Assignment # 1:

1.1 Tasks to do

1. Open the file a01.py and look between the markers. You need to save the result of adding num1 and num2 into the variable v.

2. Then, run local tests using the following command: python run.py local 3.

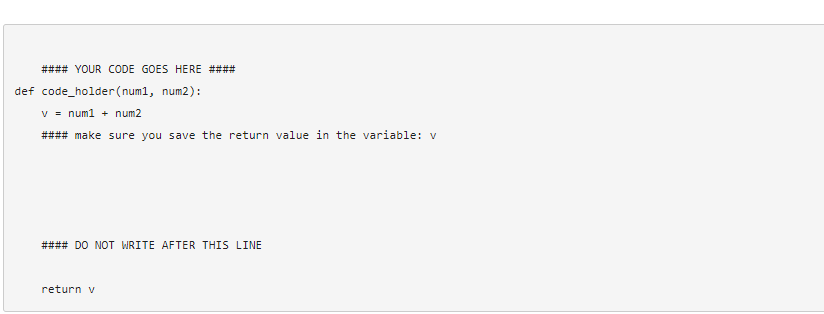
If all tests succeed, submit your assignment using the following command: python run.py remote 4.

It will ask for your email and submission password. You can get both of these from the assignment page where you downloaded the zip file from. 5.

Once the submission is successful, you may visit the Auto grader front-end to view your submission history. 6.

If you face any issue, please send me an email informing me of the error. We’re still trying to work out the issues in this new system. So, your help will be greatly appreciated.

**My-code**



ASSIGNMENT # 2

1.1 Tasks to do

1. Open the le a02.py and look between the markers. Write a function with the exact name of net Income that takes two arguments, one for current salary and other for income tax ratio. The income tax ratio may change but the default value is 2 %. Net Income function will deduct the tax from salary and return net salary. e.g. Logic: let salary=10,000 then net salary= (current salary) minus (current salary multiplied by tax) i.e. (10,000)-(10,000 x 2/100)= 10,000- 200= 9800.0

2. Then, run local tests using the following command: python run.py local

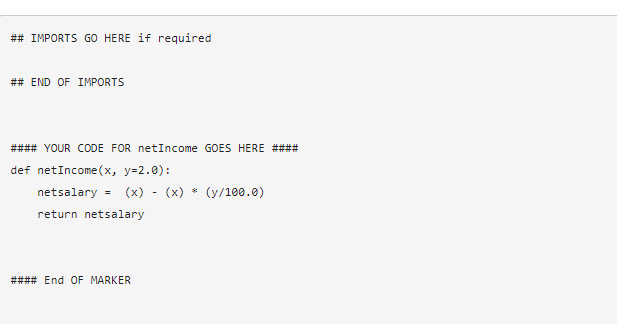
3. If all tests succeed, submit your assignment using the following command: python run.py remote

4. It will ask for your email and submission password. You can get both of these from the assignment page where you downloaded the zip le from.

5. Once the submission is successful, you may visit the Auto grader front-end to view your submission history.

6. If you face any issue, please send me an email informing me of the error. We're still trying to work out the issues in this new system. So, your help will be greatly appreciated.

**MY CODE**

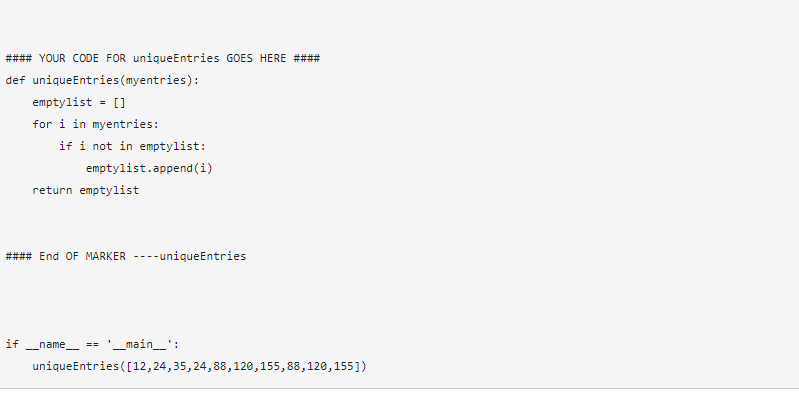


ASSIGNMENT # 3

1.1 Tasks to do:

1. Open the le a04.py and look between the markers. Write a function 'uniqueEntries' that takes a list/string and returns a list of its unique entries and duplicate entries

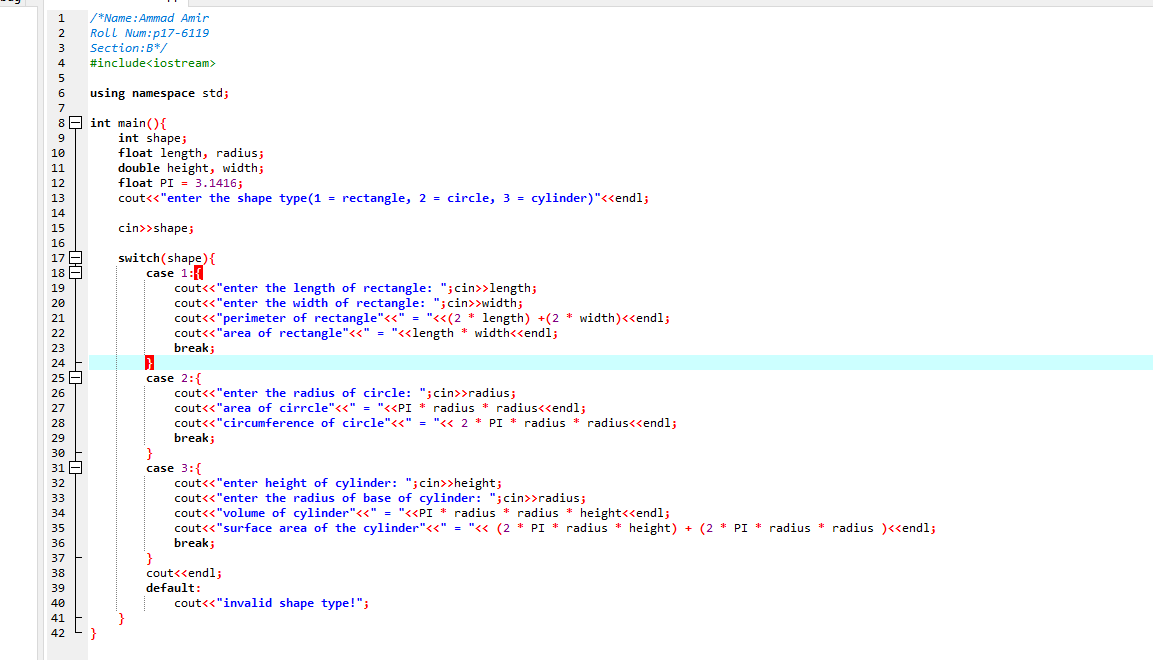
**MY CODE**



**ASSIGNMENT # 4**

This was a c++ assignment:

**MY CODE:**

OUTPUT:

