```
from abc import ABC,abstractmethod
class myclass(ABC):
  @abstractmethod
  def f1(self):
    print('myclass abstract method f1( )')
  def createwindow(self):
    print('window created')

  def settitle(self):
    print('title lebled to window')

  def setMMC(self):
    print('tagged Minimize, Maximize and close buttons')

m1 = myclass()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-ec813b425060> in <cell line: 1>()
----> 1 m1 = myclass()

TypeError: Can't instantiate abstract class myclass with abstract method f1
```

SEARCH STACK OVERFLOW

```
class mywindow(ABC):
  @abstractmethod
  def f1(self):
    pass

  def createwindow(self):
    print('window created')

  def settitle(self):
    print('title lebled to window')

  def setMMC(self):
    print('tagged Minimize, Maximize and close buttons')

class Notepad(mywindow):
  def f1(self):
    print('Notepad is not an abstract class')

n1 = Notepad()
```

n1.createwindow()

```
window created
```

n1.settitle()

```
title lebled to window
```

n1.setMMC()

```
tagged Minimize, Maximize and close buttons
```

dir(mywindow)

```
['__abstractmethods__',
 '__class__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__slots__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 '_abc_impl',
 'createwindow',
 'f1',
 'setMMC',
 'settitle']
```

✓ 0s completed at 2:00 PM