NITTE EDUCATION TRUST

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

An Autonomous Institution Approved by UGC/AICTE/Govt. of Karnataka
Accredited by NBA (Tier – I) and NAAC 'A+' Grade
Affiliated to Visveswaraya Technological University, Belagavi
Post Box No. 6429, Yelahanka, Bengaluru – 560 064, Karnataka, India

KNOWLEDGE • CHARACTER • UNITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

22CSE554- Data Science using Python

Practice Questions for MSE1

Course Coordinator: Dr.Vani V

Portion: Unit 1, 2 and Unit 3 - first half

Part A (2 Marks)

1. What is data, and why is it important in data science? (BL1)
2. List two basic data types in Python and provide an example of each. (BL1)
3. How do variables work in Python, and why are they used in data analysis? (BL2)
4. Describe the purpose of an if statement in Python. Provide a simple example. (BL2)
5. Explain how NumPy handles numerical operations on arrays more efficiently than Python lists. (BL2)
6. What is the purpose of the Pandas library in Python? Provide an example of how it is used for data manipulation. (BL2)
7. How does the DataFrame structure in Pandas differ from a Python dictionary? (BL2)
8. Describe the purpose of a NumPy array and provide an example of creating a 1D array. (BL1)
9. What is data processing, and why is it essential in data analysis? (BL1)
10. List two common techniques used in data cleaning to handle missing values in a DataFrame. (BL1)
11. How does slicing differ from indexing when working with Pandas DataFrames? Provide a brief explanation. (BL2)
12. Explain the use of the dropna() function in Pandas for handling missing data. (BL2)
13. What is the purpose of the .loc[] and .iloc[] functions in Pandas? (BL1)
14. How does the fillna() function help in dealing with missing values in a DataFrame? Provide a short explanation. (BL2)
15. What is the role of the to_csv() method in Pandas, and how is it used to save a DataFrame to a CSV file? (BL2)
16. What is the purpose of using Matplotlib in Python, and how does it help in data visualization? (BL1)
17. List two key differences between a line chart and a bar plot in Matplotlib. (BL1)
18. Explain how a scatter plot in Matplotlib helps in understanding the relationship between two variables. (BL2)
19. How can a box plot in Matplotlib be used to identify outliers in a dataset? Provide a brief explanation. (BL2)
20. Describe the use of a heatmap in Matplotlib for visualizing correlation between variables. (BL2)

Part B (5 marks)

1. Using Pandas, load a CSV file containing information about employees (columns: 'Name', 'Department', 'Salary'). Filter and display the names of employees with salaries above 50,000 in the 'HR' department. (BL3)

22CSE554- Data Science using Python

Practice Questions for MSE1

Course Coordinator: Dr.Vani V

2. Create a NumPy array of random integers between 1 and 100 with a size of 15. Use slicing to extract elements greater than 50 and calculate their mean. (BL3)
3. Using a DataFrame in Pandas, group sales data by 'Region' and calculate the total sales for each region. Assume a DataFrame with columns: 'Region' and 'Sales' with 5 sample rows. (BL3)
4. Create a Pandas DataFrame with columns 'Product', 'Sales_Q1', 'Sales_Q2', 'Sales_Q3', 'Sales_Q4'. Calculate the total sales for each product and add it as a new column 'Total_Sales'. (BL3)
5. Generate a 1D NumPy array with 20 elements and use element-wise operations to multiply each element by 2. Print the original and modified arrays. (BL3)
6. Using a Pandas DataFrame containing employee data (columns: 'Name', 'Gender', 'Age', 'Department'), filter out all employees who are under 30 and belong to the 'CSE' department. Display the names and ages of these employees. (BL3)
7. Create a DataFrame with sales data for a store, including columns 'Product', 'Price', and 'Quantity'. Calculate the total revenue for each product (Price * Quantity) and add it as a new column named 'Total_Revenue'. (BL3)
8. Using a Pandas DataFrame containing weather data (columns: 'City', 'Temperature', 'Humidity', 'Wind Speed'), find the average temperature and humidity for each city. (BL3)
9. Given a Pandas DataFrame with student data (columns: 'Student', 'Subject', 'Marks'), write a program to find the highest marks scored in each subject. Display the subject and the maximum marks. (BL3)
10. Create a DataFrame with COVID-19 data (columns: 'Country', 'Cases', 'Deaths', 'Recovered'). Calculate the mortality rate (Deaths/Cases) for each country and add it as a new column named 'Mortality Rate'. Display the countries with a mortality rate higher than 5%. (BL3)
11. Create a 2D NumPy array with values ranging from 10 to 30, reshape it into a 4x5 matrix, and compute the sum of elements in each column using np.sum(). (BL3)
12. Generate a 3x3 array of random integers between 5 and 20. Find the minimum value in each row. (BL3)
13. Create an array with 10 evenly spaced values between 0 and 5. Calculate the mean and standard deviation of these values. (BL3)
14. Create a 3x3 identity matrix. Multiply it elementwise with a 3x3 random matrix. Display the resulting matrix. (BL3)
15. Generate a 1D array of random values and filter elements that are greater than 0.5. Use comparison operators to count the number of values greater than 0.5. (BL3)
16. Create a 5x5 array of normally distributed random values. Calculate the variance of each column. (BL3)
17. Using np.array(), create a 2D array from a list of lists. Find the sum of elements in each row that are greater than 10. (BL3)

22CSE554- Data Science using Python

Practice Questions for MSE1

Course Coordinator: Dr.Vani V

18. Create an array with 15 elements. Reshape it into a 3x5 matrix and compute the median of each row. (BL3)
19. Generate a 4x4 matrix of integers ranging from 1 to 20. Check if all elements are less than 25. (BL3)
20. Create a 1D array of 20 random integers between 10 and 100 and find the 75th percentile of the array. (BL3)
21. Create a Pandas Series with custom indices ('a', 'b', 'c', 'd') and values [0.25, 0.5, 0.75, 1.0]. Extract and display the value corresponding to the index 'c'. (BL3)
22. Create a Pandas DataFrame using MultiIndex with levels ['State1', 'State2'] and years [2000, 2010]. Assign population data to the MultiIndex and unstack it to transform the MultiIndex into columns. (BL3)
23. Using a MultiIndex DataFrame created from two hierarchical columns ('year', 'visit') and three subjects ('Bob', 'Guido', 'Sue'), generate mock health data and display only the data for 'Guido'. (BL3)
24. Create a DataFrame with columns ['state', 'year', 'population'] and reset the index to convert the MultiIndex into a regular index. Display the flattened DataFrame. (BL3)
25. Generate a DataFrame with a MultiIndex of states and years. Set the MultiIndex from the flattened DataFrame and display the resulting DataFrame with hierarchical indices. (BL3)
26. Using a MultiIndex DataFrame with random health data, calculate the mean across rows and columns separately, and display the results. (BL3)
27. Concatenate three lists of integers [1, 2, 3], [4, 5, 6], and [7, 8, 9] using the Pandas. concat() function to create a single Series. Display the combined Series. (BL3)
28. Using pd.concat(), concatenate two Pandas Series with different indices and display the resulting Series. Set the axis parameter to 1 to perform column-wise concatenation. (BL3)
29. Create a DataFrame with duplicate indices and demonstrate how to handle duplicate indices by grouping the data based on the index and summing the values. (BL3)
30. Load a JSON file into a Pandas DataFrame, display the first 10 rows of the DataFrame using .head(), and display its shape and information using .shape and .info(). (BL3)
31. Read patient data from an SQLite database into a Pandas DataFrame and display the entire DataFrame. Use a SQL query to select the data from the 'diabetes' table. (BL3)
32. Create a DataFrame with hierarchical indices and use the .unstack() method to convert one level of the index into columns. Display the transformed DataFrame. (BL3)
33. Create a Pandas DataFrame with some missing values. Use the fillna() function to replace the missing values with the mean of the respective columns. Display the original and the updated DataFrame. (BL3)
34. Load a CSV file into a Pandas DataFrame. Remove rows with any missing values, and then display the first 5 rows of the cleaned DataFrame . (BL3)

22CSE554- Data Science using Python

Practice Questions for MSE1

Course Coordinator: Dr.Vani V

35. Using a sample DataFrame, demonstrate the use of slicing to select rows where the 'Age' column is greater than 30. Display only the 'Name' and 'Age' columns for the selected rows. (BL3)
36. Read a JSON file into a Pandas DataFrame and extract only the rows where the 'status' column is 'active'. Use the .loc[] function to display the selected rows. (BL3)
37. Write a Python program to read data from a SQLite database into a Pandas DataFrame. Use SQL queries to extract and display rows where the 'Salary' column is greater than 50,000. (BL3)
38. Using the .iloc[] function, select specific rows and columns from a DataFrame that contains sales data for different products. Display the first 3 rows and the first 2 columns. (BL3)
39. Load a CSV file into a Pandas DataFrame and use the groupby() function to group the data by 'Department'. Calculate and display the average salary for each department. (BL3)
40. Write a Python program to read data from a JSON file into a Pandas DataFrame and handle missing values in the 'Score' column by filling them with the median value. Display the modified DataFrame. (BL3)
41. Create a DataFrame with customer data, including 'CustomerID', 'Name', and 'Purchase Amount'. Use slicing and conditional filtering to select customers who made purchases above 1000. Display the selected customers. (BL3)
    a. Create a line chart using Matplotlib to visualize the sales trend over 12 months. Use random sales data and ensure that the chart has appropriate labels, a title, and markers for each data point. (BL3)
42. Generate a bar plot using Matplotlib to compare the number of products sold in four different regions ('North', 'South', 'East', 'West'). Use different colors for the bars and add gridlines to the plot for better readability. (BL3)
43. Using Matplotlib, create a box plot to show the distribution of test scores for a class of 30 students. The data should be randomly generated, and the plot should include labels for the x-axis, y-axis, and title. (BL3)
44. Create a scatter plot using Matplotlib to visualize the relationship between hours studied and scores obtained by students. Include a trend line in the plot to indicate the correlation between the two variables. (BL3)
45. Generate a heatmap using Matplotlib to visualize the correlation matrix of a DataFrame with numerical columns ('Temperature', 'Humidity', 'Wind Speed'). Ensure the heatmap includes annotations for the correlation values. (BL3)
46. Using Matplotlib, create a 3D scatter plot to represent three variables: 'Age', 'Height', and 'Weight' for 50 randomly generated data points. Label the axes and add a legend to indicate the variable names. (BL3)
47. Create a styled bar plot in Matplotlib to display the total revenue generated by five different products. Use a custom color palette, set a title for the plot, and add value labels above each bar. (BL3)

22CSE554- Data Science using Python

Practice Questions for MSE1

Course Coordinator: Dr.Vani V

48. Write a Python program to create a line chart in Matplotlib to show the daily temperature variations over a week. Use different line styles (e.g., dashed, solid) for weekdays and weekends. Add a legend to the plot. (BL3)
49. Create a box plot in Matplotlib to compare the distribution of scores across three subjects ('AIML', 'OOP', 'DSP') for a group of 30 students. (BL3)
50. Using Matplotlib, generate a heatmap to represent the confusion matrix of a classification model with four classes. Ensure the heatmap includes a color bar and annotations for the matrix values. (BL3)