

### 3a. Write a Python program to illustrate liner plotting with line formatting using Matplotlib

#### 1. Introduction.

- Linear plotting is a type of graph used to represent data that follows a straight-line relationship. If we have a dataset where the values increase or decrease at a constant rate, the linear plot will show a straight line. The aim of the program is to illustrate linear plotting using Matplotlib. It compares the revenue growth of two companies, Company 1 and Company 2, over the years from 2015 to 2023. The program uses Matplotlib to create a visually appealing line plot that compares the revenue growth of two companies over a specific time period. It demonstrates the use of different line formatting options for improved data representation.
- The program uses different line formatting (solid and dashed lines) for the two companies to enhance data visualization. The ability to format lines with different styles and markers provides flexibility in conveying information effectively.

#### 2. Program code.

```
import matplotlib.pyplot as plt

# Updated sample data for two companies with years from 2015 to 2023
company1_years = list(range(2015, 2024))
company1_revenue = [150, 170, 190, 200, 220, 250, 280, 300, 320]

company2_years = list(range(2015, 2024))
company2_revenue = [130, 150, 170, 180, 200, 230, 260, 280, 300]

# Plotting the data with different line formatting
plt.plot(company1_years, company1_revenue, label='Company 1', linestyle='-',
marker='o', color='blue')
plt.plot(company2_years, company2_revenue, label='Company 2', linestyle='--',
marker='s', color='green')

# Adding labels and title
plt.xlabel('Years')
plt.ylabel('Revenue (in millions)')
```

```
plt.title('Comparison of Company Revenue Over Time')

# Adding a legend
plt.legend()

# Display the plot
plt.grid(True)
plt.show()
```

### 3. Explanation of the code

#### 3a: Company1 years and revenue

*company1\_years = list(range(2015, 2024))*

- `range(2015, 2024)`: Generates a sequence of numbers starting from 2015 up to (but not including) 2024.
- `list(...)`: Converts the generated range into a list.
- `company1_years`: Assigns the resulting list to the variable `company1_years`.

This line initializes the variable `company1_years` with a list of years from 2015 to 2023 (inclusive).

This list represents the x-axis values for Company 1's revenue data.

*company1\_revenue = [150, 170, 190, 200, 220, 250, 280, 300, 320]*

- `company1_revenue`: Assigns a list of revenue values to the variable `company1_revenue`.
- This line initializes the variable `company1_revenue` with a list of corresponding revenue values for each year in `company1_years`. For instance, in the first year (2015), the revenue is 150 million, and this pattern continues for each subsequent year.

These two lines of code define the data for Company 1's revenue over a specific period (2015 to 2023). The `company1_years` list represents the years, and the `company1_revenue` list holds the corresponding revenue values for each year. These data structures are crucial for creating meaningful plots and visualizations, enabling the analysis of trends and patterns over time.

**The same follows for Company2 years and revenue.**

### 3b: Plotting Company 1 and 2 Data

```
plt.plot(company1_years, company1_revenue, label='Company 1', linestyle='-', marker='o', color='blue')
```

- This line creates a line plot for Company 1's revenue data. The `company1_years` list provides the x-axis values, `company1_revenue` provides the y-axis values, and the formatting parameters define how the line and markers should appear on the plot.
- `plt.plot(...)`: Invokes the plot function from the Matplotlib library, used for creating line plots.
  - `company1_years`: X-axis values representing the years for Company 1.
  - `company1_revenue`: Y-axis values representing the corresponding revenue for Company 1.
  - `label='Company 1'`: Assigns a label to identify Company 1's line in the legend.
  - `linestyle='-'`: Sets the line style to solid.
  - `marker='o'`: Sets circular markers at each data point.
  - `color='blue'`: Specifies the line color as blue.
- The same follows for **Company 2 Data** with `linestyle='--'`: Sets the line style to dashed.

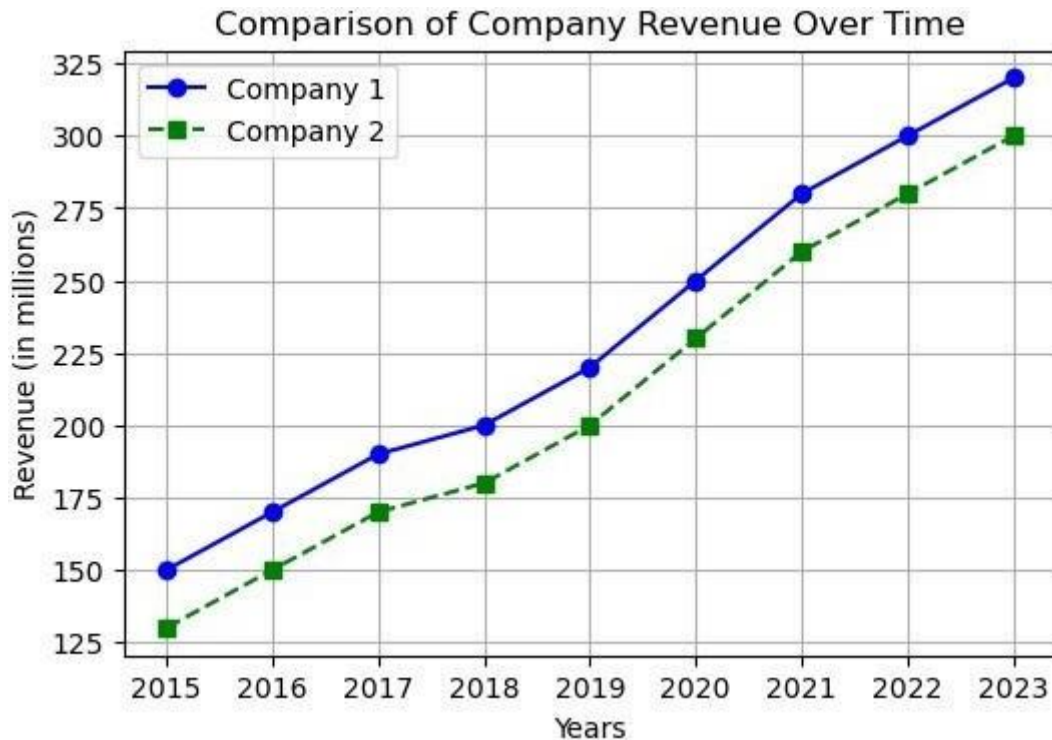
### 3C. `plt.grid(...)` Function

```
plt.grid(True)
```

- The purpose of this line is to enhance the visual presentation of the plot by adding a grid. A grid consists of horizontal and vertical lines that intersect at regular intervals, forming a background lattice. It aids in better readability and interpretation of the data points on the plot.
- The parameter `True` specifies that the grid should be visible. If `False` were used instead, the grid would be turned off.
- Including a grid on the plot is particularly useful when dealing with numerical data, as it allows for easier estimation of values, identification of trends, and comparison of data points against the gridlines.

## 4. Output

The program generates a line plot comparing the revenue of Company 1 and Company 2 over the years. Two lines are drawn with different styles and markers, and the legend indicates which line corresponds to which company.



### Chart Analysis

- .....
1. This chart shows how the revenue of two companies changed over time from 2015 to 2023. Revenue is the amount of money that a company earns from selling its products or services. The chart compares the revenue of Company 1 and Company 2 in each year.
  2. Both companies increased their revenue over time, which means they were growing and making more money.
  3. The chart suggests that Company 1 had a stronger competitive advantage than Company 2, which means it had a better product, service, or strategy that attracted more customers and generated more sales.
- .....

4. The chart also implies that Company 1 had more potential for future growth than Company 2, which means it could continue to increase its revenue and market share in the coming years.
5. The chart could be used as a tool for evaluating the performance and progress of the two companies, as well as identifying their strengths and weaknesses.

.....