



Eg:

```
1) from abc import *
2) class Fruit:
3)     @abstractmethod
4)     def taste(self):
5)         pass
```

Child classes are responsible to provide implementation for parent class abstract methods.

## Abstract class:

Some times implementation of a class is not complete, such type of partially implementation classes are called abstract classes. Every abstract class in Python should be derived from ABC class which is present in abc module.

Case-1:

```
1) from abc import *
2) class Test:
3)     pass
4)
5) t=Test()
```

In the above code we can create object for Test class b'z it is concrete class and it does not contain any abstract method.

Case-2:

```
1) from abc import *
2) class Test(ABC):
3)     pass
4)
5) t=Test()
```

In the above code we can create object, even it is derived from ABC class, b'z it does not contain any abstract method.

Case-3:

```
1) from abc import *
2) class Test(ABC):
3)     @abstractmethod
4)     def m1(self):
5)         pass
6)
7) t=Test()
```