PYTHON FOR DATA SCIENCE

MCQ :

1) What will be the output of the following code snippet?

```
a = "10"
b = float(a) + 5
result = str(b) + "123"
print(type(result))
```

- ○ <class 'float'>
- ◉ <class 'str'>
- ○ <class 'int'>
- ○ The code will give an error.

2) How can you concatenate the strings “data” and “science” with a hyphen(-) between them?

- ☐ “data”.join(“science”)
- ☑ “-”.join([“data”, “science”])
- ☑ “data” + “-” + “science”
- ☐ None of the above.

DESCRIPTIVE QUES:

1) Explain the difference between a list, tuple, set, and dictionary in Python.
   -Mention whether they are ordered/unordered, mutable/immutable, and whether they allow duplicate values.
   -Give one short code example of each.

   ANS- **List**:

- Ordered, mutable, allows duplicates.
- Example:
- my_list = [1, 2, 2, 3]
- print(my_list) # [1, 2, 2, 3]

Tuple:

- Ordered, immutable, allows duplicates.
- Example:
- my_tuple = (1, 2, 2, 3)
- print(my_tuple) # (1, 2, 2, 3)

Set:

- Unordered, mutable, does **not** allow duplicates.
- Example:
- my_set = {1, 2, 2, 3}
- print(my_set) # {1, 2, 3}

Dictionary:

- Unordered (in Python 3.6+, insertion order is preserved), mutable, stores **key-value pairs**.
- Keys must be unique; values can be duplicated.
- Example:
- my_dict = {"a": 1, "b": 2, "c": 2}

  print(my_dict) # {'a': 1, 'b': 2, 'c': 2}

2) You are given a dataset of cars in a Pandas DataFrame with columns: Car Name, Brand, Type, and Price.

   Write the Python code to:

a. Extract only the Type column as a separate DataFrame.
b. Fill missing values in the Type column with the most frequent category.
c. Display the summary statistics of only the numeric column(s).

ANS- import pandas as pd

```
# Sample DataFrame
df_cars = pd.DataFrame({
    'Car Name': ['CarA', 'CarB', 'CarC'],
    'Brand': ['BrandX', 'BrandY', 'BrandZ'],
    'Type': ['SUV', None, 'Sedan'],
    'Price': [10.5, 12.3, 15.8]
})

# 1. Extract only the 'Type' column as a separate DataFrame
type_df = df_cars[['Type']]
print("Type column as DataFrame:\n", type_df)

# 2. Fill missing values in 'Type' column with most frequent category (mode)
df_cars['Type'].fillna(df_cars['Type'].mode()[0], inplace=True)
print("\nAfter filling missing values:\n", df_cars)
```

```
# 3. Display summary statistics of only numeric columns

print("\nSummary statistics of numeric columns:\n",
df_cars.describe())
```