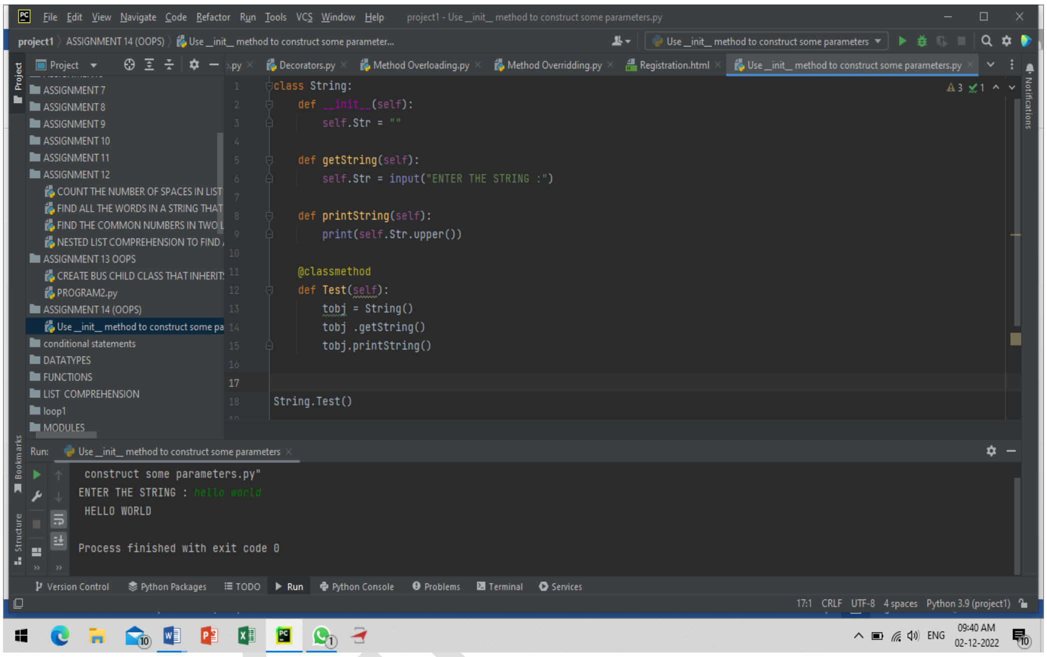
SREEKUTTY C
PYTHON OCTOBER 2022 BATCH 1.30PM – SHIVA

ASSIGNMENT 14

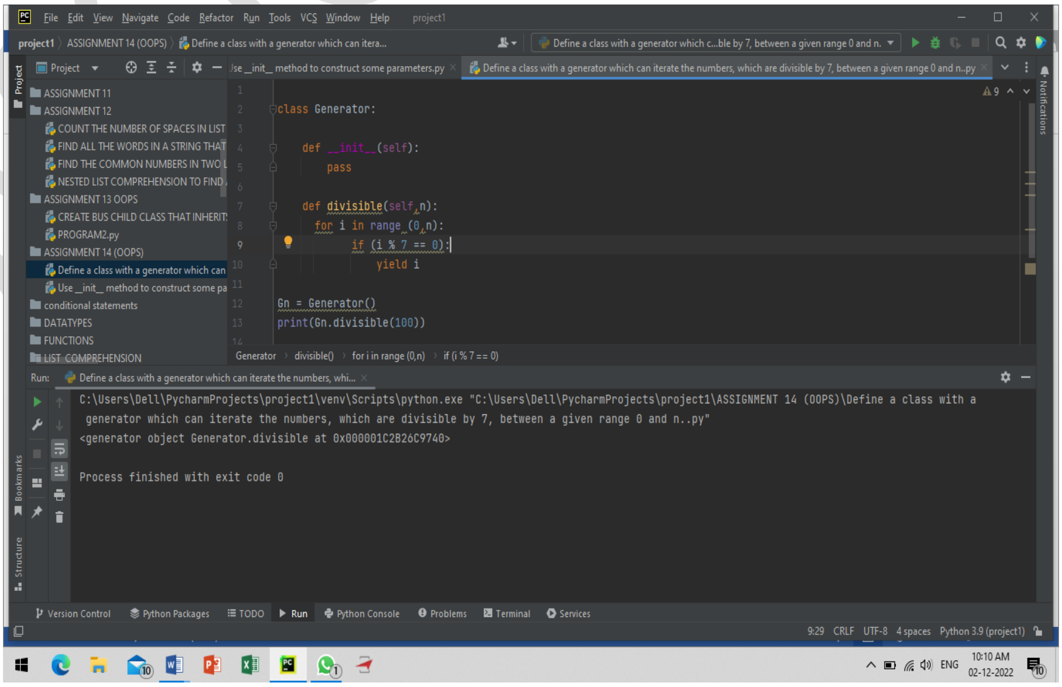
1. Define a class which has at least two methods: getString: to get a string from console input. printString: to print the string in upper case. Also please include simple test function to test the class method .Hints: Use __init__ method to construct some parameters



```
class String:
    def __init__(self):
        self.Str = ""

    def getString(self):
        self.Str = input("ENTER THE STRING :")

    def printString(self):
        print(self.Str.upper())

    @classmethod
    def Test(self):
        tobj = String()
        tobj .getString()
        tobj.printString()


String.Test()
```

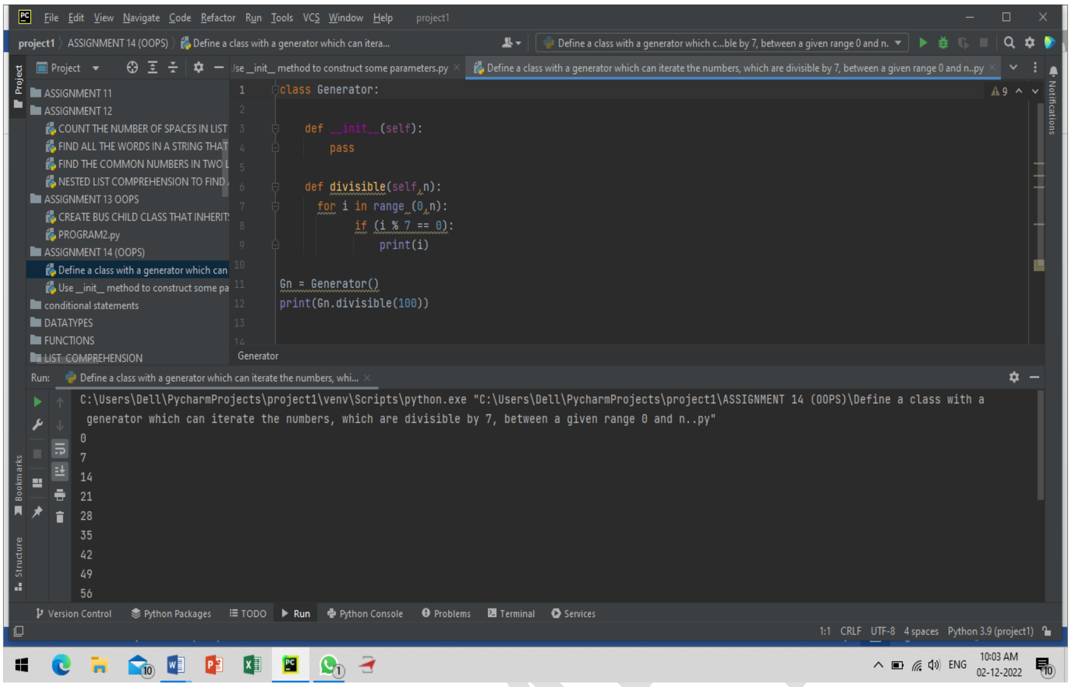
```
 construct some parameters.py"
ENTER THE STRING : hello world
 HELLO WORLD

Process finished with exit code 0
```

2. Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given ranges 0 and n. Hints:_Consider using yield



```

class Generator:

    def __init__(self):
        pass

    def divisible(self,n):
      for i in range (0,n):
            if (i % 7 == 0):
                yield i

Gn = Generator()
print(Gn.divisible(100))
```

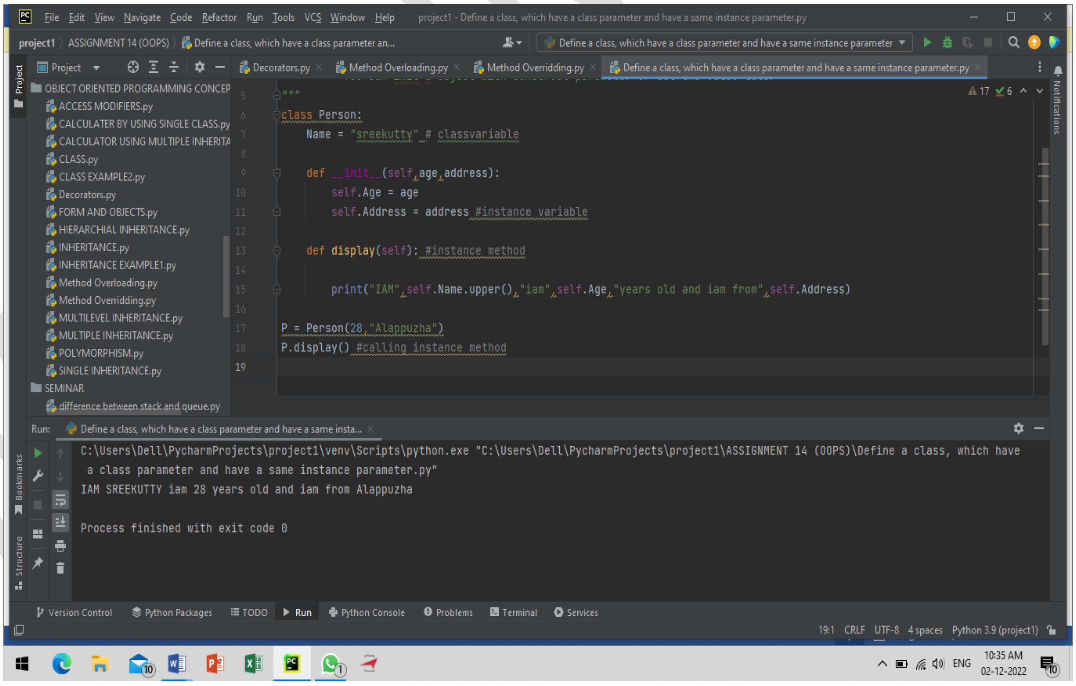
```
C:\Users\Dell\PycharmProjects\project1\venv\Scripts\python.exe "C:\Users\Dell\PycharmProjects\project1\ASSIGNMENT 14 (OOPS)\Define a class with a
 generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n..py"
<generator object Generator.divisible at 0x000001C2B26C9740>

Process finished with exit code 0
```



```
class Generator:

    def __init__(self):
        pass

    def divisible(self,n):
        for i in range (0,n):
            if (i % 7 == 0):
                print(i)

Gn = Generator()
print(Gn.divisible(100))
```

```
C:\Users\Dell\PycharmProjects\project1\venv\Scripts\python.exe "C:\Users\Dell\PycharmProjects\project1\ASSIGNMENT 14 (OOPS)\Define a class with a
 generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n..py"
0
7
14
21
28
35
42
49
56
```

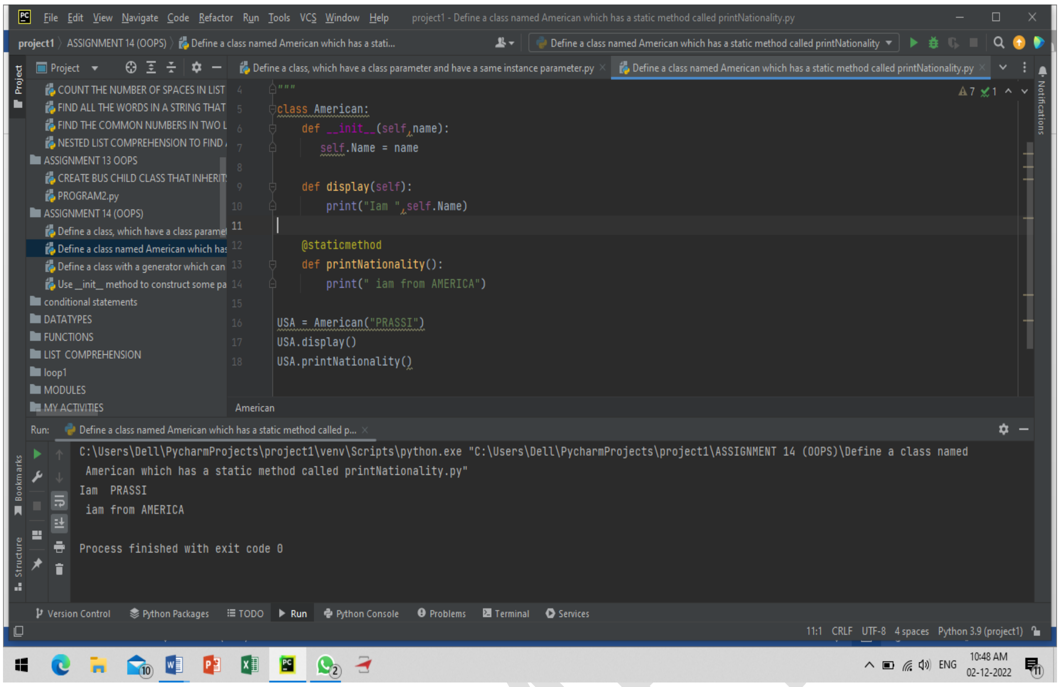
3. Define a class, which have a class parameter and have a same instance parameter. Hints: Define a instance parameter, need add it in __init__ method. You can init a object with construct parameter or set the value later.



```
"""
class Person:
    Name = "sreekutty" # classvariable

    def __init__(self,age,address):
        self.Age = age
        self.Address = address #instance variable

    def display(self): #instance method

        print("IAM",self.Name.upper(),"iam",self.Age,"years old and iam from",self.Address)

P = Person(28,"Alappuzha")
P.display() #calling instance method
```

```
C:\Users\Dell\PycharmProjects\project1\venv\Scripts\python.exe "C:\Users\Dell\PycharmProjects\project1\ASSIGNMENT 14 (OOPS)\Define a class, which have
 a class parameter and have a same instance parameter.py"
IAM SREEKUTTY iam 28 years old and iam from Alappuzha

Process finished with exit code 0
```

4. Define a class named American which has a static method called printNationality. Hints: Use @staticmethod decorator to define class static method.

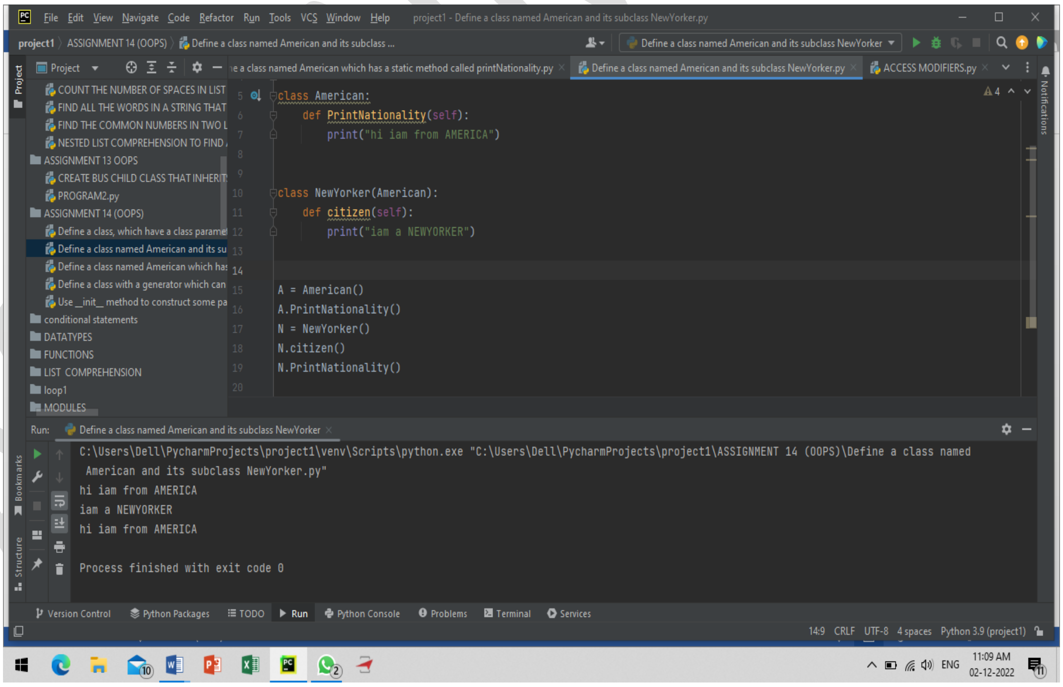


project1 - Define a class named American which has a static method called printNationality.py

```
"""
class American:
    def __init__(self,name):
        self.Name = name

    def display(self):
        print("Iam ",self.Name)

    @staticmethod
    def printNationality():
        print(" iam from AMERICA")

USA = American("PRASSI")
USA.display()
USA.printNationality()
```

```
C:\Users\Dell\PycharmProjects\project1\venv\Scripts\python.exe "C:\Users\Dell\PycharmProjects\project1\ASSIGNMENT 14 (OOPS)\Define a class named
 American which has a static method called printNationality.py"
Iam  PRASSI
 iam from AMERICA

Process finished with exit code 0
```

5. Define a class named American and its subclass NewYorker. Hints: Use class Subclass(ParentClass) to define a subclass

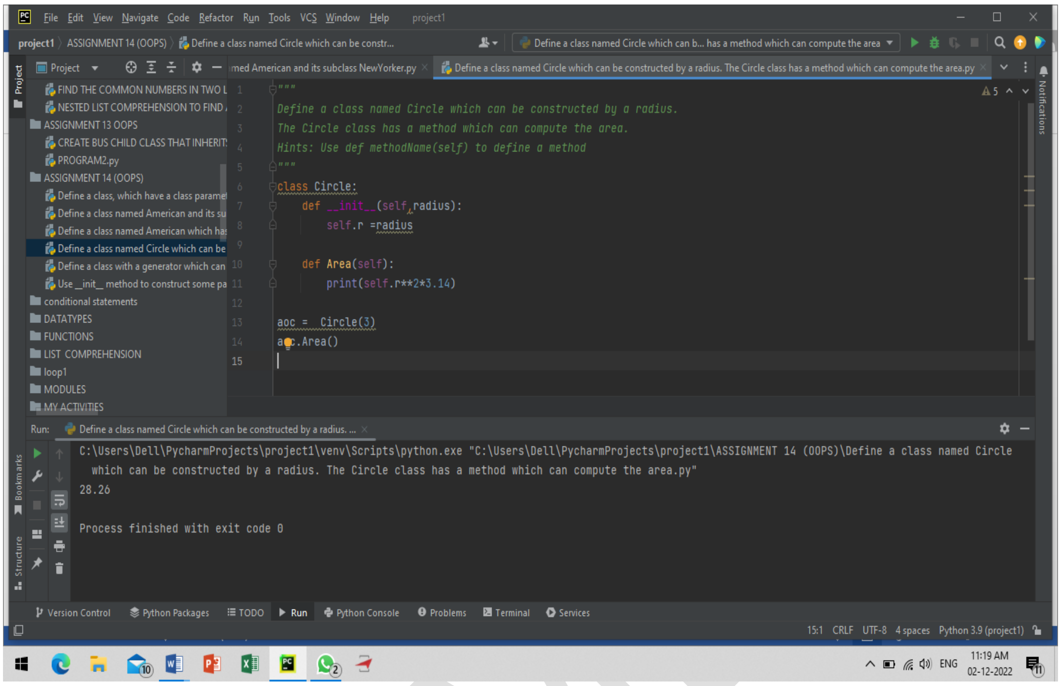


project1 - Define a class named American and its subclass NewYorker.py

```
class American:
    def PrintNationality(self):
        print("hi iam from AMERICA")


class NewYorker(American):
    def citizen(self):
        print("iam a NEWYORKER")


A = American()
A.PrintNationality()
N = NewYorker()
N.citizen()
N.PrintNationality()
```

```
C:\Users\Dell\PycharmProjects\project1\venv\Scripts\python.exe "C:\Users\Dell\PycharmProjects\project1\ASSIGNMENT 14 (OOPS)\Define a class named
 American and its subclass NewYorker.py"
hi iam from AMERICA
iam a NEWYORKER
hi iam from AMERICA

Process finished with exit code 0
```

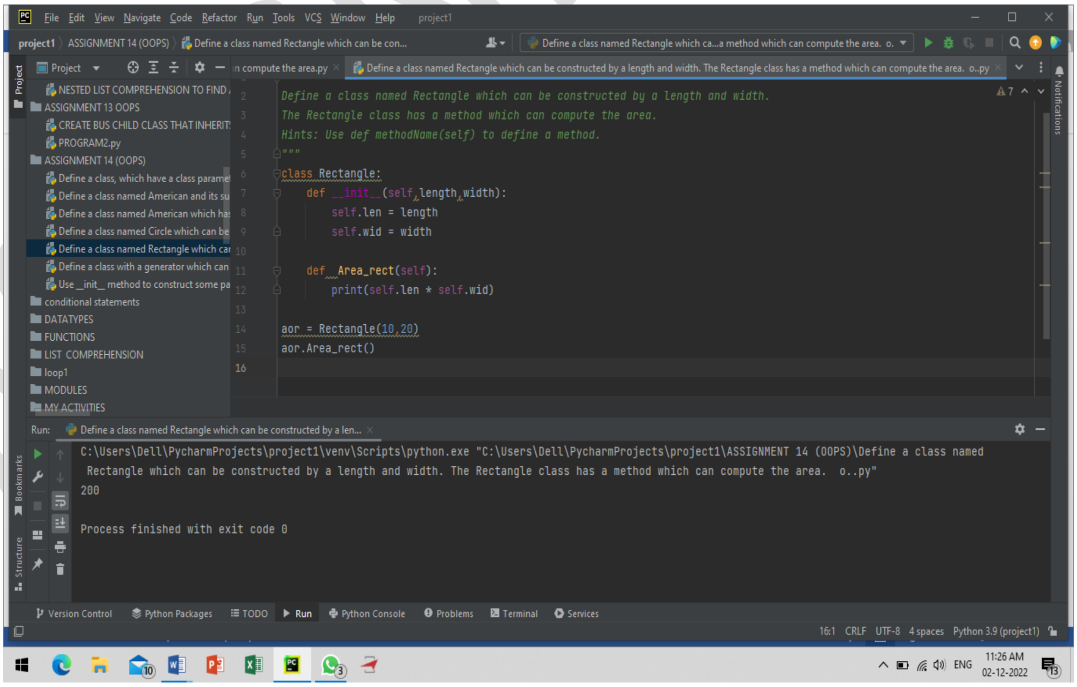
6. Define a class named Circle which can be constructed by a radius. The Circle class has a method which can compute the area. [Hints:](#) Use def methodName(self) to define a method.



```python
"""
Define a class named Circle which can be constructed by a radius.
The Circle class has a method which can compute the area.
Hints: Use def methodName(self) to define a method
"""
class Circle:
    def __init__(self,radius):
        self.r =radius

    def Area(self):
        print(self.r**2*3.14)

aoc =  Circle(3)
aoc.Area()
```

```
C:\Users\Dell\PycharmProjects\project1\venv\Scripts\python.exe "C:\Users\Dell\PycharmProjects\project1\ASSIGNMENT 14 (OOPS)\Define a class named Circle
 which can be constructed by a radius. The Circle class has a method which can compute the area.py"
28.26

Process finished with exit code 0
```

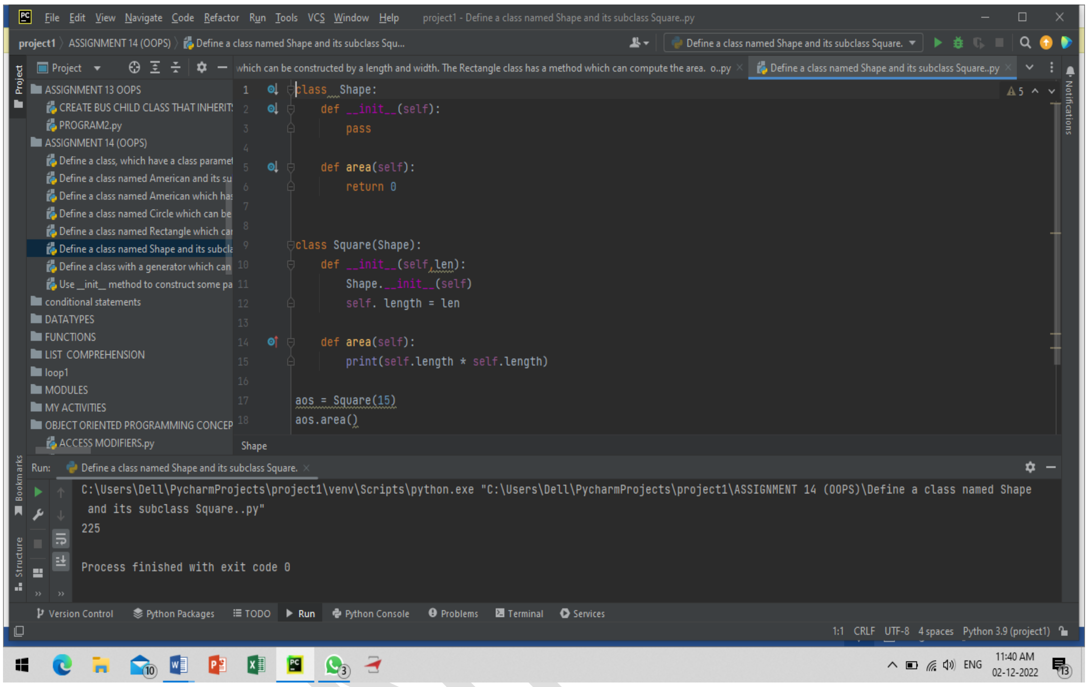
7. Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area. [Hints](#): Use def methodName(self) to define a method.



```python
Define a class named Rectangle which can be constructed by a length and width.
The Rectangle class has a method which can compute the area.
Hints: Use def methodName(self) to define a method.
"""
class Rectangle:
    def __init__(self,length,width):
        self.len = length
        self.wid = width

    def  Area_rect(self):
        print(self.len * self.wid)

aor = Rectangle(10,20)
aor.Area_rect()
```

```
C:\Users\Dell\PycharmProjects\project1\venv\Scripts\python.exe "C:\Users\Dell\PycharmProjects\project1\ASSIGNMENT 14 (OOPS)\Define a class named
 Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area.  o..py"
200

Process finished with exit code 0
```

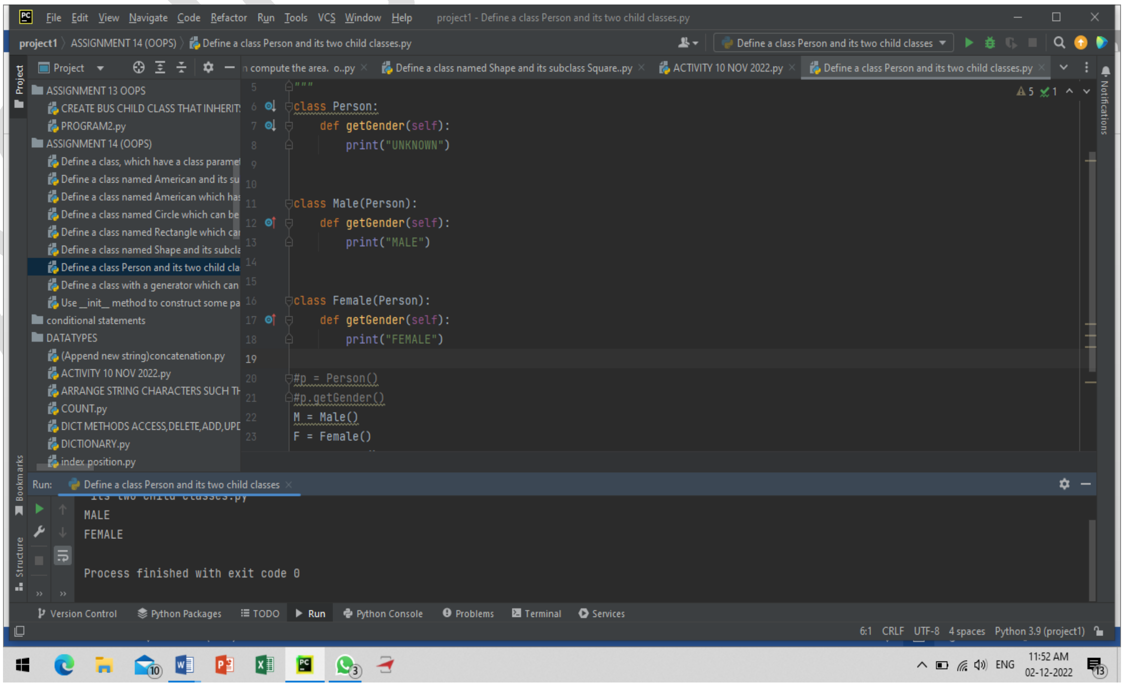
8. Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have a area function which can print the area of the shape where Shape's area is 0 by default. Hints: To override a method in super class, we can define a method with the same name in the super class.



```
class  Shape:
    def __init__(self):
        pass

    def area(self):
        return 0


class Square(Shape):
    def __init__(self,len):
        Shape.__init__(self)
        self. length = len

    def area(self):
        print(self.length * self.length)

aos = Square(15)
aos.area()
```

```
C:\Users\Dell\PycharmProjects\project1\venv\Scripts\python.exe "C:\Users\Dell\PycharmProjects\project1\ASSIGNMENT 14 (OOPS)\Define a class named Shape
 and its subclass Square..py"
225

Process finished with exit code 0
```

9. Define a class Person and its two child classes: Male and Female. All classes have a method "getGender" which can print "Male" for Male class and "Female" for Female class. Hints: Use Subclass(Parentclass) to define a child class.



```
"""
class Person:
    def getGender(self):
        print("UNKNOWN")


class Male(Person):
    def getGender(self):
        print("MALE")


class Female(Person):
    def getGender(self):
        print("FEMALE")

#p = Person()
#p.getGender()
M = Male()
F = Female()
```

```
its two child classes.py
MALE
FEMALE

Process finished with exit code 0
```