Q1.Declare and print variables of different data types

```
# Integer
age = 25
print("Age:", age)

# Float
height = 5.9
print("Height:", height)

# String
name = "Alice"
print("Name:", name)

# Boolean
is_student = True
print("Is Student:", is_student)

# List
marks = [85, 90, 95]
print("Marks:", marks)

# Tuple
coordinates = (10, 20)
print("Coordinates:", coordinates)

# Dictionary
student = {"name": "Alice", "age": 25, "course": "Data Science"}
print("Student Info:", student)

# Set
unique_ids = {101, 102, 103}
print("Unique IDs:", unique_ids)
```

Q2.Write a program using if-else and elif

```
num = int(input("Enter a number: "))

if num > 0:
    print("The number is positive.")
elif num == 0:
    print("The number is zero.")
else:
    print("The number is negative.")
```

Q1. Demonstrate type conversion and evaluate expressions

```
# Type Conversion
num_str = "10"
num_int = int(num_str)  # converting string to integer

print("String:", num_str)
print("Converted to Integer:", num_int)

# Evaluate Expressions
a = 5
b = 3
result = (a + b) * 2

print("Expression Result:", result)
```

```
# Converting float to int
pi = 3.14
rounded = int(pi)
print("Float:", pi)
print("Converted to Integer:", rounded)

# Convert number to string
num = 100
num_text = str(num)
print("Number as Text:", num_text)
```

Q2. Use a for loop to print even numbers from 1–50

```
# Program to print even numbers from 1 to 50

print("Even numbers from 1 to 50 are:")
for i in range(1, 51):
    if i % 2 == 0:
        print(i, end=" ")
```

Q1. Perform basic operations on lists (add, remove, sort)

```
# Create a list of numbers
numbers = ['A','B','C','D','E']

# Add an element to the list
numbers.append('F')
print("After adding F:", numbers)

# Remove an element from the list
numbers.remove('B')
print("After removing B:", numbers)

# Sort the list in ascending order
numbers.sort()
print("After sorting:", numbers)
```

Q2. Write a Python program that defines a user-defined function which accepts parameters, performs a specific task, and returns a result. For example, create a function that takes the radius of a circle as a parameter and returns its area.

```
# Import the math module to use pi
import math

# Function to calculate area of a circle
def circle_area(radius):
    area = math.pi * radius ** 2
    return area

# Take input from the user
r = float(input("Enter the radius of the circle: "))

# Call the function and display the result
result = circle_area(r)
print("The area of the circle is:", result)
```

Q1. **Import and use math, random, and datetime modules**

**i. ii. iii.**

**math: Perform operations such as calculating the square root of a number, finding the cosine of a value (in radians), and displaying the value of π.**

**random: Generate a random integer within a specified range (e.g., 1 to 100) and select a random item from a list.**

**datetime: Display the current date and time, extract the current year, and format the date in dd-mm-yyyy hh:mm:ss format.**

```
# Import modules
import math
import random
from datetime import datetime

# --- Math Module ---
num = 36
print("Square root of", num, "is", math.sqrt(num))
print("Value of pi is:", math.pi)

# --- Random Module ---
print("Random integer (1-100):", random.randint(1, 100))
print("Random choice from list:", random.choice(['apple', 'banana', 'cherry']))

# --- Datetime Module ---
now = datetime.now()
print("Current date and time:", now)
print("Current year:", now.year)
```

Q2. Display basic info using .head(), .describe(), .info()

```
import pandas as pd

# Create a sample DataFrame
data = {
    "Name": ["Raj", "Sanjay", "Ramesh", "Pooja", "Parth"],
    "Age": [25, 30, 35, 40, 28],
    "Salary": [50000, 60000, 70000, 80000, 55000]
}
df = pd.DataFrame(data)

# Display first few rows
print("First 5 rows:")
print(df.head())

# Display summary statistics
print("\nSummary statistics:")
print(df.describe())

# Display basic info about the DataFrame
print("\nDataFrame info:")
print(df.info())
```

Q1. Read data from a CSV/Excel file using Pandas

```
import pandas as pd

# Read data from the CSV file
data = pd.read_csv('data.csv')

# Display the data
print("Data from CSV file:\n")
print(data)
```

Q2. **Remove duplicate rows from a DataFrame**

```
import pandas as pd

# Read the data
data = pd.read_csv('data.csv')
print("Original Data:\n", data)

# Remove duplicate rows
data_no_duplicates = data.drop_duplicates()
print("\nData after removing duplicate rows:\n", data_no_duplicates)
```

Q1. Calculate descriptive statistics (mean, median, mode)

```python
import pandas as pd

# Sample dataset
data = {
    "Marks": [85, 90, 78, 92, 85, 78, 95, 90, 85]
}

df = pd.DataFrame(data)

# Calculate Mean
mean_value = df["Marks"].mean()
print("Mean:", mean_value)

# Calculate Median
median_value = df["Marks"].median()
print("Median:", median_value)

# Calculate Mode
mode_value = df["Marks"].mode()[0]   # mode() returns a Series
print("Mode:", mode_value)
```

Q2. Use Seaborn for a heatmap or pair plot (basic usage)

```python
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# ---- 1. Heatmap Example ----
# Create random data
data = np.random.rand(6, 6)
df = pd.DataFrame(data, columns=list("ABCDEF"))

plt.figure(figsize=(6, 4))
sns.heatmap(df, annot=True, cmap="coolwarm")
plt.title("Heatmap Example")
plt.show()

# ---- 2. Pair Plot Example ----
# Load built-in dataset (Iris)
iris = sns.load_dataset("iris")

# Create pair plot
sns.pairplot(iris, hue="species")
plt.suptitle("Pair Plot Example", y=1.02)
plt.show()
```

Q1. Demonstrate type conversion and evaluate expressions

```
# Type Conversion
num_str = "10"
num_int = int(num_str)  # converting string to integer

print("String:", num_str)
print("Converted to Integer:", num_int)

# Evaluate Expressions
a = 5
b = 3
result = (a + b) * 2

print("Expression Result:", result)
```

```
# Converting float to int
pi = 3.14
rounded = int(pi)
print("Float:", pi)
print("Converted to Integer:", rounded)

# Convert number to string
num = 100
num_text = str(num)
print("Number as Text:", num_text)
```

Q2. Use Seaborn for a heatmap or pair plot (basic usage)

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# ---- 1. Heatmap Example ----
# Create random data
data = np.random.rand(6, 6)
df = pd.DataFrame(data, columns=list("ABCDEF"))

plt.figure(figsize=(6, 4))
sns.heatmap(df, annot=True, cmap="coolwarm")
plt.title("Heatmap Example")
plt.show()

# ---- 2. Pair Plot Example ----
# Load built-in dataset (Iris)
iris = sns.load_dataset("iris")

# Create pair plot
sns.pairplot(iris, hue="species")
plt.suptitle("Pair Plot Example", y=1.02)
plt.show()
```

Q1. Write a Python program that defines a user-defined function which accepts parameters, performs a specific task, and returns a result. For example, create a function that takes the radius of a circle as a parameter and returns its area.

```
# Import the math module to use pi
import math

# Function to calculate area of a circle
def circle_area(radius):
    area = math.pi * radius ** 2
    return area

# Take input from the user
r = float(input("Enter the radius of the circle: "))

# Call the function and display the result
result = circle_area(r)
print("The area of the circle is:", result)
```

Q2. **Create a DataFrame from a dictionary**

```
import pandas as pd

# Create a dictionary
data = {
    "Name": ["Raj", "Sanjay", "Pooja"],
    "Age": [25, 30, 35],
    "City": ["Mumbai", "Delhi", "Bangalore"]
}

# Create a DataFrame from the dictionary
df = pd.DataFrame(data)

# Display the DataFrame
print("Pandas DataFrame:")
print(df)
```

Q1. Generate random numbers and calculate square roots

```
import random
import math

# Generate 5 random numbers between 1 and 100
for i in range(5):
    num = random.randint(1, 100)
    sqrt_num = math.sqrt(num)
    print("Number:", num, "-> Square root:", round(sqrt_num, 2))
```

Q2. Handle missing data using dropna() and fillna().

```
import pandas as pd

# Read the data
data = pd.read_csv('data.csv')
print("Original Data:\n", data)

# Handle missing data

# a) Remove rows with missing values
data_dropna = data.dropna()
print("\nAfter using dropna():\n", data_dropna)

# b) Fill missing values with specific values
data_fillna = data.fillna({'Age': 0, 'City': 'Unknown'})
print("\nAfter using fillna():\n", data_fillna)
```

Q1. Use .corr() to check correlation

```
import pandas as pd

# Sample dataset
data = {
    "PDS":   [85, 78, 90, 95, 88, 76, 92],
    "OS": [80, 75, 88, 92, 85, 78, 90],
    "CJ":  [60, 65, 70, 72, 68, 66, 69]
}

df = pd.DataFrame(data)

# Correlation matrix
correlation_matrix = df.corr()

print("Correlation Matrix:\n")
print(correlation_matrix)
```

Q2. Create bar, line, and pie charts using Matplotlib

```
import matplotlib.pyplot as plt

# Sample data
x = ['A', 'B', 'C', 'D', 'E']
y = [10, 24, 36, 40, 20]

# 1. Bar Chart
plt.bar(x, y)
plt.title("Bar Chart Example")
plt.xlabel("Categories")
plt.ylabel("Values")
plt.show()

# 2. Line Chart
plt.plot(x, y, marker='o')
plt.title("Line Chart Example")
plt.xlabel("Categories")
plt.ylabel("Values")
plt.show()

# 3. Pie Chart
plt.pie(y, labels=x, autopct='%1.1f%%')
plt.title("Pie Chart Example")
plt.show()
```

Q1. Create a simple NumPy array and perform arithmetic operations

```
import numpy as np

# Create a NumPy array
arr = np.array([2, 4, 6, 8, 10])
print("Original array:", arr)

# Arithmetic operations
print("Add 5:", arr + 5)
print("Subtract 2:", arr - 2)
print("Multiply by 3:", arr * 3)
print("Divide by 2:", arr / 2)
```

Q2. Plot histograms and scatter plots

```
import matplotlib.pyplot as plt
import numpy as np

# Sample data
data = np.random.randn(1000)  # random numbers (normal distribution)
x = np.random.rand(50)        # random x values
y = np.random.rand(50)        # random y values

# 1. Histogram
plt.hist(data, bins=30, edgecolor='black')
plt.title("Histogram Example")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.show()

# 2. Scatter Plot
plt.scatter(x, y, color='blue', marker='o')
plt.title("Scatter Plot Example")
plt.xlabel("X values")
plt.ylabel("Y values")
plt.show()
```