2. Programming Style **Theory:**

❖ **Understanding Python’s PEP 8 guidelines.**

PEP 8, or Python Enhancement Proposal 8, is the official style guide for Python code. Authored by Guido van Rossum, Barry Warsaw, and Alyssa Coghlan, it was introduced in 2001 to promote readability and consistency in Python codebases. By adhering to PEP 8, developers ensure that their code is clean, understandable, and maintainable, which is particularly beneficial in collaborative environments.

❖ **Indentation, comments, and naming conventions in Python.**

Indentation

- Spaces over Tabs : Use 4 spaces per indentation level. Avoid mixing tabs and spaces.

- Line Length: Limit all lines to a maximum of 79 characters. For comments and docstrings, limit lines to 72 characters.
- Continuation Lines: Prefer implicit line continuation inside parentheses, brackets, and braces. If necessary, use a backslash (\) for explicit continuation.

Comments

- Block Comments: Use block comments to explain code in detail. Each line should start with a # followed by a single space. Indent block comments to the same level as the code they describe.
- Inline Comments: Place inline comments on the same line as the statement they refer to, separated by at least two spaces. Use them sparingly and only to clarify complex code.
- Docstrings: Use triple double quotes (""") for docstrings. Provide a summary of the method's purpose, its parameters, and return values. For multiline docstrings, include a blank line after the summary.

Naming Conventions

- Avoid single-letter variable names like l, O, or I to prevent confusion with 1 and 0.
- Function Arguments: If a function argument name clashes with a keyword, append an underscore (e.g., break_).
- Private Members: Use a leading underscore (_) for non-public methods and variables. For name mangling, use double underscores (__).

❖ Writing readable and maintainable code.

Writing clean, readable, and maintainable Python code is crucial for long-term project success. By adhering to best practices, you ensure that your code is understandable, scalable, and easy to modify.

1. Follow PEP 8 Guidelines

2. Use Meaningful and Consistent Naming

3. Apply the DRY Principle (Don’t Repeat Yourself)

4. Write Testable Code

5. Refactor Regularly

6. Document Your Code

7. Keep Functions and Methods Small

8. Implement Effective Error Handling

9. Use Version Control

10. Keep Up with Best Practices and New Technologies