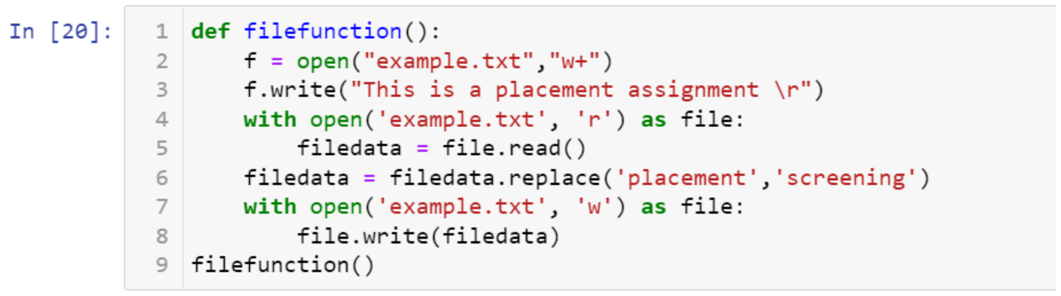
Python Screening Assignment

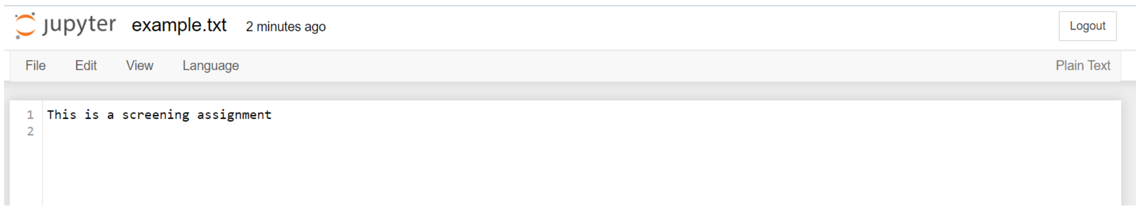
1. Create a function in python to read the text file and replace specific content of the file.

File name	example.txt
Origin file content	This is a placement assignment
Replace string	Placement should be replaced by screening
Replaced file content	This is a screening assignment

Solution:



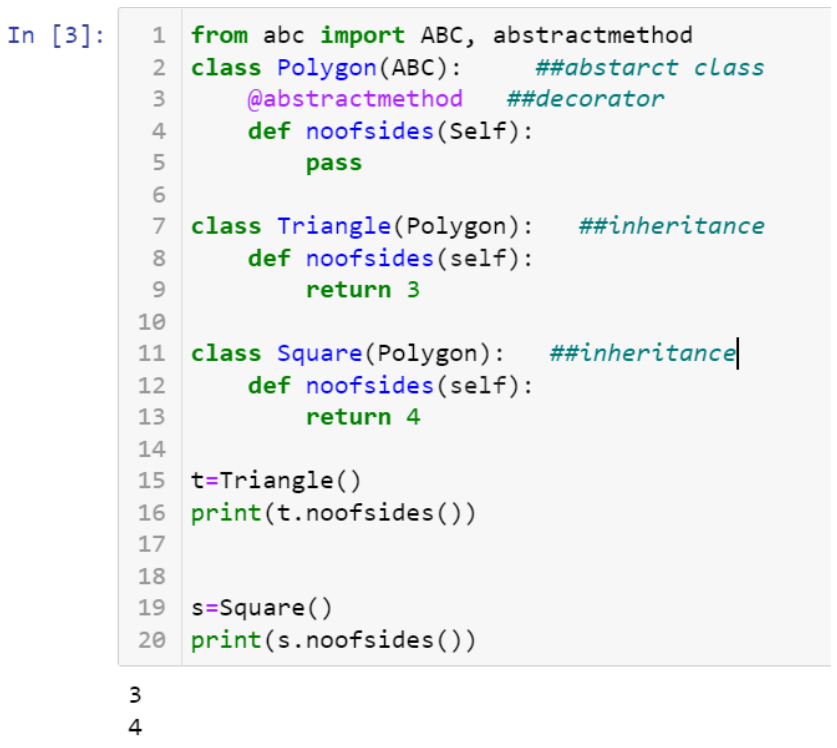
```
In [20]:
def filefunction():
    f = open("example.txt","w+")
    f.write("This is a placement assignment \r")
    with open('example.txt', 'r') as file:
        filedata = file.read()
    filedata = filedata.replace('placement','screening')
    with open('example.txt', 'w') as file:
        file.write(filedata)
filefunction()
```



```
This is a screening assignment
```

2. Demonstrate use of abstract class, multiple inheritance and decorator in python using examples.

Solution:


In [3]:
```python
from abc import ABC, abstractmethod
class Polygon(ABC):      ##abstarct class
    @abstractmethod    ##decorator
    def noofsides(Self):
        pass

class Triangle(Polygon):   ##inheritance
    def noofsides(self):
        return 3

class Square(Polygon):   ##inheritance
    def noofsides(self):
        return 4

t=Triangle()
print(t.noofsides())


s=Square()
print(s.noofsides())
```

```
3
4
```

According to the above example:

- Abstract class: Partially implemented classes that are a child of ABC class. Objects of the abstract class cannot be created, but can be created to the child's class.
- Inheritance: child classes provide implementation to the parent abstract class.
- Decorator: Here the @abstractmethod is a decorator. This is used to ensure that the child class provides implementation to the parent class.