1.1 Write a Python Program(with class concepts) to find the area of the triangle using the below

formula.

area = (s\*(s-a)\*(s-b)\*(s-c)) \*\* 0.5

Function to take the length of the sides of triangle from user should be defined in the parent

class and function to calculate the area should be defined in subclass.

Code:

class Triangle:

def \_\_init\_\_(self, side1, side2, side3):

self.side1 = side1

self.side2 = side2

self.side3 = side3

print ("Initialised Triagle parent class [" + str(side1) + "," + str(side2) + "," + str(side3) + "]")

class Triangle\_Utilities(Triangle):

def \_\_init\_\_(self, side1, side2, side3):

print ("Initialised Child class" )

super(Triangle\_Utilities, self).\_\_init\_\_(side1, side2, side3)

def get\_area(self):

s = (self.side1 + self.side2 + self.side3)/2

print (str(s))

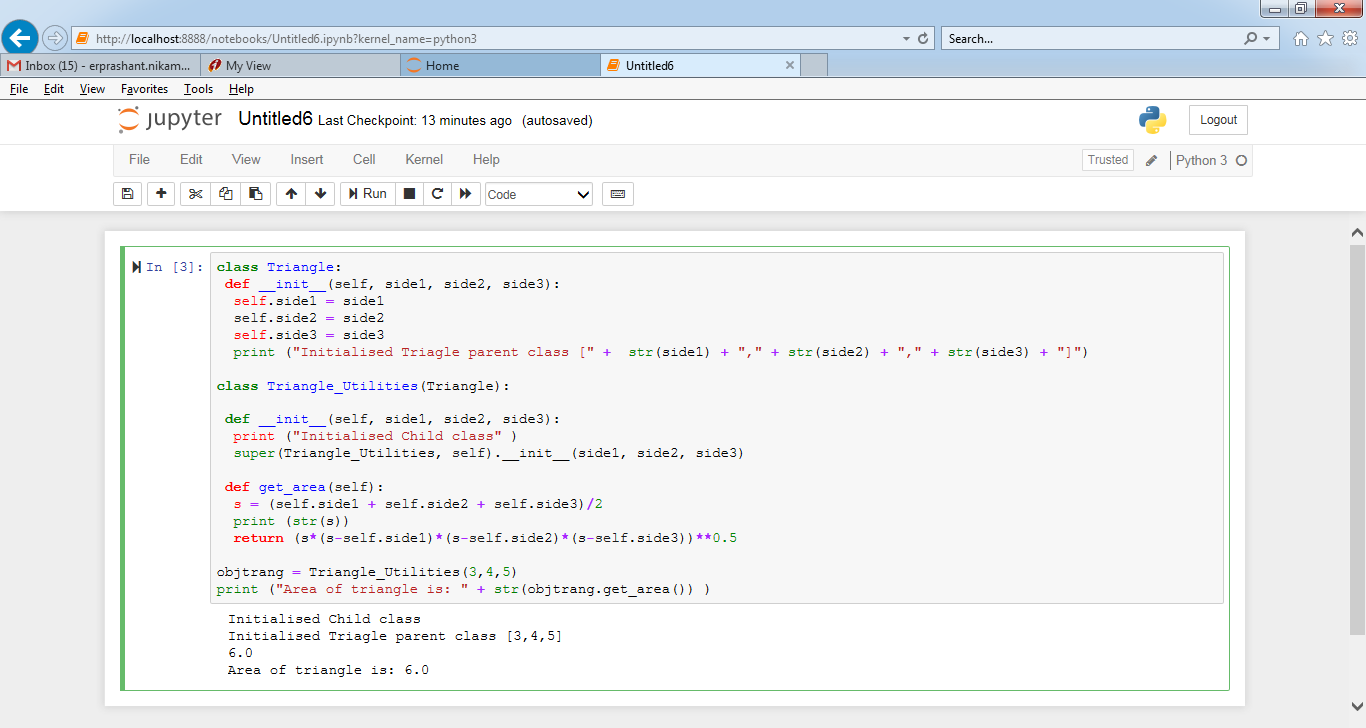
return (s\*(s-self.side1)\*(s-self.side2)\*(s-self.side3))\*\*0.5

objtrang = Triangle\_Utilities(3,4,5)

print ("Area of triangle is: " + str(objtrang.get\_area()) )

OutPut:

;



1.2 Write a function filter\_long\_words() that takes a list of words and an integer n and returns

the list of words that are longer than n.

Code:

class list\_Utilities:

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

self.wordlist = wordlist

print ("Initialised list\_Utilities object")

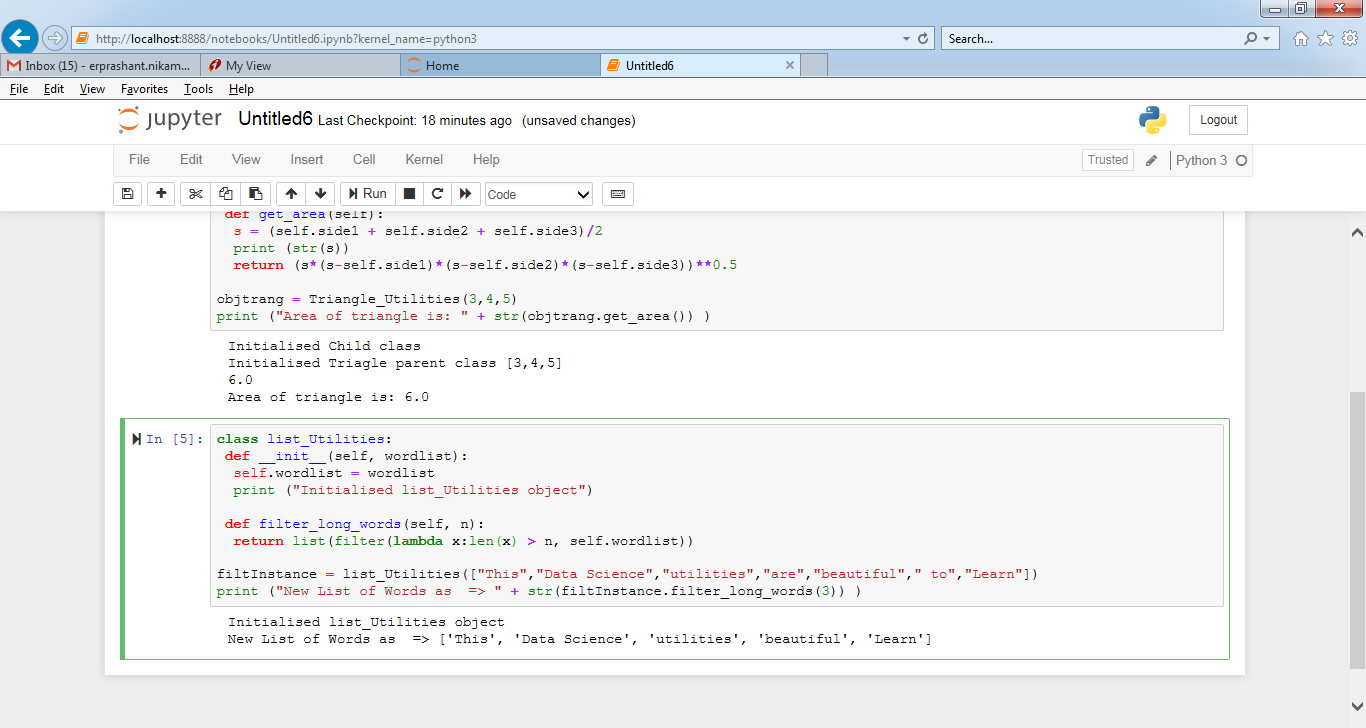
def filter\_long\_words(self, n):

return list(filter(lambda x:len(x) > n, self.wordlist))

filtInstance = list\_Utilities(["This","Data Science","utilities","are","beautiful"," to","Learn"])

print ("New List of Words as => " + str(filtInstance.filter\_long\_words(3)) )

OutPut:



2.1 Write a Python program using function concept that maps list of words into a list of integers

representing the lengths of the corresponding words.

Hint: ​If a list [ ab,cde,erty] is passed on to the python function output should come as [2,3,4]

Here 2,3 and 4 are the lengths of the words in the list.

Code:

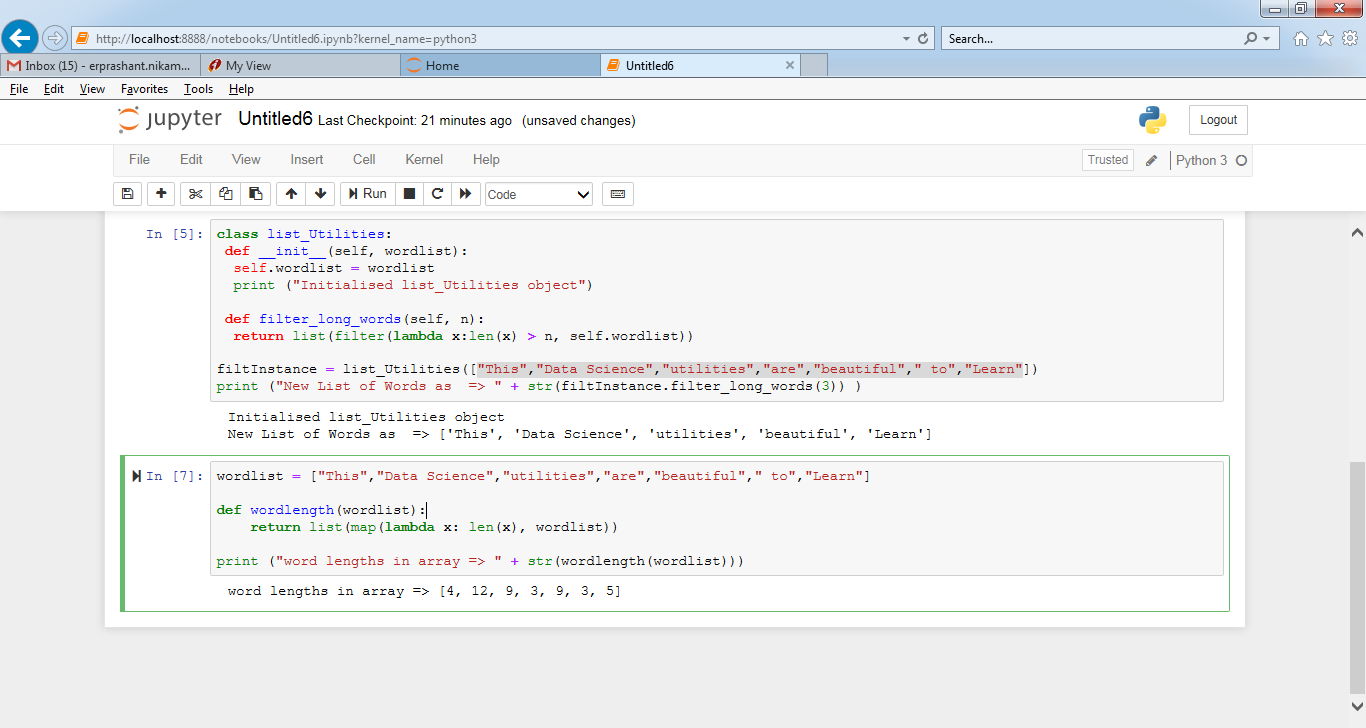
wordlist = ["This","Data Science","utilities","are","beautiful"," to","Learn"]

def wordlength(wordlist):

return list(map(lambda x: len(x), wordlist))

print ("word lengths in array => " + str(wordlength(wordlist)))

OutPut:



2.2 Write a Python function which takes a character (i.e. a string of length 1) and returns True if

it is a vowel, False otherwise.

Code:

def vowel\_check(char):

if(char == 'a' or char == 'e' or char == 'i' or char == 'o' or char == 'u'):

return True

else:

return False

char = input("Enter character: ");

if (char.isalpha() == False):

exit();

# Invoke function

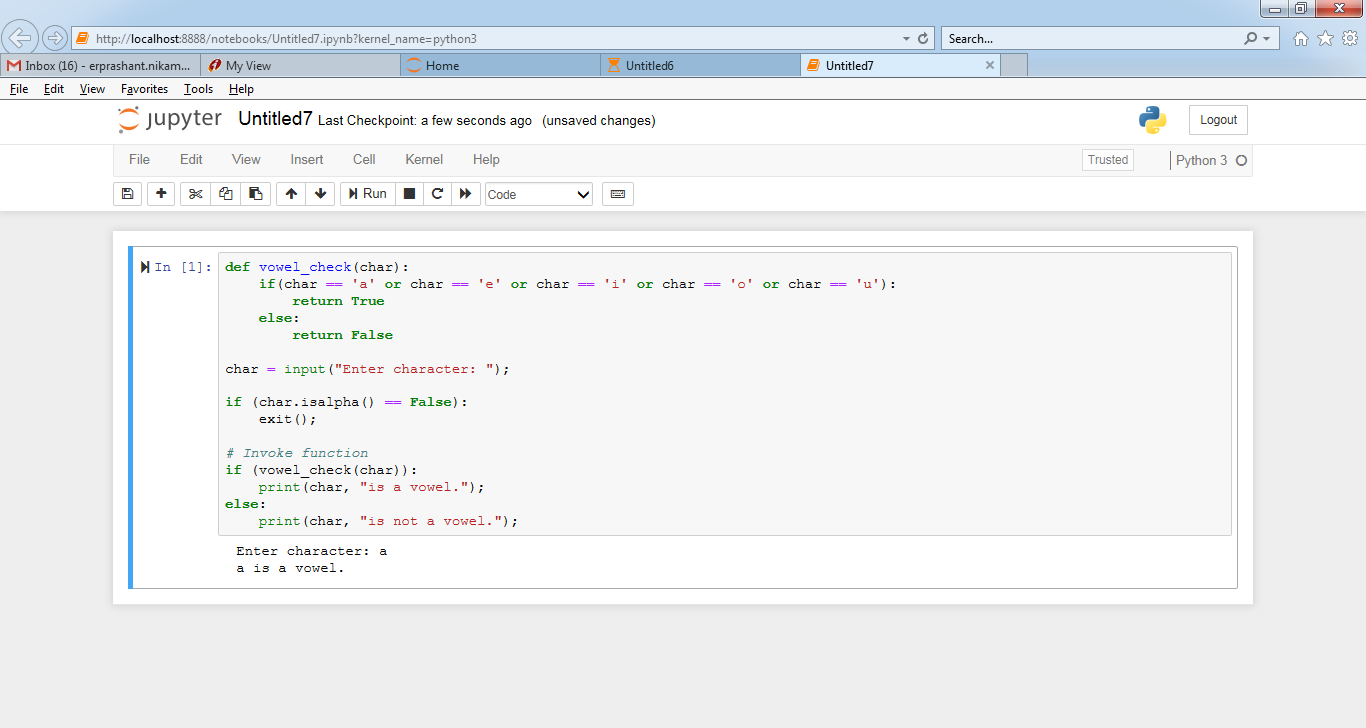
if (vowel\_check(char)):

print(char, "is a vowel.");

else:

print(char, "is not a vowel.");

OutPut:



NOTE:​ ​The​ ​solution​ ​shared​ ​through​ ​Github​ ​should​ ​contain​ ​the​ ​source code​ ​used​ ​and​ ​the​

​screenshot​ ​of​ ​the​ ​output.