**Single-Choice Questions**

1. Which method is used to group a DataFrame by multiple columns?

A) group\_by()

B) groupby()

C) multi\_group()

D) group()

2. Which function is used to concatenate two DataFrames vertically?

A) pd.concat()

B) pd.append()

C) pd.join()

D) pd.merge()

3. How do you convert a string-type date to a pandas Datetime type?

A) pd.to\_datetime()

B) pd.strptime()

C) pd.to\_time()

D) pd.parse\_date()

**Code Complete**

**1.** Fill in the blank to create a pandas DataFrame with a DateTimeIndex.

import pandas as pd

dates = pd.date\_range('20230101', periods=6)

df = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

print(df)

Output:

Values

2023-01-01 1

2023-01-02 2

2023-01-03 3

2023-01-04 4

2023-01-05 5

2023-01-06 6

Answer:

df = pd.DataFrame({'Values': range(1, 7)}, index=dates)

**Finding The Error**

1. Line \_\_\_\_\_\_\_\_ has an error, the correct code should be \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Hint: The code attempts to convert a string to a pandas Datetime object.

1 import pandas as pd

2 date\_string = '2024-01-01'

3 datetime\_object = pd.to\_date(date\_string)

4 print(datetime\_object)

Expected Output:

2024-01-01 00:00:00

Answer:

Error: Line 3 has an error.

Correct Code: datetime\_object = pd.to\_datetime(date\_string)

**Multiple-Choice Questions**

1. Which of the following are valid ways to create a pandas DataFrame? ()

A) pd.DataFrame()

B) pd.DataFrame({'Column1': [1, 2], 'Column2': [3, 4]})

C) pd.DataFrame([[1, 2], [3, 4]], columns=['Column1', 'Column2'])

D) pd.DataFrame({'Column1': [1, 2], 'Column2': [3, 4]}, index=[1, 2])

2. Which of the following are valid methods to group a DataFrame by a single column? ()

A) group\_by()

B) groupby()

C) by()

D) pivot()

3. Which of the following are valid functions to combine two DataFrames horizontally? ()

A) pd.concat()

B) pd.append()

C) pd.merge()

D) pd.join()

**Programming Questions**

Write a Python script to accomplish the following tasks:

1. Import the pandas library.
2. Create a DataFrame with employee information, including employee ID ('EmployeeID'), department ('Department'), and salary ('Salary').
3. Group the DataFrame by department and calculate the average salary for each department.
4. Reshape the results into a new DataFrame that includes the department and the average salary.

**Employee Information:**

data = {

'EmployeeID': [1, 2, 3, 4, 5],

'Department': ['Sales', 'Marketing', 'Sales', 'IT', 'Marketing'],

'Salary': [50000, 60000, 55000, 70000, 65000]

}

import pandas as pd

data = {

'EmployeeID': [1, 2, 3, 4, 5],

'Department': ['Sales', 'Marketing', 'Sales', 'IT', 'Marketing'],

'Salary': [50000, 60000, 55000, 70000, 65000]

}

df = pd.DataFrame(data)

grouped\_salaries = df.groupby('Department')['Salary'].mean()

average\_salary\_df = grouped\_salaries.reset\_index(name='AverageSalary')

print(average\_salary\_df)

