ADVANCE MATHEMATICAL FUNCTIONS CALCULATOR

CODE:

#!/usr/bin/python

# -\*- coding: utf-8 -\*-

import re

class cal1:

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

self.p = p

def ctof(self):

# p is celsius

# this function comes under temperature conversion

print ('conversion from c to f is : %f' % (self.p \* 9 / 5 + 32))

# celcius

def ctok(self):

# p is celsius

# this function comes under temperature conversion

print ('conversion from celsius to kelvin is : %f' % (self.p+273.15))

def ftoc(self):

# p is fahrenheit

# this function comes under temperature conversion

print ('conversion from f to c is : %f' % ((self.p - 32) \* 5 / 9))

def ftok(self):

# p is fahrenheit

# this function comes under temperature conversion

print ('conversion from ff to k is : %f' % ((self.p - 32)\*5/9+273.15))

def ktoc(self):

# p is kelvin

# this function comes under temperature conversion

print ('conversion from kelvin to celsius is : %f' % (self.p-273.15))

def ktof(self):

# p is kelvin

# this function comes under temperature conversion

print ('conversion from k to f is : %f' % (self.p - 273.15))

def area\_circle(self):

# p is radius of the circle

# this operation comes under measurements of different shapes

print ('area of the circle is : %f' % (3.147 \* self.p \* self.p))

def area\_square(self):

# p is side of the square

# this operation comes under measurements of different shapes

print ('area of the square is : %f' % (self.p \* self.p))

def peri\_square(self):

# p is side of the square

# this operation comes under measurements of different shapes

print ('perimeter of the square is : %f' % (4 \* self.p))

def peri\_circle(self):

# p is radius of the circle

# this operation comes under measurements of different shapes

print ('perimeter of the circle is : %f' % (2 \* 3.147 \* self.p))

class cal2(cal1):

def \_\_init\_\_(self, p, q):

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

self.q = q

def APY(self):

# p is stated annual interest

# q is number of times compounded

# this function comes under banking

print ('APY in Banking is : %f' % ((1 + self.p / self.q)\*\*self.q - 1))

def DIR(self):

# p is monthly debt payment

# q is gross monthly income

# this function comes under banking

print ('Debt to Income Ratio in Banking is : %f' % (self.p/self.q))

def LV(self):

# p is loan amount

# q is value of collateral

# this function comes under banking sector

print ('Loan to Value in Banking is : %f' % (self.p / self.q))

def NII(self):

# p is interest income

# q is interest expense

# this function comes under banking sector

print ('Net Interest Income in Banking is : %f' % (self.p-self.q))

def LDR(self):

# p is loans

# q is deposit

# this function comes under banking sector

print ('Loan to Deposit Ratio is : %f' % (self.p / self.q))

def NIS(self):

# p is interest income rate

# q is interest expense rate

# this function comes under banking sector

print ('Net Interest Spread in Banking is : %f' % (self.p-self.q))

def NIM(self):

# p is net interest income

# q is avg earning assets

# this function comes under banking sector

print ('Net Interest Margin in Banking is : %f' % (self.p/self.q))

def profit(self):

# p is selling price of a product

# q is cost price of a product

# this function comes under banking sector

print ('profit is : %f' % (self.p - self.q))

def loss(self):

# p is cost price of a product

# q is selling price of a product

# this function comes under banking sector

print ('loss is : %f' % (self.p - self.q))

def area\_triangle(self):

# p is height of the triangle

# q is base of the triangle

# this operation comes under measurements of different shapes

print ('area of the triangle is : %f' % (self.p \* self.q / 2))

def area\_rectangle(self):

# p is height of the rectangle

# q is width of the rectangle

# this operation comes under measurements of different shapes

print ('area of the rectangle is : %f' % (self.p \* self.q))

def peri\_rectangle(self):

# p is height of the rectangle

# q is width of the rectangle

# this operation comes under measurements of different shapes

print ('perimeter of the rectangle is : %f' % (2 \* (self.p+self.q)))

superObj = cal1(50)

subObj = cal2(15, 10)

superObj.ctok()

superObj.ftoc()

superObj.ftok()

superObj.ktoc()

superObj.ktof()

superObj.area\_circle()

superObj.area\_square()

superObj.peri\_square()

superObj.peri\_circle()

subObj.ctof()

subObj.APY()

subObj.DIR()

subObj.LV()

subObj.NII()

subObj.NIS()

subObj.LDR()

subObj.NIM()

subObj.profit()

subObj.loss()

subObj.area\_triangle()

subObj.area\_rectangle()

subObj.peri\_rectangle()

# Magic Method

print (dir())

print (id(cal1))

print (id(cal2))

num = 100

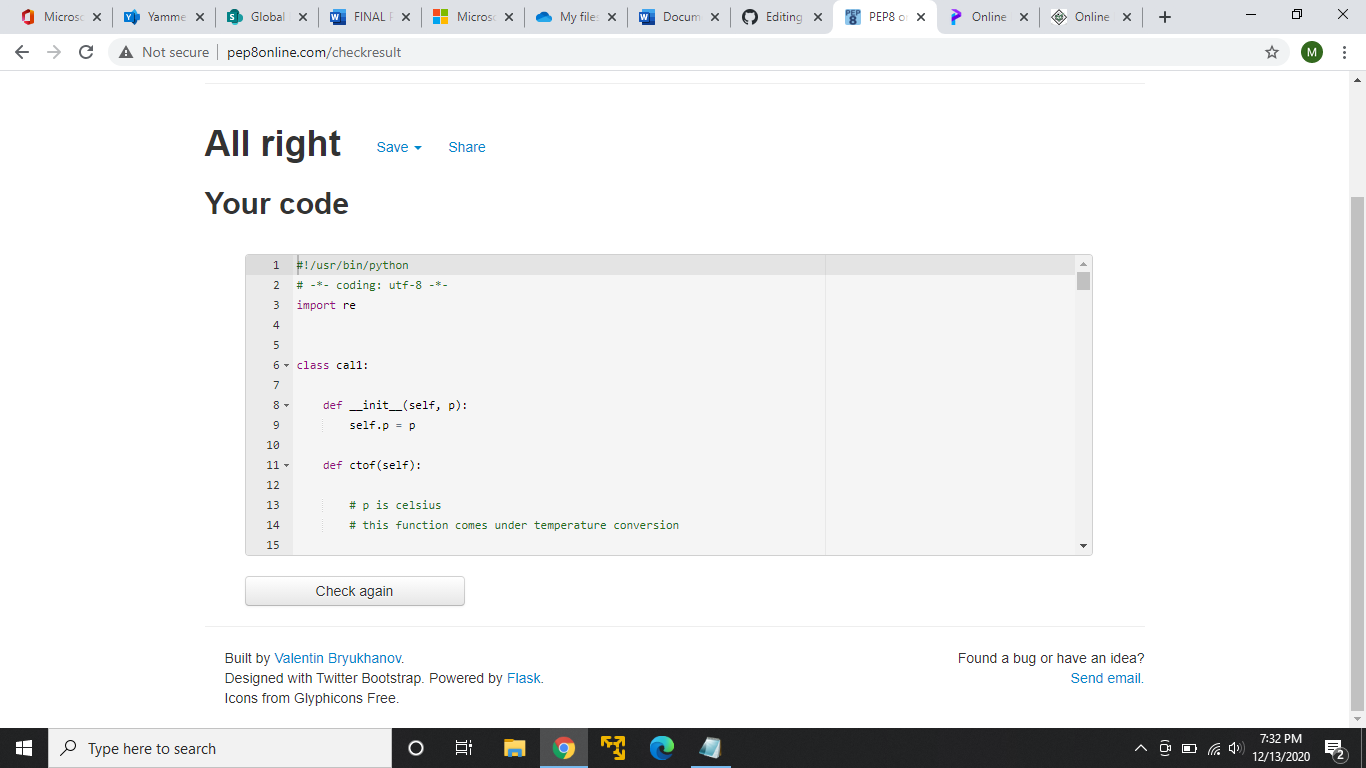
print (num.\_\_add\_\_(500))

print (num.\_\_floordiv\_\_(5))

print (num.\_\_sub\_\_(5))

print (re.match("[a-z 0-9]+@[gmail]+\.[a-z]{3}", 'madhu5@gmail.com'))

PEP8 ONLINE CHECK:



GITHUB LINK: https://github.com/99003154/Python.Mini