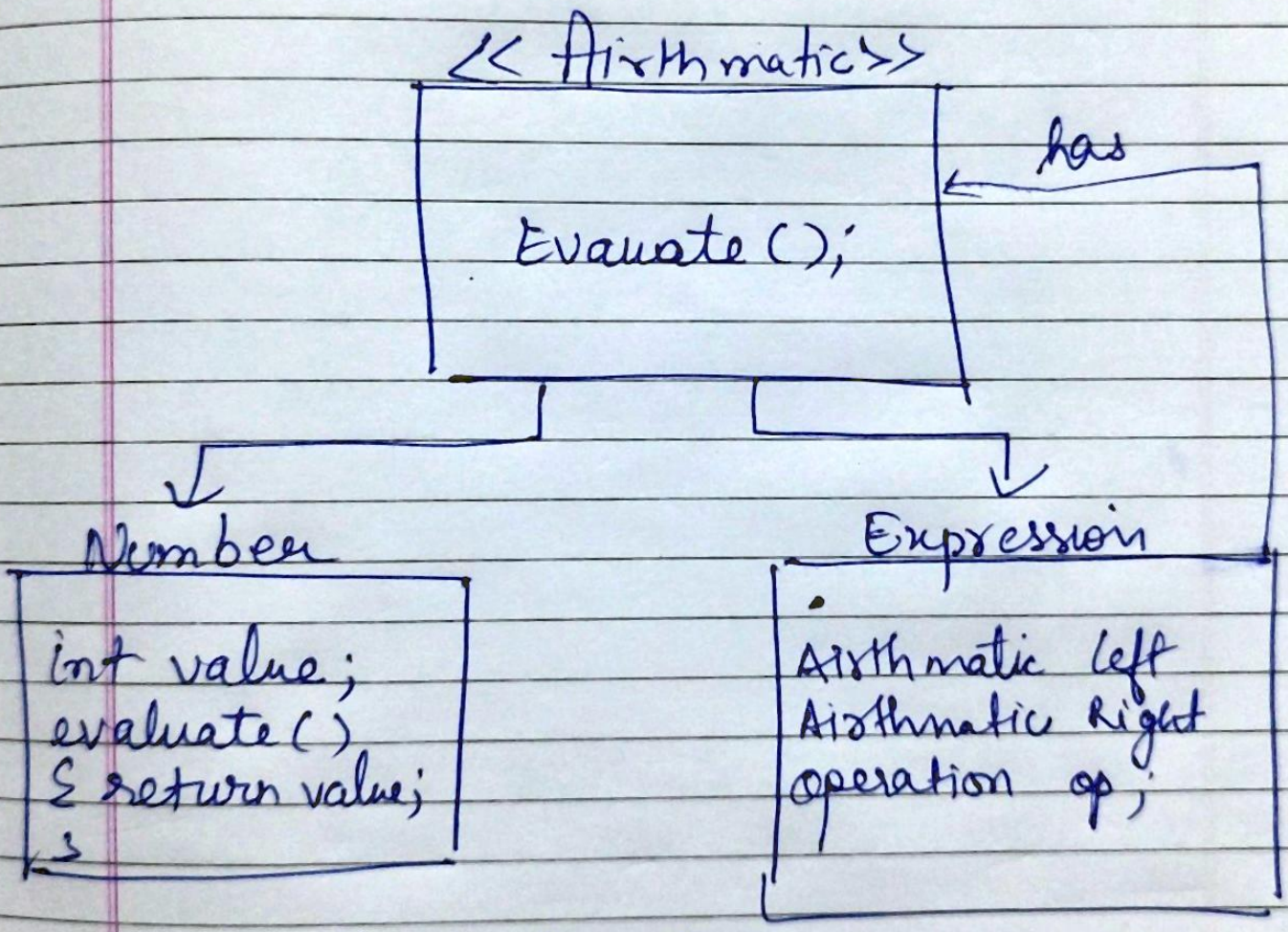
Here we are not using if else to specify object (no need to typecast)

Page No.
Date / /

Example Calculator

2 * (1 + 7)






```
Enum Operation
{
    + ;
    * ;
    / ;
    - ;
};
```