Python Screening Assignment

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

```
import pandas as pd

df = pd.read_csv("example.txt")

df

type(df)

df.columns

df1 = df.rename(columns = {'placement':'screening'})

df1
```

2. Abstract Class, decorators and Multiple inheritance:

   Abstract class: The class that contains abstract method is called abstract class. Abstract method is a method which has only declaration but no definition.

```
class ineuron(ABC):
    @abstractmethod      #Decorator
    def output(self):
        None

    @abstractmethod      #Decorator
    def show(self):
        None

class Demo(ineuron):
    def output(self):
        print("Abstract Class")

    def show(self):
        print("Abstract class is working")

obj=Demo()
obj.output()
obj.show()
```

   Decorators: Decorator allows a used to add new functionality to an existing object without modifying its structure.

```
def main_function(func):

    def sub_function():
```

```
        print("Hi, this is Rakesh Adiga")
        func()
    return sub_function()
@main_function                          #decorator
def new_function():
    print("I study in ineuron")
```

```
Output: Hi, this is Rakesh Adiga
I study in ineuron
```

Multiple inheritance is the type of inheritance in which one class is inherited from two or more different base classes.

Example:

```
class A:
    def m1(self):
        print("A class")
class B(A):
    def m1(self):
        print("B class")
    def m2(self):
        print("B2")
class C(A):
    def m1(self):
        print("C class")
class D(B,C):
    def m1(self):
        print("D class")
        C.m1(self)
        B.m1(self)
        A.m1(self)
d=D()
d.m1()
```

```
Output: D class
C class
B class
A class
```