**DAY -25 TASK**

**TASK: OBJECT ORIENTED**

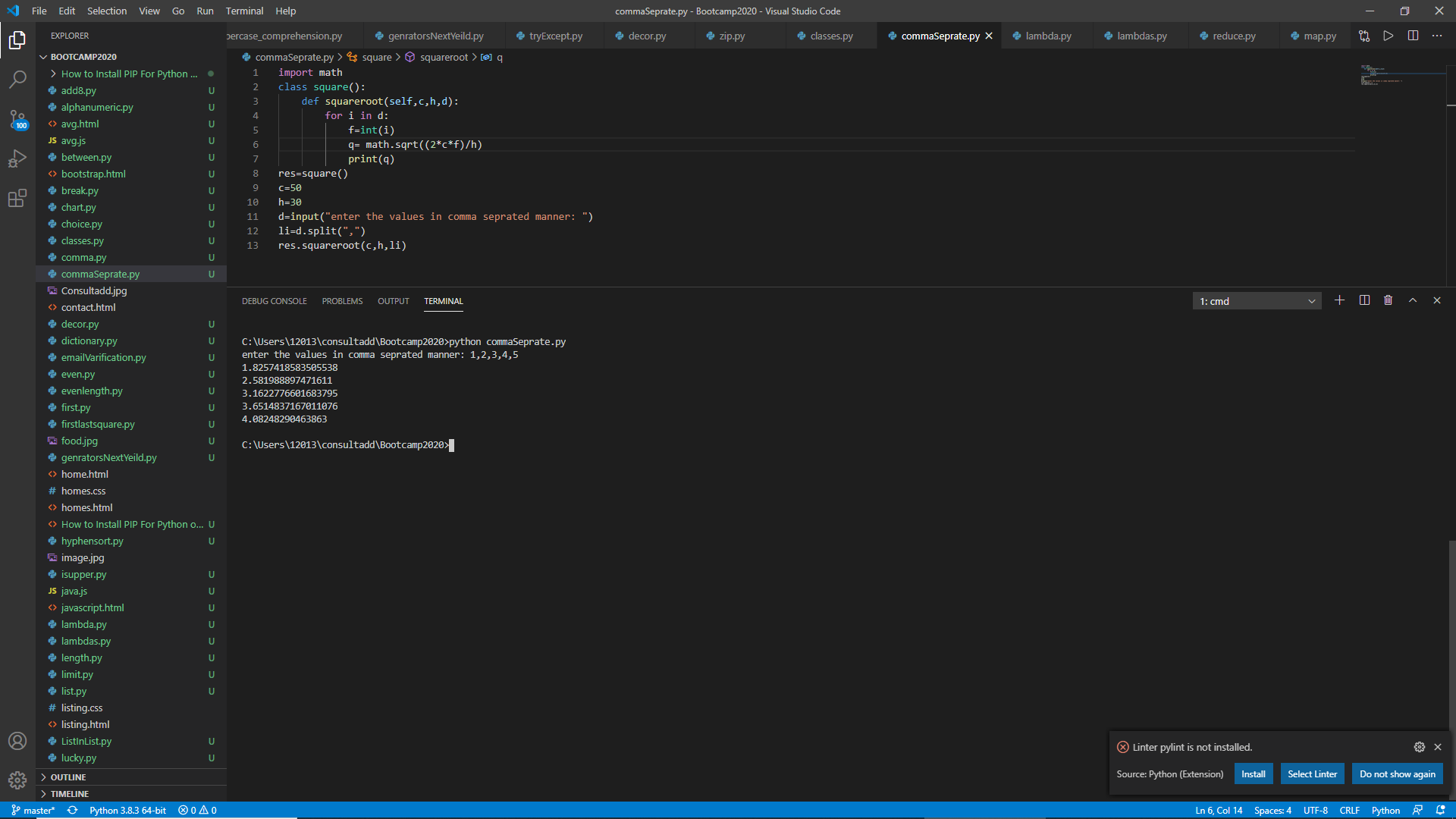
1. Write a program that calculates and prints the value according to the given formula:

Q= Square root of [(2\*C\*D)/H]

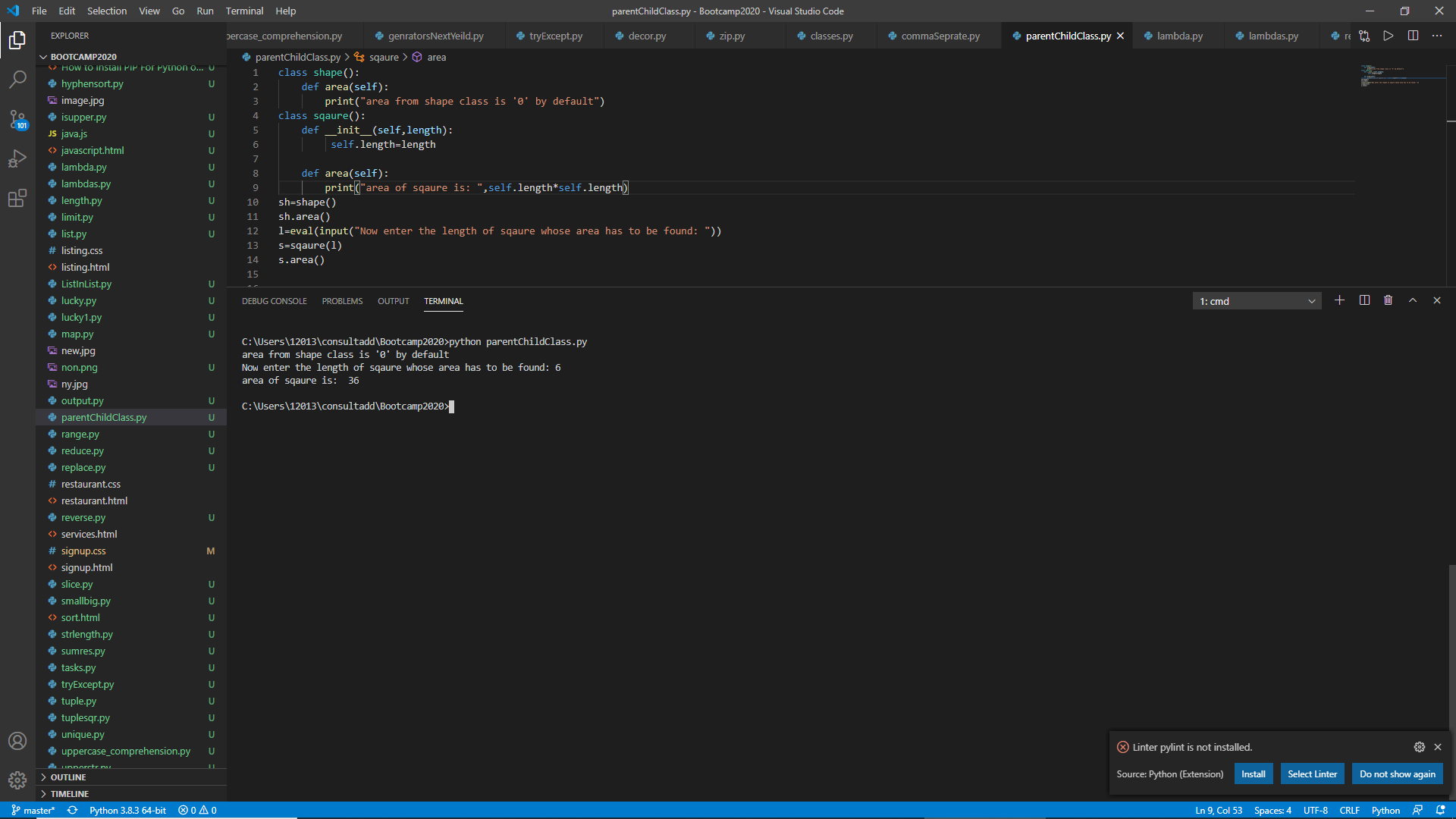
Following are the fixed values of C and H:

C is 50. H is 30.

D is the variable whose values should be input to your program in a comma-separated sequence



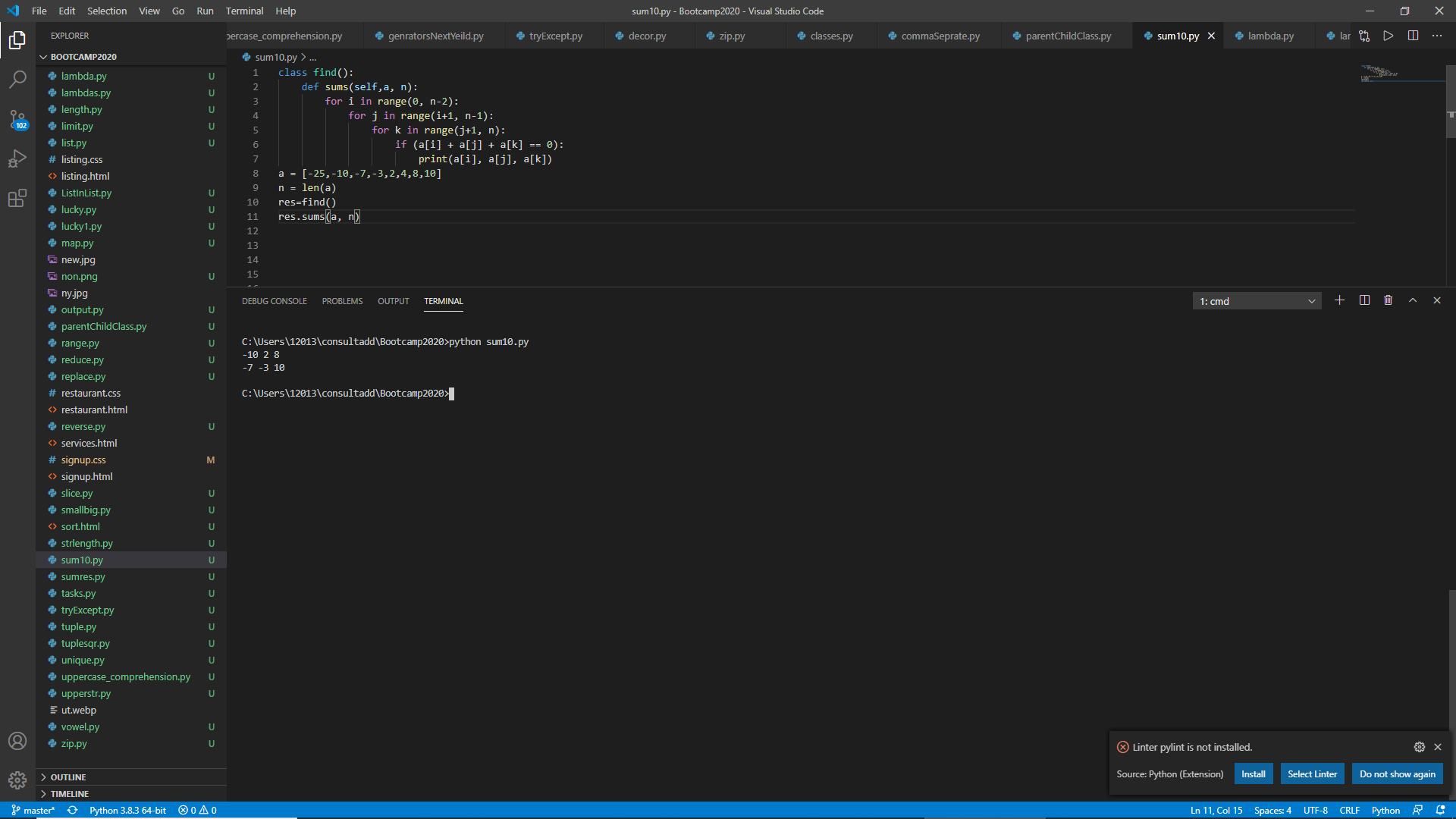
2. Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have an area function which can print the area of the shape where Shape’s area is 0 by default.



3. Create a class to find the three elements that sum to zero from a set of n real numbers.

Input array: [-25,-10,-7,-3,2,4,8,10]

Output: [[-10,2,8],[-7,-3,10]]



4. What is the output of the following code? Explain your answer as well.

* class Test:

def \_\_init\_\_(self):

self.x = 0

class Derived\_Test(Test):

def \_\_init\_\_(self):

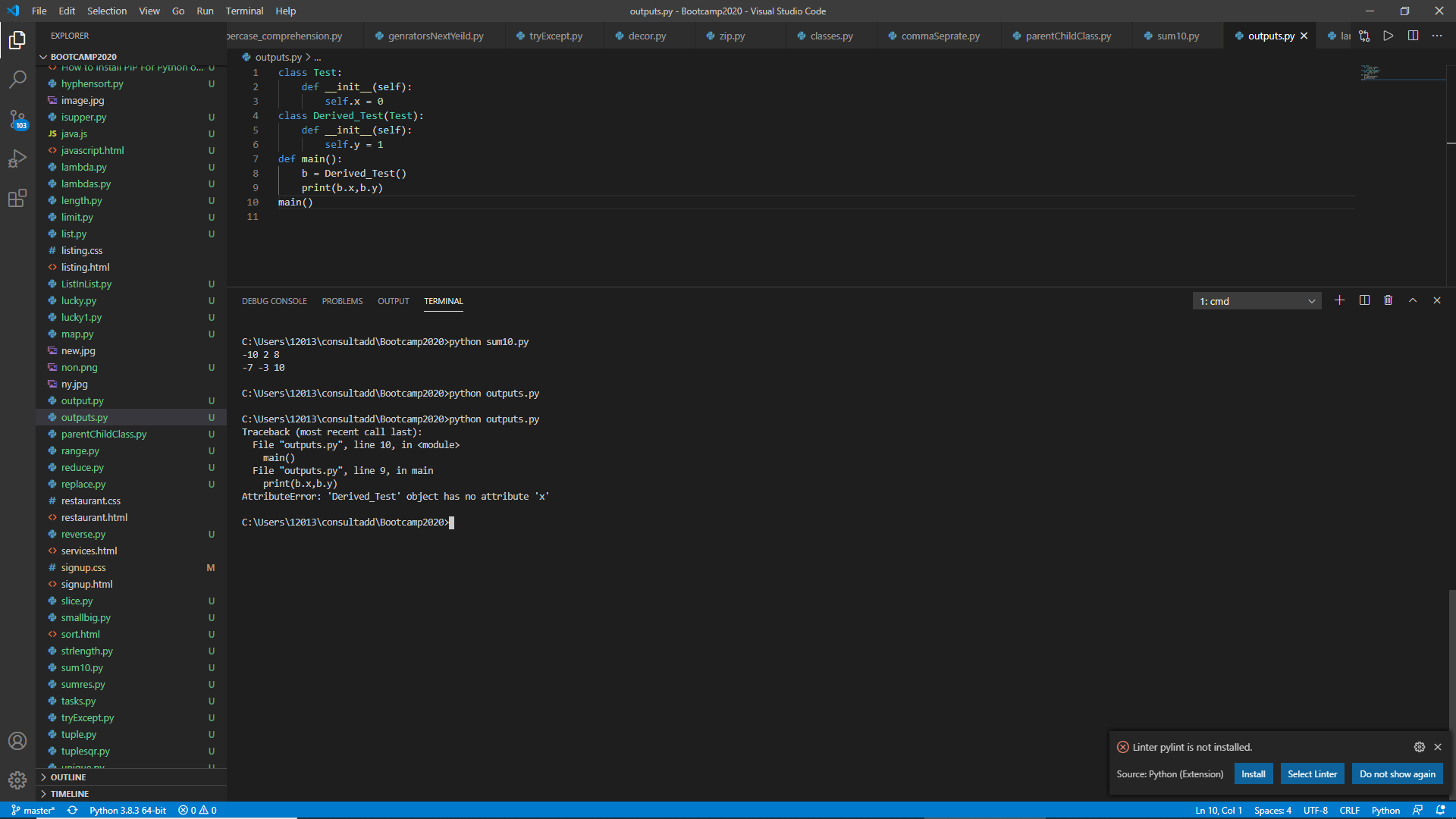
self.y = 1

def main():

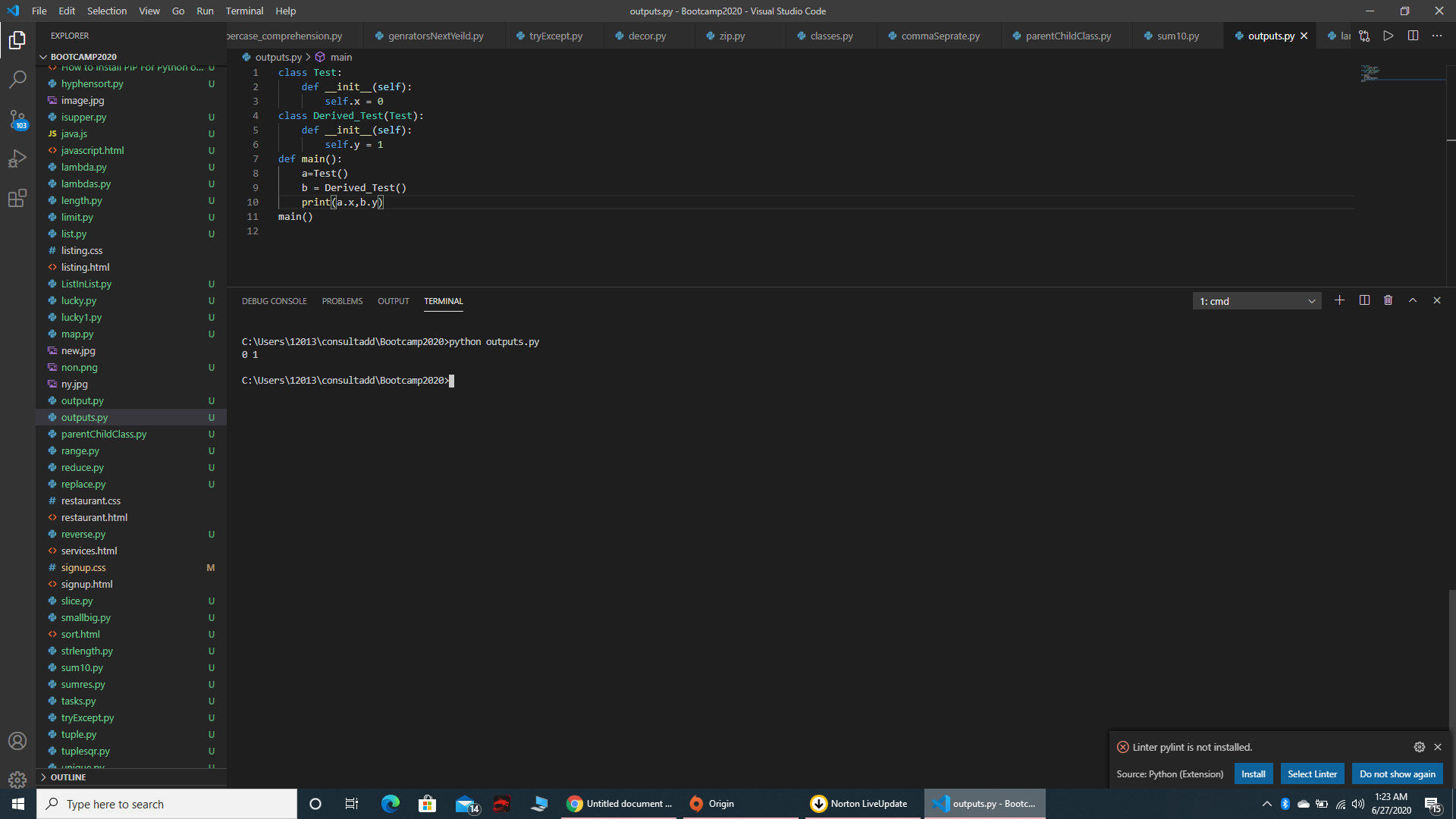
b = Derived\_Test()

print(b.x,b.y)

main()



Here in this program we are calling main() and in it we are making instance of Derived\_test and using that instance we are trying to call x but X is there in the constructor of test() function and cant be called using instance of Derived class in order to call X we have to initiate Test() by creating the object of Test()



* class A:

def \_\_init\_\_(self, x= 1):

self.x = x

class der(A):

def \_\_init\_\_(self,y = 2):

super().\_\_init\_\_()

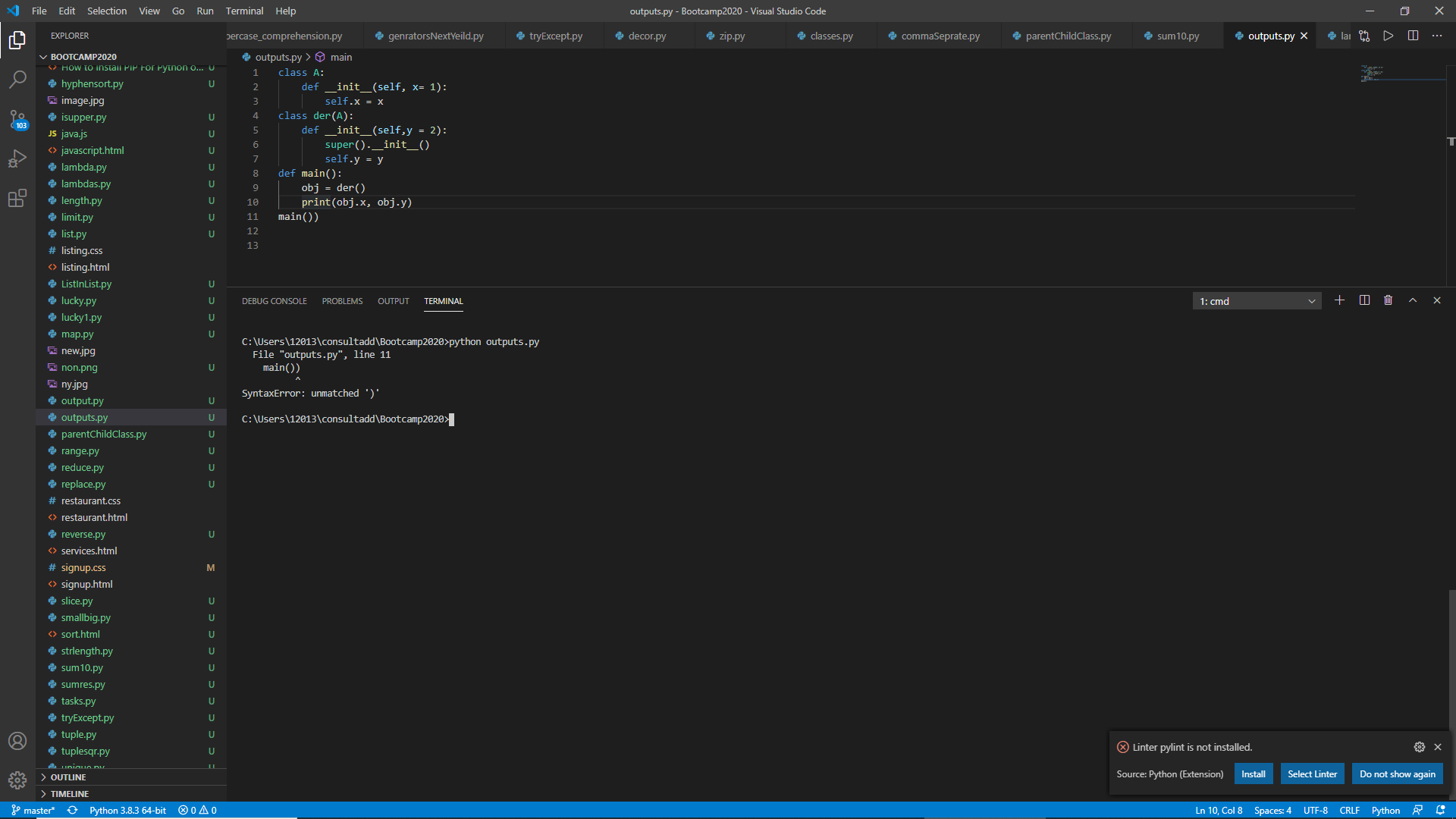
self.y = y

def main():

obj = der()

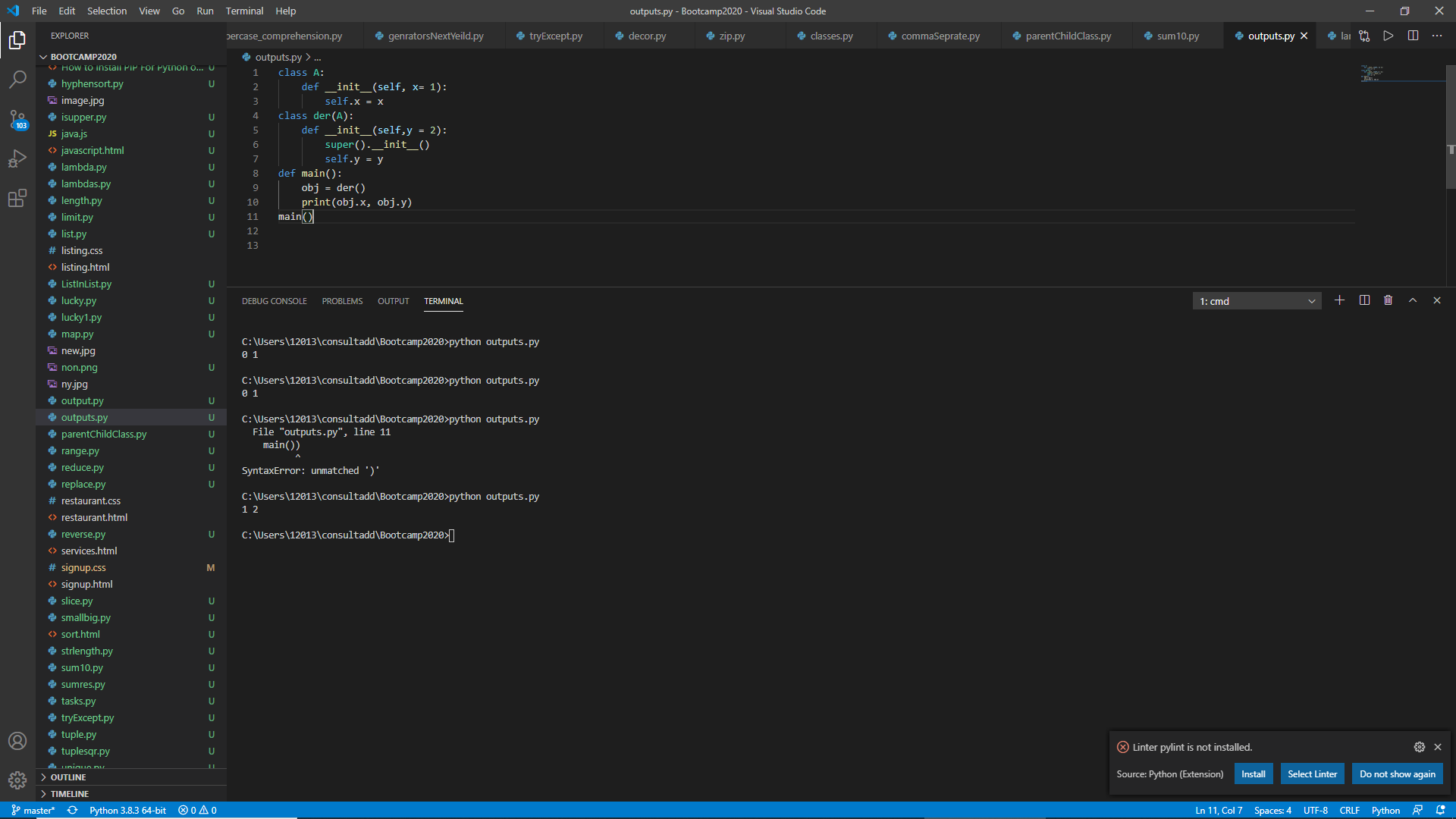
print(obj.x, obj.y)

main())



Here main has one extra bracket ie: main())

Corrected one :



* class A:

def \_\_init\_\_(self,x):

self.x = x

def count(self,x):

self.x = self.x+1

class B(A):

def \_\_init\_\_(self, y=0):

A.\_\_init\_\_(self, 3)

self.y = y

def count(self):

self.y += 1

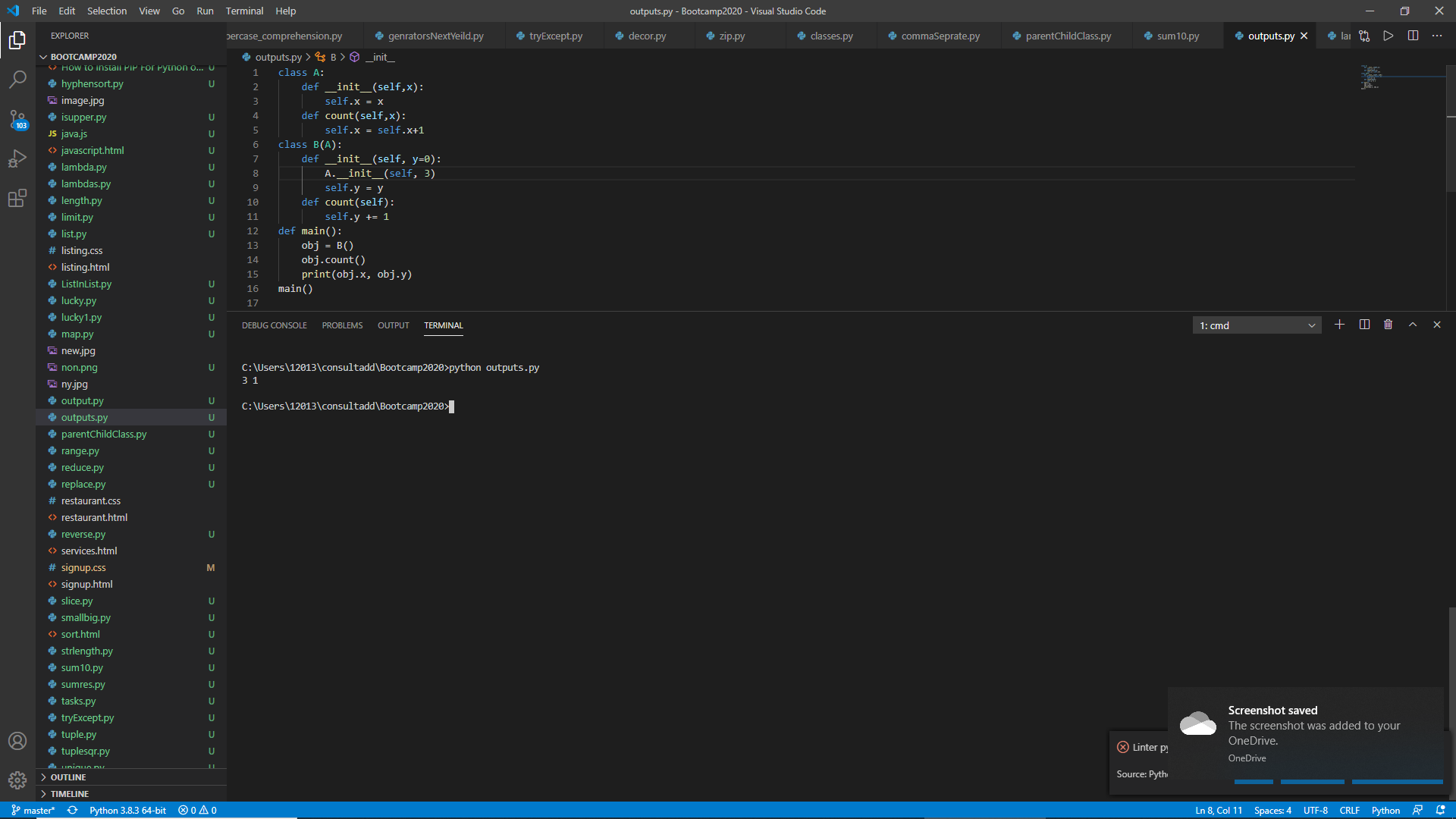
def main():

obj = B()

obj.count()

print(obj.x, obj.y)

main()



Here first we are calling main() then in main() we are creating obj of class B while instantiating the class B constructor gets called which sets Y=0 and then comes A.\_\_init\_\_ which actually calls constructor of class A and sends 3 to that constructor as parameter and X=3 now count() function of Class B gets called and Y gets incremented becomes 1

So output is

X=3, Y=1

* class A:

def \_\_init\_\_(self):

self.multiply(15)

print(self.i)

def multiply(self, i):

self.i = 4 \* i;

class B(A):

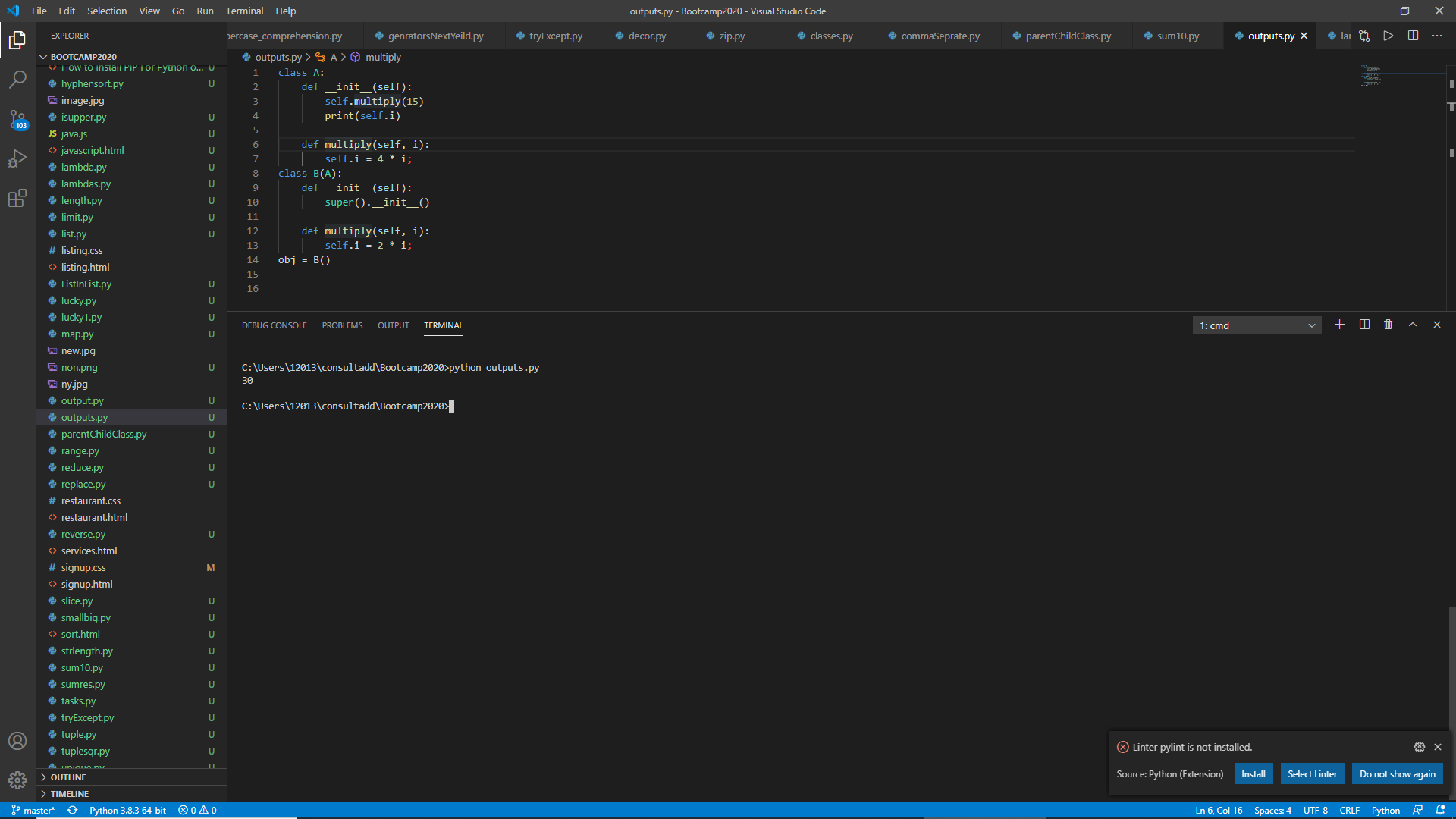
def \_\_init\_\_(self):

super().\_\_init\_\_()

def multiply(self, i):

self.i = 2 \* i;

obj = B()



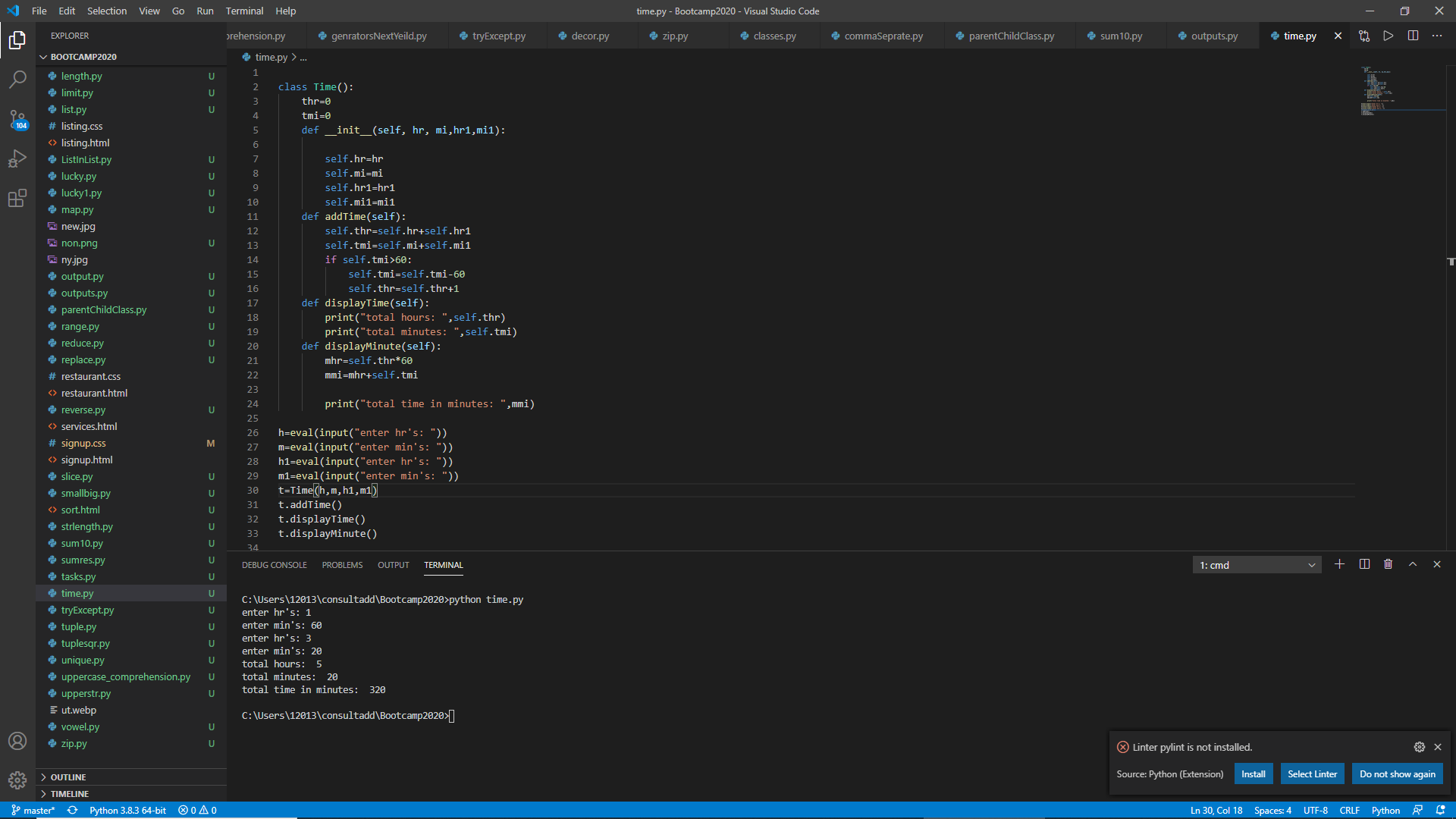
instantiating the object of Class B which is inheriting the properties of Class A now constructor of B gets called and using super we call constructor of A which then call multiply() function from B while passing 15 as argument now interpreter goes to multiply() and multiply 15 with 2 and returns to the print statement and prints I=30

5. Create a Time class and initialize it with hours and minutes.

Make a method addTime which should take two time object and add them. E.g.- (2 hour and 50 min)+(1 hr and 20 min) is (4 hr and 10 min)

Make a method displayTime which should print the time.

Make a method DisplayMinute which should display the total minutes in the Time. E.g.- (1 hr 2 min) should display 62 minute.



6. Write a Person class with an instance variable, , and a constructor that takes an integer, , as a parameter. The constructor must assign to after confirming the argument passed as is not negative; if a negative argument is passed as , the constructor should set to and print Age is not valid, setting age to 0.. In addition, you must write the following instance methods:

1. yearPasses() should increase the instance variable by .
2. amIOld() should perform the following conditional actions:
   * If , print You are young..
   * If and , print You are a teenager..
   * Otherwise, print You are old..

Sample Input:

4

-1

10

16

18

Sample Output:

Age is not valid, setting age to 0.

You are young.

You are young.

You are young.

You are a teenager.

You are a teenager.

You are old.

You are old.

You are old.

