XII – IP (2024-25)
Cycle Test - I key

SECTION - A

1. To create an empty Series object, you can use:

a. pd.Series(empty) b. pd.Series(np.NaN)

c. ***pd.Series()*** d. all of these

2. To specify datatype int16 for a Series object, you can write:

a. pd.Series(data = array, dtype = int16)

b. pd.Series(data = array, dtype = numpy.int16)

c. pd.Series(data = array, dtype = pandas.int16)

d. all of the above

3. Missing data in Pandas object is represented through:

a. Null b. None c. Missing d.***NaN***

4. Given a Pandas series called Sequences, the command which will display the first 4 rows is _____

a. print(Sequences.head(4))

b. print(Sequences.Head(4))

c. print(Sequences.heads(4))

d. print(Sequences.Heads(4))

5. To get the transpose of a dataframe D1, you can write _____.

a. ***D1.T*** b. D1.Transpose

c. D1.Swap d. All of these

6. To check if the Series object contains NaN values, _____ attribute is displayed.

a. ***hasnans*** b. nbytes

c. ndim d. dtype

7. To delete a row from a DataFrame, you may use _____ statement.

a. remove b. del ***c.drop*** d. cancel

8. To display the 3rd, 4th and 5th columns from the 6th to 9th rows of a dataframe DF, you can write _____.

a. DF.loc[6:9,3:5] b.DF.loc[6:10,3:6]

c. ***DF.iloc[6:10,3:6]*** d. DF.iloc[6:9,3:5]

9. The axis 1 identifies a dataframe's _____

a. rows **b.** ***columns*** *c.* values d.datatype

SECTION – B

10. Write a program to create a Series object using the Python sequence [1101, 1301, 1501, 1701, 1901].

Assume that Pandas is imported as alias name (your name).

```
import pandas as mohan
ser = mohan.Series([1101,1301,1501,1701,1901])
print(ser)
```

11. Write a program to create a Series object using individual characters 'h', 'e', 'l', 'l', 'o'.

Assume that Pandas is imported as alias name pd.

```
import pandas as pd
ser1 = pd.Series(['h','e','l','l','o'])
print(ser1)
```

OR

Write a program to create a Series object using a string: 'very good'.

Assume that Pandas is imported as alias name pd.

```
import pandas as pd
ser2 = pd.Series(['very good'])
print(ser2)
```

12. Total number of medals to be won is 200 in the Inter University games held every alternate year. Write code to create a Series object that stores these medals for games to be held in the decade 2020 – 2029.

```
import pandas as pd
ser3 = pd.Series(200,index=range(2020,2029,2))
print(ser3)
```

13. Write a program to create a Series object using an ndarray that has 5 elements in the range 24 to 64.

```
import pandas as pd
import numpy as np
ser4 = pd.Series(np.linspace(24,64,5))
print(ser4)
```

SECTION – C

14. Write a program to create a DataFrame from a 2D array as shown below:

101	113	124
130	140	200
115	216	217

```
ser5 = pd.DataFrame(np.array([[101,113,124],
        [130,140,200],[115,216,217]]))
print(ser5)
```

15. Write the differences between Series and DataFrames

Property	Series	DataFrame
Dimensions	**1 Dimensional**	**2-Dimensional**
Type of Data	**Homegeneous**, i.e., all the elements must be of same type in a Series object	**Heterogeneous**, i.e., a DataFrame object can have elements of different data types
Mutability	**Value mutable**, i.e., their elements value can change	**Value mutable**, i.e., their elements value can change
	Size-immutable, i.e., size of a Series object, once created, cannot change. If we want to add/drop an element, internally a new Series object will be created	**Size-mutable**,i.e., size of a Dataframe object, once created, can change in place. That is, you can add/drop elements in an existing dataframe object.

OR

Write the differences between Series and Lists

16. Write a program to create a DataFrame df as shown below:

	Population	*Hospitals*	*Schools*
Delhi	10927986	189	7916
Mumbai	12691836	208	8508
Kolkata	4631392	149	7226
Chennai	4328063	157	7617

```
import pandas as pd
import numpy as np
dic = {'population':{'delhi':10927986,
'mumbai':12691836,'kolkata':4631392,
'chennai':4328063},  'hospitals':{'delhi':189,
'mumbai':208, 'kolkata':149,'chennai':157},

'schools':{'delhi':7916,'mumbai':8508,'kolkata':7226,'ch
ennai':7617}}
df = pd.DataFrame(dic)
print(df)
```

SECTION – D

17. Consider the given Series object obj and do the following actions

```
a    101
b    102
c    103
d    104
```

a. Write the statement to display the number of bytes in the underlying data.

   Obj.nbytes

b. Write the statement to assign new name to index.

   Obj.index=[‘aa’,’bb’,’cc’,’dd’]

c. Write the statement to assign name to Series object

   Obj.name = ‘newname’

d. Write the statement to find if there are any NaN values.

Obj.hasnans

OR

Write the statement to return a tuple of the shape of the underlying data.

Obj.shape

SECTION – E

	Population	*Hospitals*	*Schools*
Delhi	10927986	189	7916
Mumbai	12691836	208	8508
Kolkata	4631392	149	7226
Chennai	4328063	157	7617

18. Write the answers for the following questions based on the above output (dataframe df)

a. Write a statement to display only Hospitals column.

Df[‘Hospitals’]

b. Write a statement to display only Mumbai row

Df.loc[‘Mumbai’,:]

c. Write a statement to display only 208 element from the above dataframe

Df.at[‘Mumbai’,’Hospitals’]

d. Write the output for the following 2 statements.

i. df.iloc[0:2,1:2]

	Hospitals
delhi	**189**
mumbai	**208**

ii. df.iloc[:,:]

	Population	**Hospitals**	**Schools**
delhi	**10927986**	**189**	**7916**
mumbai	**12691836**	**208**	**8508**
kolkata	**4631392**	**149**	**7226**
chennai	**4328063**	**157**	**7617**

OR

Write the answers for the following questions based on the above output (dataframe df)

a. Write a statement to add a column **libraries** with data [129, 345, 456, 567]

Df[‘libraries]= [129, 345, 456, 567]

b. Write a statement to add one more row Hyderabad with data [1408973,474, 587, 245]

Df.loc[‘Hyderabad’,:] = [1408973,474, 587, 245]

c. Write a statement to change the value 208 to 118.

Df.at[‘Mumbai’,’Hospitals’]=118

d. Write the python statement to change the values of column schools to [111,145.178,547]

Df.loc[:,’Schools’]=[111,145,178,547]

e. Write the python statement to change the values of row Kolkata to [4441392, 150, 750, 350]

Df.loc[‘Kolkata’]=[4441392, 150, 750, 350]