PART 1: Python

1. What is file handling in Python? Name different file modes.

File handling in Python is used to store data permanently in files and to perform operations like reading, writing, appending, and updating files.

Python uses the open() function to work with files.
Common file modes:

- r – Read (default mode)
- w – Write (overwrites existing file)
- a – Append (adds data at end)
- x – Create a new file
- r+ – Read and write
- w+ – Write and read
- rb / wb – Binary modes

2. Difference between read(), readline(), and readlines()

Method	**Description**
read()	Reads the entire file as a single string
readline()	Reads one line at a time
readlines()	Reads all lines and returns them as a list

These methods are used based on how much data we want to read from a file.

3. What is exception hierarchy in Python?

Exception hierarchy is the classification of all built-in exceptions in Python in a tree structure. All exceptions are derived from the base class BaseException, and most user-handled errors come under the Exception class.
Example hierarchy:
BaseException → Exception → ArithmeticError → ZeroDivisionError
This hierarchy helps Python identify and handle different types of errors properly.

4. Difference between try-except and try-except-finally

- try-except is used to catch and handle runtime errors.
- try-except-finally includes a finally block which always executes, whether an error occurs or not.

The finally block is mainly used for closing files or releasing resources.

5. Purpose of the with statement

The with statement is used for automatic resource management. It ensures that files or connections are properly closed after execution, even if an error occurs.
This improves code readability and prevents memory leaks.
Example:

```
with open("data.txt", "r") as f:
    content = f.read()
```

6. What are iterators in Python?

Iterators are objects that allow us to traverse through a collection one element at a time. They follow the iterator protocol using __iter__() and __next__() methods.
Examples of iterable objects: list, tuple, string, set.

7. What are generators and how are they different from functions?

Generators are special functions that return values one at a time using the yield keyword instead of return.
Unlike normal functions, generators do not store all values in memory. They generate values only when needed, which makes them memory efficient.

8. What is *args and **kwargs?

- *args allows passing a variable number of positional arguments to a function.
- **kwargs allows passing a variable number of keyword arguments.

They make functions more flexible.

9. What is type casting? Give examples.

Type casting is the process of converting one data type into another data type.
Python supports both implicit and explicit type casting.
Examples:

```
int("20")
float(10)
str(5.5)
```

10. What are built-in functions in Python? Name any five.

Built-in functions are functions that are already available in Python without importing any module. They help in performing common tasks easily.
Examples:
print(), len(), type(), input(), sum()

PART 2: Matplotlib

1. What is Matplotlib and why is it used?

Matplotlib is a Python library used for data visualization. It helps in creating graphs such as line charts, bar charts, histograms, and scatter plots.
It is mainly used to understand data patterns, trends, and relationships in a visual way, which makes analysis easier.

2. Difference between line plot, bar plot, and scatter plot

Plot Type	Use
Line plot	Shows trends over time or continuous data
Bar plot	Compares values between different categories
Scatter plot	Shows relationship between two variables

Each plot is chosen based on the type of data and analysis requirement.

3. What are figure and axes in Matplotlib?
Answer:

- **Figure** is the complete window or page that contains the plot.
- **Axes** is the actual plotting area where data is drawn.

A figure can contain multiple axes (multiple plots).

4. How do labels, titles, and legends improve data visualization?
Answer:
Labels, titles, and legends make graphs easy to understand.

- Titles describe what the graph represents.
- Labels show what each axis means.
- Legends explain different data lines or colors.

They improve clarity and communication of information.

5. Why is data visualization important before building AI models?
Answer:
Data visualization helps to understand data distribution, detect missing values, outliers, and patterns.
It allows better data cleaning and feature selection, which improves the accuracy and performance of AI models.

PART 3: Seaborn

1. What is Seaborn and how is it different from Matplotlib?

Seaborn is a Python data visualization library built on top of Matplotlib. It is mainly used to create statistical and attractive graphs with very little coding.
Matplotlib gives more control over customization, while Seaborn focuses on easy plotting with beautiful default styles and statistical features. Seaborn is commonly used for data analysis and exploration. It is especially popular in machine learning and AI projects.

2. What is a statistical plot?

A statistical plot is a graph used to represent statistical properties of data such as distribution, central tendency, spread, and relationships between variables.
These plots help in understanding how data behaves and are very useful in data analysis and machine learning projects. They allow analysts to make better decisions based on visual patterns.

3. Difference between distplot / histplot and boxplot

Distplot or histplot shows how data values are distributed using bars or curves. It helps to understand the shape and frequency of data.
Boxplot shows the minimum, maximum, median, quartiles, and outliers in the data. It gives a clear statistical summary of the dataset in a compact form. This makes it easy to compare multiple datasets.

4. What is a heatmap and where is it used?

A heatmap is a graphical representation of data using different colors to show value intensity.
It is mainly used to display correlation between variables, confusion matrices in machine learning, and large datasets where color makes comparison easier. Heatmaps quickly highlight strong and weak relationships.

5. Why is Seaborn useful for data exploration in AI projects?

Seaborn helps in quickly visualizing relationships, patterns, and distributions in data.
It makes data exploration easier, helps in detecting outliers and correlations, and supports better decision-making before training AI models. This leads to more accurate and reliable AI models.