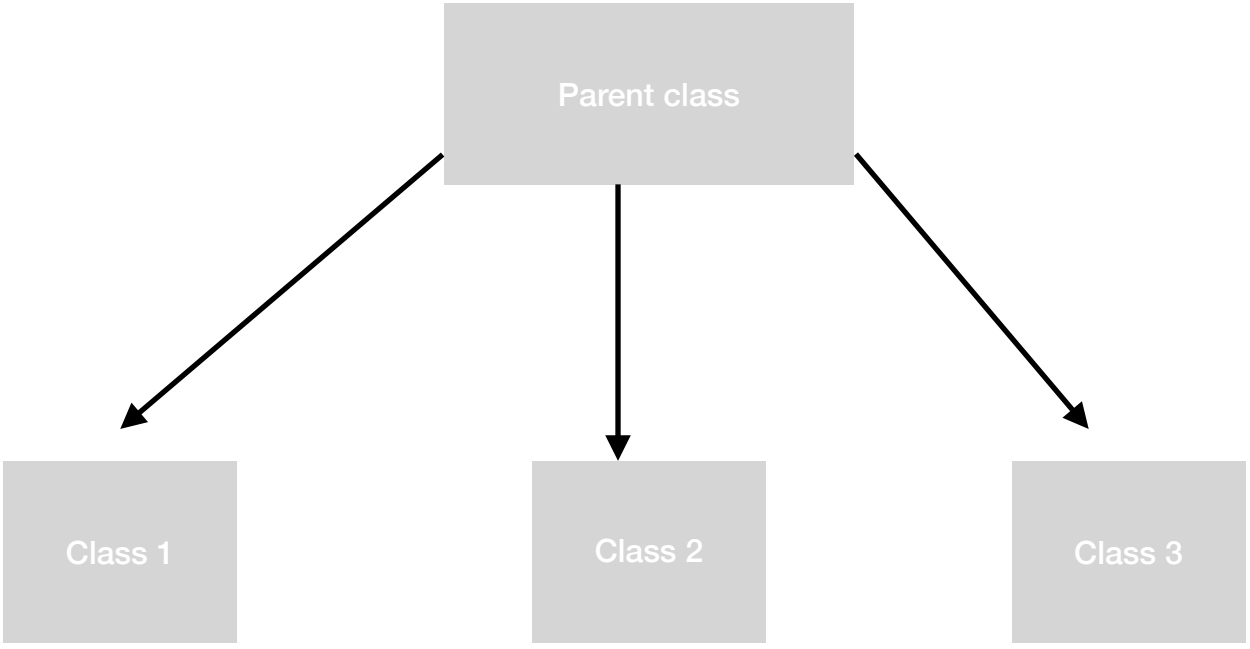
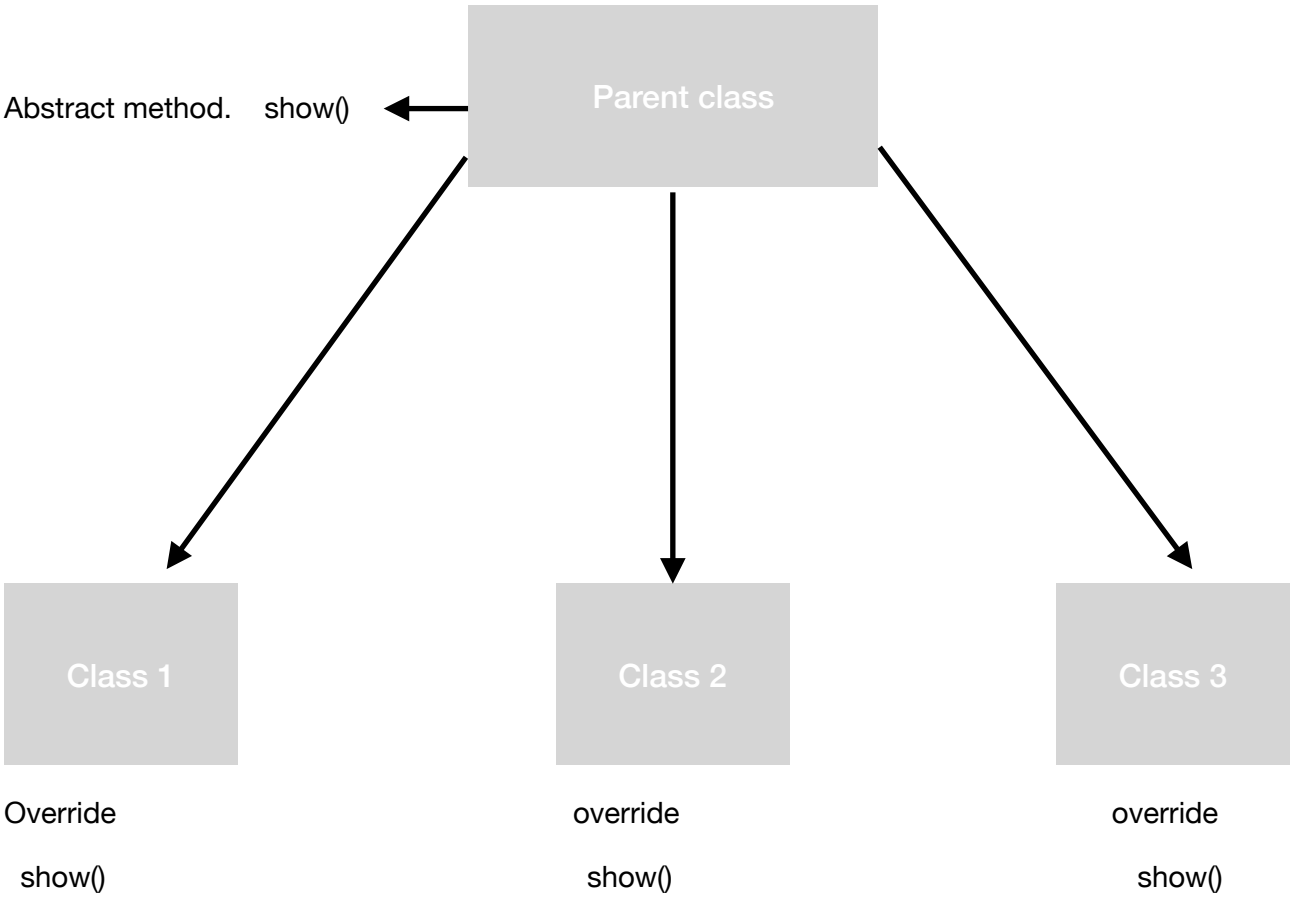
Abstract classes and interface

- Instead of writing same method in each class and every class we can write a parent class and can inherit from parent class
- So, if we have any common code we can write it in a parent class.
- If we write a parent class then all these three class can inherit that from parent class
- As parent class is not directly used so, we can make it abstract
- Abstract means which cannot be instantiated.



- The whole and sole purpose of this class is to share the content with it's child classes this is one reason to write abstract classes
- Abstract classes cannot be instantiated but when we write the child classes for this abstract classes then we can create the instance of these child classes.
- But there is one more reason to write abstract classes.



- Classes must have some set of methods. One or more classes.
- Which is mandatory.
- We can force that classes to inherit them from parent class.

- If the body is not there then this is abstract method

- These classes must override the method in parent class so then that parent class is called as abstract class

Abstract: abstract class may have some abstract method and some concrete method

- Concrete method for sharing the code and abstract method for forcing the class to override method

Interface: interface means abstract class only but which is having only abstract class. Just for forcing child classes to override the method that is why we call it as an interface.

- Abstract classes interfaces are not directly supported by interpreter of python but it is provided as a module.

```python
from abc import ABC, abstractmethod

class Parent(ABC):#abstract class #any class inheriting abc becomes abstract class

    @abstractmethod
    def show(self):
        pass

    def display(self):#concrete method #a method which have body is called concrete
        print('Parent Display')


class Child(Parent):
    def show(self):#must override show as show method is declared as abstract in parent
        print('Show from Child')

c = Child
c.show()#calling method
c.display()
```

- Both are abstract it is nothing but interface. One as abstract and one as concrete
- Interface is also written in same way but all the methods are abstract.
- It is directly not available in python it is available through a module ABC