```
!date
```

```
Fri Feb 11 01:14:35 UTC 2022
```

Please run the above line to refresh the date before your submission.

CSCI-SHU 210 Data Structures

Recitation 1 Object-Oriented Programming Review

You should work on the tasks as written in the worksheet.

- For students who have recitation on Wednesday, you should submit your solutions by **Feb 11th** Friday 11:59pm.
- For students who have recitation on Thursday, you should submit your solutions by **Feb 12th** Saturday 11:59pm.
- For students who have recitation on Friday, you should submit your solutions by **Feb 13th** Sunday 11:59pm.

Name: Peter Yao

NetID: yy4108

Please submit the following items to the Gradescope:

- URL: Your Colab notebook link. Click the Share button at the top-right, share with NYU, and paste to Gradescope
- PDF: The printout of your run in Colab notebook in pdf format

Topic 1 (Creating a class)

```
class Student:
    def __init__(self, name, age, GPA):
        self.name=name
        self.age=age
        self.GPA=GPA


    def get_GPA(self):
        return self.GPA

    def set_GPA(self, GPA):
        self.GPA=GPA
```

```
def main():
    bob = Student("Bob", 15, 3.0)
    print(bob.get_GPA()) #3.0

    bob.set_GPA(4.0)
    print(bob.get_GPA()) #4.0

if __name__ == '__main__':
    main()
```

```
3.0
4.0
```

What does the keyword self do in Python?

it refers to all the future initializd objects of this class

Topic 2 (underscore ***** functions):

```
class Pizza:
    def __init__(self, price):
        self.price = price

    def __add__(self, other):
        new_pizza = Pizza(self.price)
        new_pizza += other
        return new_pizza

    def __iadd__(self, other):
        self.price += other.price
        return self

    def __str__(self):
        return "the price is, " + str(self.price)



def main():
    pizza1 = Pizza(5)
    pizza2 = Pizza(6)
    pizza1 + pizza2
    pizza1 += pizza2
    print(pizza1)

if __name__ == '__main__':

    main()
```

```
the price is, 11
```

a) What does the code above print? Don’t run the program, try to predict the output first.

the price is, 11

b) Complete the following table, suppose the variable name is X. When will these underscore functions get called? Answer for 1st row has been given for your convenience.

Double-click to edit

1. X.**getitem**(self, index) : X[index]
2. X.**setitem**(self, index, value) : X[index]=value
3. X.**delitem**(self, index) : del X[index]
4. X.**add**(self, other) : X+other
5. X.**iadd**(self, other) : X+=other
6. X.**eq**(self, other) : X==other
7. X.**len**(self) : len(X)
8. X.**str**(self) : str(X)
9. X.**repr**(self) : repr(X)
10. X.**contains**(self, value) : value in X
11. X.**iter**(self) : iter(X)

Topic 3 (Inheritance):

```
class Tree:
    def __init__(self, name, age):
        self._name = name
        self._age = age

    def get_name(self):
        return self._name


class Palm(Tree): # Palm(Tree) means, Palm inherits Tree.
    def __init__(self, name, age, color):
        # First you have to initialize the parent class. What should we write here?
        super().__init__(name, age)

        self._color = color

    def get_color(self):

        return self._color

def main():
    palm1 = Palm("Lucky", 30, "Green")
    print(palm1.get_name())  # What does this print (1)?
    print(palm1.get_color())  # What does this print (2)?
    tree1 = Tree("Funny", 20)
```

```
    print(tree1.get_name())  # What does this print (3)?
    print(tree1.get_color())  # What does this print (4)?

if __name__ == '__main__':
    main()
```

```
Lucky
Green
Funny
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-14-daa53178885f> in <module>()
     27
     28 if __name__ == '__main__':
---> 29     main()

<ipython-input-14-daa53178885f> in main()
     24     tree1 = Tree("Funny", 20)
     25     print(tree1.get_name())  # What does this print (3)?
---> 26     print(tree1.get_color())  # What does this print (4)?
     27
     28 if __name__ == '__main__':

AttributeError: 'Tree' object has no attribute 'get_color'
```

SEARCH STACK OVERFLOW

What does the code above print? Don't run the program, try to predict the output first.

1. What is the output for print (1)? Lucky
2. What is the output for print (2)? Green
3. What is the output for print (3)? Funny
4. What is the output for print (4)? an error

Topic 4 (Misc):

```
# Coding 1
import math
class Shape:
    def __init__(self, name):
        self.name = name

    def get_name(self):

        return self.name

class Circle:
    def __init__(self, name, radius):
        self.name=name
        self.radius=radius
```

```
    def calc_area(self):
        return 3.14*(self.radius**2)

    def calc_perimeter(self):
        return 2*3.14*self.radius

class Rectangle:
    def __init__(self, name, width, height):
        self.name=name
        self.width=width
        self.height=height

    def calc_area(self):
        return self.width*self.height

    def calc_perimeter(self):
        return 2*(self.width+self.height)

def main():
    circle1 = Circle("fancy", 5)
    print(circle1.calc_area())  #78.5
    print(circle1.calc_perimeter())  #31.400000000000002

    rectangle1 = Rectangle("lucky", 3, 4)
    print(rectangle1.calc_area())  #12
    print(rectangle1.calc_perimeter())  #14

if __name__ == '__main__':
    main()

     78.5
     31.400000000000002
     12
     14


# Coding 2

class Polynomial:
    def __init__(self, coeffs):
        self.coeffs = coeffs

    def evaluate_at(self,num):
      n=len(self.coeffs)
      res=0
      for i in range(n):
        res+=self.coeffs[i]*(num**(n-i-1))
      return res

    def __str__(self):
        n=len(self.coeffs)
        res=[]
        for i in range(n-1):
            res.append(str(self.coeffs[i])+'x^'+str(n-i-1))
        res.append(str(self.coeffs[-1]))
        return ' + '.join(res)

    def __iadd__(self, other):
        while len(other.coeffs)<len(self.coeffs):
```

```
            other.coeffs.insert(0,0)
        while len(other.coeffs)>len(self.coeffs):
            self.coeffs.insert(0,0)
        n=len(self.coeffs)
        temp=[0 for i in range(n)]
        for i in range(n):
            temp[i]=self.coeffs[i]+other.coeffs[i]
        return Polynomial(temp)




def main():
    # 1x^4 + 2x^3 + 3x^2 + 4x + 5
    coeffs = [1,2,3,4,5]
    poly = Polynomial(coeffs)
    print(poly.evaluate_at(2))   # 57
    print(poly.evaluate_at(3))   # 179
    print(poly)  # Outputs: 1x^4 + 2x^3 + 3x^2 + 4x^1 + 5

    # 4x^3 + 6x^2 + 8x^1 + 10
    coeffs = [4,6,8,10]
    poly2 = Polynomial(coeffs)
    print(poly2)  # Outputs: 4x^3 + 6x^2 + 8x^1 + 10
    poly += poly2
    print(poly)  # Outputs: 1x^4 + 6x^3 + 9x^2 + 12x^1 + 15

if __name__ == '__main__':
    main()


 57
 179
 1x^4 + 2x^3 + 3x^2 + 4x^1 + 5
 4x^3 + 6x^2 + 8x^1 + 10
 1x^4 + 6x^3 + 9x^2 + 12x^1 + 15
```

Topic 5 Problem 1 Reverse Digit

```
#Given a 32-bit signed integer, return the reversed digits of this integer.
#Note:
#Try to solve this problem using math equations.
#Eg: don’t cast this number to str/list/etc\]

def reverse(x):
    q=[]
    def negative_reverse(x):
        return -reverse(-x)
    if x<0:
        return negative_reverse(x)
    while x:
        q.append(x%10)
        x=x//10
```

```
    n=len(q)
    res=0
    for i in range(n):
        temp=q.pop(0)
        res+=temp*(10**(n-i-1))
    return res



# test case
print(reverse(1200)) #21
print(reverse(123)) #321
print(reverse(-123)) #-321
```

```
21
321
-321
```

Topic 5 Problem 2

```
#Write a program to check whether a given number is a Funny number.
#Funny numbers are positive numbers whose prime factors only include 2, 3, 5.
#For example, 6, 8 are Funny while 14 is not Funny since it includes another prime factor


def isFunny(num):
    while num%2 ==0:
        num=num/2
    while num%3==0:
        num=num/3
    while num%5==0:
        num=num/5
    if num==1:
        return True
    return False



# test case
print(isFunny(6)) #True
print(isFunny(8)) #True
print(isFunny(14)) #False
```

```
True
True
False
```

0 秒　完成时间：09:15