DSO-599 Practice Exam I (with Solutions)

Part I. Multiple Choice (1 point each)

1. Predict the result of the following expression.

```
str(int('4'+'3')+2)+'1'
```

A) '721'
B) '73'
C) '433'
D) '451'
E) '4321'

```
[1]: str(int('4'+'3')+2)+'1'
```

```
'451'
```

2. Predict the output of the following code.

```
a=3
b=5
c=1
if a>b:
    c=a+b+c
elif a<b:
    c=a-b-c
else:
    c=a*2
print(c)
```

A) 9
B) 6
C) 5
D) 1
E) -3

```
[2]: a=3
     b=5
     c=1
     if a>b:
         c=a+b+c
     elif a<b:
         c=a-b-c
     else:
         c=a*2
     print(c)
```

```
-3
```

3. Predict the output of the following code.

```
def a(x):
    return 2*x
def b(x):
    return x+1
x=0
x=a(b(x))
x=b(a(x))
print(x)
```

A) 5
B) 4
C) 3
D) 2
E) 0

```
[3]: def a(x):
         return 2*x
     def b(x):
         return x+1
     x=0
     x=a(b(x))
     x=b(a(x))
     print(x)
```

```
5
```

4. Predict the output of the following code

```
s=0
for i in range(5):
    s+=i
print(s)
```

A) 0
B) 4
C) 5
D) 10
E) 15

```
[4]: s=0
     for i in range(5):
         s+=i
     print(s)
```

```
10
```

5. Predict the output of the following code

```
l=[1,4,2,5,1]
x=0
for i in l:
    if i>2:
        x=i*2
print(x)
```

A) 13
B) 11
C) 10
D) 8
E) 2

```
[5]: l=[1,4,2,5,1]
     x=0
     for i in l:
         if i>2:
             x=i*2
     print(x)
```

```
10
```

6. Predict the output of the following code

```
import pandas as pd
df1=pd.DataFrame([[1,2],[3,4]])
df2=pd.DataFrame([[5,6],[7,8]])
print(pd.concat([df1,df2]).iloc[2,0])
```

A) 2
B) 3
C) 5
D) 7
E) 10

```
[6]: import pandas as pd
     df1=pd.DataFrame([[1,2],[3,4]])
     df2=pd.DataFrame([[5,6],[7,8]])
     print(pd.concat([df1,df2]).iloc[2,0])
```

```
5
```

7. Predict the result of the last line of the following code

```
import pandas as pd
s=pd.Series({'apple':5,'orange':6,'banana':7,'grape':8})
(s>=7).sum()
```

A) 2
B) 8
C) 15
D) `'grape'`
E) `'bananagrape'`

```
[7]: import pandas as pd
     s=pd.Series({'apple':5,'orange':6,'banana':7,'grape':8})
     print( (s>=7).sum() )
```

```
2
```

8. Predict the result of the last line of the following code

```
import pandas as pd
df=pd.DataFrame([[1,3],[1,5],[2,3]],columns=['A','B'])
df.groupby('A')['B'].mean()[1]
```

```
[8]: import pandas as pd
     df=pd.DataFrame([[1,3],[1,5],[2,3]],columns=['A','B'])
     df.groupby('A')['B'].mean()[1]
```

```
4
```

9. Which of the following lines has a syntax error?

A) `a={5:[1,3],6:{2:4},'9':2}`
B) `{6:2,5:1}[6]+3`
C) `len([3,5]+[7])`
D) `[3,5,2](0)+2`
E) `('56'+str(2+5*5)).strip()`

```
[9]: # The following line has error because the list [3,5,2] is not a function.
     [3,5,2](0)+2
```

10. Which of the following lines has a syntax error?

Assume the following code is executed first:

```
import pandas as pd
df=pd.DataFrame([['Apple',5,10],['Orange',6,5]],columns=['fruit','length','taste'])
```

A) `df.loc['Apple','length']*2`
B) `df['length'].plot()`
C) `(df['taste']*2).sum()`
D) `df['fruit'][0]+' is great!'`
E) `df.iloc[df.iloc[1,1]-df.iloc[0,1],:]`

```
[ ]: import pandas as pd
     df=pd.DataFrame([['Apple',5,10],['Orange',6,5]],columns=['fruit','length','taste'])
     # The following line has an error because 'Apple' is a value in column 0, and not a row
     df.loc['Apple','length']*2
```

Part II. Writing Code

11. Combining Datasets (3 points)

Given the following DataFrames “income” and “debt”, write code to create a DataFrame called “combined” as follows. (Note: your code must work given whatever data in “income” and “debt”; in other words, you will not obtain points if you hard-code the answer.)

DataFrame called “income”:

	Name	Income
0	Alice	10000
1	Bob	8000
2	Charlie	12000

DataFrame called “debt”:

	first_name	debt
0	Eli	50000
1	Alice	0
2	Charlie	10000

After running your code, the DataFrame “combined” should be:

	first_name	debt	Name	Income
0	Eli	50000	NaN	NaN
1	Alice	0	Alice	10000.0
2	Charlie	10000	Charlie	12000.0

Complete the following code to make the DataFrame called “combined”:

```
import pandas as pd
income=pd.DataFrame([['Alice',10000],['Bob',8000],['Charlie',12000]]\
                    ,columns=['Name','Income'])
debt=pd.DataFrame([['Eli',50000],['Alice',0],['Charlie',10000]]\
                  ,columns=['first_name','debt'])
```

```
[17]: import pandas as pd
      income=pd.DataFrame([['Alice',10000],['Bob',8000],['Charlie',12000]]\
                          ,columns=['Name','Income'])
      debt=pd.DataFrame([['Eli',50000],['Alice',0],['Charlie',10000]]\
                        ,columns=['first_name','debt'])
      combined=debt.merge(income,left_on='first_name',right_on='Name',how='left')
```

12. Manipulating Dictionaries (6 points)

Write a function called “sumDomains” with one input:

```
- dic: a dictionary in which the keys are strings and the values are numbers.
```

The function should return the sum of the values for which the key ends with “.com”, “.edu”. (Case insensitive)

Example 1:
Running

```
sumDomains({'apple.com':10, 'google':5, 'USC.EDU':3})
```

yields the output:

```
13
```

Example 2:
Running

```
sumDomains({'a':5,'b':6,'c':3,'happy.org':5})
```

yields the output:

```
0
```

Example 3:
Running

```
sumDomains({'usc.edu':3,'ucla.edu':-1,'Google.COM':2})
```

yields the output:

```
4
```

```
[11]: def sumDomains(dic):
          s=0
          for key in dic:
              if key.lower().endswith('.edu') or key.lower().endswith('.com'):
                  s+=dic[key]
          return s
```

```
[12]: sumDomains({'apple.com':10, 'google':5, 'USC.EDU':3})
```

```
13
```

```
[13]: sumDomains({'a':5,'b':6,'c':3,'happy.org':5})
```

```
0
```

```
[14]: sumDomains({'usc.edu':3,'ucla.edu':-1,'Google.COM':2})
```

```
4
```

14. Computing Course Grade (6 points)

Write a program to prompt for a score between 0.0 and 1.0, using the prompt string “Enter score:”. If the score is out of range, print “Bad score”. If the score is between 0.0 and 1.0 (inclusive), print a grade using the following table:

Score	Grade
>= 0.9	A
>= 0.8	B
>= 0.7	C
>= 0.6	D
< 0.6	F

The program should terminate after asking user for one input (you do not need to make the program repeat forever). Moreover, you can assume that the user will only input valid numbers. Below are four sample runs.

Sample run 1:

```
Enter score: 0.95
A
```

Sample run 2:

```
Enter score: 10
Bad score
```

Sample run 3:

```
Enter score: 0
F
```

Sample run 4:

```
Enter score: -0.5
Bad score
```

```
[15]: score=float(input('Enter score: '))
      if score>1 or score<0:
          output='Bad score'
      elif score>=0.9:
          output='A'
      elif score>=0.8:
          output='B'
      elif score>=0.7:
          output='C'
      elif score>=0.6:
          output='D'
      else:
          output='F'
      print(output)
```

```
Enter score: 1
A
```