Python Assignment

Data Types and Type Conversions

1) Write a Python program to get a string made of the first 2 and the last 2 chars from a given string. If the string length is less than 2, return instead of the empty string.

```
def fun(str):
  if len(str)<2:
    return ''


  return str[0:2]+str[-2:]



print(fun('Nihal'))
print(fun('w3resource'))
print(fun('w3'))
print(fun('w'))
```

```
Nial
w3ce
w3w3
```

2) Write a Python program to get the largest number from a list.

```
l=[2,4,6,7,12,30,14,8,9,27]
print(max(l))
```

```
30
```

3) Write a Python script to concatenate following dictionaries to create a new one.

```
dic1={1:10, 2:20}

dic2={3:30, 4:40}

dic3={5:50,6:60}

d={}

for i in (dic1 , dic2 , dic3):
  d.update(i)
print(d)
```

```
print(d)
```

```
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

Operators

1)Write a Python program which accepts the radius of a circle from the user and compute the area.

```
from math import pi

r=float(input("Enter radious of circle:"))
area=str(pi * r**2)
print(area)
```

```
Enter radious of circle:1.1
3.8013271108436504
```

2)Write a Python program that accepts an integer (n) and computes the value of n+nn+nnn.

```
a=int(input("Enter integer number: "))
b1=int("%s" %a)
b2=int("%s%s"%(a,a))
b3=int("%s%s%s"%(a,a,a))
b4=b1+b2+b3
print(b4)
```

```
Enter integer number: 5
615
```

3)Write a Python program to calculate the number of days between two dates.

```
from datetime import date
s=date(2014,7,2)
l=date(2014,7,11)
days=l-s
print(days)
```

```
9 days, 0:00:00
```

Decision Control

1)Write a Python program to get a new string from a given string where "Is" has been added to the front. If the given string already begins with "Is" then return the string unchanged.

```
def fun(str):
  if len(str)>=2 and str[0:2]=='Is':
    return str

  return 'Is'+str

print(fun('Isempty'))
print(fun('Nihal'))
```

```
Isempty
IsNihal
```

2)Write a Python program to find whether a given number (accept from the user) is even or odd, print out an appropriate message to the user.

```
num=int(input("Enter number: "))
x=num%2
if x==0:
  print("Given number is even ")
else:
  print("Given number is odd ")
```

```
Enter number: 80
Given number is even
```

3)Write a Python program to test whether a passed letter is a vowel or not.

```
l=['a','e','i','o','u','A','E','I','O','U']

x=input("Enter letter: ")
if x in l[0:]:
  print("True")
else :
  print("False")
```

```
Enter letter: U
True
```

Loops

1) Write a Python program to concatenate all elements in a list into a string and return it.

```
def fun(list):
  temp=''
  for i in list:
    temp+=str(i)
  return temp

print(fun([1,2,3,5,6,7]))
```

```
print(fun([1,2,3,5,6,7]))
```

```
123567
```

2)Write a Python program to print out a set containing all the colors from color_list_1 which are not present in color_list_2.

```
color_list1=set(['White','Black','Red'])
color_list2=set(['Red','Green'])

print(color_list1.difference(color_list2))
```

```
{'Black', 'White'}
```

Functions

1)Write a function in a Python program to compute the greatest common divisor (GCD) of two positive integers.

```
def gcd(x,y):
  res=1
  if x%y==0:
    return y

  for i in range(int(y/2),0,-1):
    if x%i==0 and y%i==0 :
      gcd=i
      break

  return gcd

print(gcd(14,22))
print(gcd(12,16))
print(gcd(45,66))
```

```
2
4
3
```

2)Write a Python function which accepts two arguments x and y and returns (x + y) * (x + y).

```
def fun(x,y):
  z=x*x + 2*x*y + y*y
  return z

p=4
q=3
z=fun(p,q)
print("({},{}) ^2) = {}".format(p,q,z))
```

```
(4,3) ^2) = 49
```

Variable Scoping

1)Given the below code snippet, answer the following questions.

```
def num_square(num):
  square = num * num
  return square


input_num = 100
if input_num >100:
  result = num_square(5)


print(result)
```

```
25
25
```

1-What is the scope of num, square, input_num, result?

```
#Local scope = num , square
#Global scope = input_num,result
```

2-What is the lifetime of each variable? When will they be created and destroyed?

```
#When the funtion is called the local variable num and square is created and destroyed aft

#input_num and result are the global which is last long even the funtion is terminated
```

3-What would happen if input_num had a value of less than 100?

```
#Even if input_num is 100 or greater the result will be 25 because result is a global vari
```

4-What would be the scope and value of `result` if input_num has a value of less than 100?

```
#scope of result will be global and value will be the same 25
```

Classes and OOP

1)

```
class Country:

  AVG_POPULATION=10

  def __init__(self, country_name, country_code):
    self.country_name=country_name
    self.country_code=country_code
    Country.AVG_POPULATION=Country.AVG_POPULATION

    if type(self.country_name)==str and type(self.country_code)==str and len(self.country_
      print("Correct")
    else :
      print("Error")


  def country_short_form(self, country_name):
    return country_name[0:2].upper()

  @classmethod
  def is_densly_populated(cls,population):
    if population > cls.AVG_POPULATION:
      return 1
    else:
      return 0


  @staticmethod
  def world_avg_population(array):
    n=len(array)
    sum=0
    for i in array:
      sum+=i

    return sum/n


  @property
  def formatted_country_name(self):
    return "{}({})".format(self.country_name,self.country_code)


  @formatted_country_name.setter
  def formatted_country_name(self,formatted_country_name):
    self.country_name=formatted_country_name.split("-")[0]
    self.country_code=formatted_country_name.split("-")[1]


  @formatted_country_name.deleter
  def formatted_country_name(self):
    self.country_name='NA'
    self.country_code='NA'
```

```
        #del self.formatted_country_name


c1='india'
code1='IND'
ob1=Country(c1,code1)
print(ob1.country_short_form(c1))
ob2=Country.is_densly_populated(11)
print(ob2)
ob3=Country.world_avg_population([2,5,9,3,7])
print(ob3)
print(ob1.formatted_country_name)
ob1.formatted_country_name="America-AME"
print(ob1.formatted_country_name)
del ob1.formatted_country_name
print(ob1.formatted_country_name)
```

```
Correct
IN
1
5.2
india(IND)
America(AME)
NA(NA)
```

2)

```
class Shapes:

  def __init__(self, l, b):
    self.l=l
    self.b=b

  def type(self):
    return "({}) ({}) length and breath of shape".format(self.l,self.b)

  def area(self):
    return 1/2*self.l*self.b


class Triangle(Shapes):
  pass
  def __init__(self, l, a):
    self.l=l
    self.a=a


  def area(self):
    return 1/2*self.l*self.a

  def quad(self):
    print("This one is triangle")


class Quadrilateral(Shapes):
```

```
class Quadrilateral(Shapes):
  pass
  def __init__(self, l, s):
    self.l=l
    self.s=s


  def area(self):
    return self.l*self.s

  def quad(self):
    print("This one is square")

class trichild(Triangle):
  pass
  def __init__(self, l, c, k):
    self.l=l
    self.c=c
    self.k=k

  def area(self):
    return 1/4*self.l*self.c

  def grand(self):
    print("This one is grandchild of triangle")

class quadchild(Quadrilateral):
  pass
  def __init__(self, l, c, k):
   self.l=l
   self.c=c
   self.k=k

  def area(self):
    return self.k*self.l*self.c

  def grand(self):
    print("This one is grandchild of quadrilateral")


obj1 = Shapes(4,5)
print(obj1.type())
print(obj1.area())
obj1=Triangle(5,9)
print(obj1.area())
```

```
 (4) (5) length and breath of shape
 10.0
 22.5
```

Exception Handling

1)Write a function that raises 5 built in python exceptions without using the raise key word and print appropriate messages for each exception.

```
#valueError
try:
    print (float('nihal'))
except ValueError:
    print ('ValueError: could not convert string to float: \'DataCamp\'')
else:
    print ('Success, no error!')
```

```
 ValueError: could not convert string to float: 'DataCamp'
```

```
#exception EOFError (End-of-file)
count=0a
while True:
    data = input('Enter name : ')
    print ('Hello  ', data)
    count+=1
    if count==4:
      print("End-of-file error")
      break
```

```
 Enter name : nihal
 Hello   nihal
 Enter name : ankur
 Hello   ankur
 Enter name : mit
 Hello   mit
 Enter name : atul
 Hello   atul
 End-of-file error
```

```
#exception ImportError
import module_does_not_exist
```

```
#exception IndexError
array = [ 0, 1, 2 ]
print(arra[3])
```

```
 ---------------------------------------------------------------------------
 IndexError                                Traceback (most recent call last)
 <ipython-input-27-163b36a27854> in <module>()
       1 #exception IndexError
       2 array = [ 0, 1, 2 ]
 ----> 3 print (array[3])

 IndexError: list index out of range
```

SEARCH STACK OVERFLOW

```
#exception KeyError
array = { 'a':1, 'b':2 }
print (array['c'])
```

2)

```
import re
regex = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
persons=[]
class exception_1:
  def __init__(self, name):
    self.name=name

  def name_func(self):
    if not all(x.isalpha or x=="" for x in self.name):
      print("Invalid name :")



class exception_2(exception_1):
  def __init__(self, age):
    self.age=age

  def age_func(self):
    if type(self.age)!=int:
      print("Invalid age :")
    else:
      print("age = {}".format(self.age))

class exception_3(exception_2):
  def __init__(self, email):
    self.email=email

  def email_func(self):
    if(re.fullmatch(regex, self.email)):
        print("Valid Email")

    else:
        print("Invalid Email")



def monthcheck(month):
    return 0 < month <= 12

monthcheck(4)
```

```
 True
```

Regular Expressions

```
import re

def Datematch( strdate):

  all = re.findall(r"(January|February|March|April|May|June|July|August|September|October|
  if len(all) == 0:
    return 0
  return 1

s = "January 02 2005"
Datematch(s)
```

```
1
```

```
import re

def validate_credit_cards(credit_cards):
    valid_structure = r"[349]\d{3}(-?\d{4}){3}$"
    no_four_repeats = r"((\d)-?(?!(-?\2){1})){16}"
    filters = valid_structure, no_four_repeats

    for cc in credit_cards:
        if all(re.match(f, cc) for f in filters):
            return true
        else:
            return false


credit_cards = ['2536258796157802','4253625879615786',
          '4424424424442444', '5122-2368-7954-3214',
          '4424444424442444']
validate_credit_cards(credit_cards)
```

Decorators

```
from functools import wraps
from time import time


def timing(f):
    @wraps(f)
    def wrapper(*args, **kwargs):
        start = time()
        result = f(*args, **kwargs)
        end = time()
        print ('Elapsed time: {}'.format(end-start))
        return result
    return wrapper
```

```
@timing
def f(a):
    for _ in range(a):
        pass


print(f(20000))
```

```
     Elapsed time: 0.0007638931274414062
     None
```

Modules

1

```
import math

a = math.pi / 6

print ("sin function = ", end ="")
print (math.sin(a))
```

```
     sin function = 0.49999999999999994
```

2

```
 '''modue name City

  def road():
  print("Road looks good")

def market():
  print("Everything is availabe here!")


def doctor():
  print("There are many doctors in this city")'''
```

```
     Road looks good
```

```
'''above code is the module that I have used here '''

import City as c

ans = c.road()

print(c)
```

Packages

1

```
import numpy as np

a = np.zeros(3)

print(type(a[0]))
print(a)

b = np.ones(5)
print(b)

c = np.array([5,6])
print(c)
```

```
<class 'numpy.float64'>
[0. 0. 0.]
[1. 1. 1. 1. 1.]
[5 6]
```

2

```
'''All three modules are stored in Animals file which we call package'''

'''Animals'''

'''module1.py'''
def city():
  print("Nice city")

def population():
  print("Huge population")


'''module2.py'''
def dog():
  print("I found Shiba Inu dog")

def population_dog():
  print("But their population are low")


'''module3.py'''
def cat():
  print("Cats are evil")
```

```
def population_cat():
  print("Also their population are high")
```

File Handeling