Interface

In Python, The interface concept is not explicitly available, like available in other languages e.g. Java.

In Python, an interface is an abstract class which contains only abstract method but not a single concrete method.

Interface

```
from abc import ABC, abstractmethod
class Father(ABC):
    @abstractmethod
    def disp(self):
        pass

    def show(self):
            print(“Concrete Method”)
```

Rules

All methods of an interface is abstract.

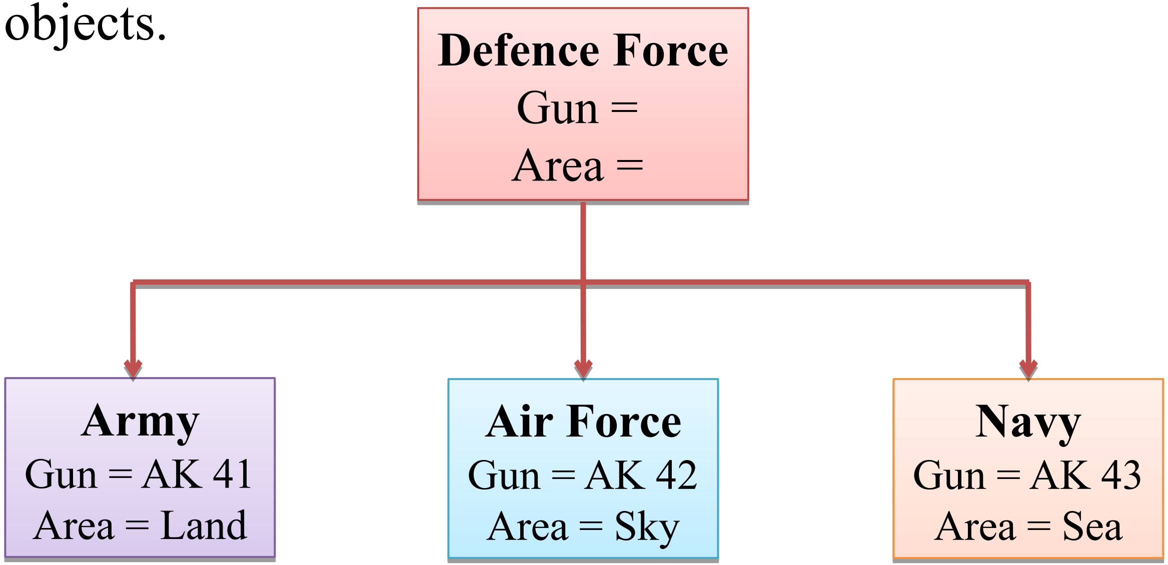
We can not create object of interface.

If a class is implementing an interface it has to define all the methods given in that interface.

If a class does not implement all the methods declared in the interface, the class must be declared abstract.

When use Interface

We use interface when all the features need to be implemented differently for different objects.



Defence Force
Gun =
Area =

Army
Gun = AK 41
Area = Land

Air Force
Gun = AK 42
Area = Sky

Navy
Gun = AK 43
Area = Sea

Abstract Class vs Interface

- An abstract class can have abstract methods as well as concrete methods, but All methods of an interface are abstract.
- We use abstract class when there are some common feature shared by all the objects as they are while we use interface if all the feature need to be implemented differently for different objects.
- Its programmer job to write child class for abstract class while in interface, any third party vendor will take responsibility to write child class.
- Interfaces are slow when compared to abstract class.