

```
In [ ]: # Abstraction:
#       For a abstract class we unable to create an object
#       An abstract class can be considered as a skeleton for other class
# google definition:
# Abstraction is used to hide the internal functionality of the function
# The user only interacts with the basic implementation of the function

# Note: Only you can inherit in the child class
# and you can able to use that

# How to create an abstract class
# The class must inherit from ABC and one method should be any one of
# 1. abstractclassmethod,
# 2. abstractmethod,
# 3. abstractproperty,
# 4. abstractstaticmethod
```

```
In [1]: from abc import (
        ABC,
        abstractclassmethod,
        abstractproperty,
        abstractstaticmethod,
        abstractmethod
    )

    class Parent(ABC):
        @abstractmethod
        def foo(self):
            print("I am in Abstract.foo")

        def bar(self):
            print("I am in Abstract.bar")
```

```
In [2]: obj = Parent()
```

```
-----
-----
TypeError                                 Traceback (most recent call
l last)
<ipython-input-2-8b7647820f0b> in <module>
----> 1 obj = Parent()

TypeError: Can't instantiate abstract class Parent with abstract met
hod foo
```

```
In [5]: # inherit the abstract class in the child class
class Test(Parent):
    def show(self):
        print("I am in Test.show")
```

```
In [6]: obj = Test()
```

```
-----  
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-6-a581470ae006> in <module>  
----> 1 obj = Test()  
  
TypeError: Can't instantiate abstract class Test with abstract method foo
```

```
In [7]: # In the child class you have to override all abstract methods,  
# then only you can create an object for child class  
# 1. abstractclassmethod,  
# 2. abstractmethod,  
# 3. abstractproperty,  
# 4. abstractstaticmethod  
  
class Test(Parent):  
    def foo(self):  
        print("I am in Test.foo")  
    def show(self):  
        print("I am in Test.show")
```

```
In [8]: obj = Test()
```

```
In [9]: obj.foo()
```

```
I am in Test.foo
```

```
In [10]: obj.bar()
```

```
I am in Abstract.bar
```

```
In [11]: obj.show()
```

```
I am in Test.show
```

```
In [ ]:
```