**Que 1 :Understanding Python’s PEP 8 guidelines.**

**What is PEP 8?**

**PEP 8** stands for **Python Enhancement Proposal 8**.  
It is the **official style guide** for writing clean, readable, and consistent Python code.

Think of it as **rules for writing “beautiful” Python code** so others can easily understand and maintain it.

**8 Guidelines :**

**1. Indentation**

* Use **4 spaces per indentation level** (not tabs).

**2. Line Length**

* Keep lines **under 79 characters** long.

**3. Blank Lines**

* Use **blank lines** to separate functions or sections.

**4. Naming Conventions**

|  |  |  |
| --- | --- | --- |
| **Type** | **Naming Style** | **Example** |
| Variable/function | lowercase\_with\_underscores | total\_count |
| Class | CapitalizedWords | StudentData |
| Constant | UPPERCASE | MAX\_SPEED |

**5. Imports**

* Imports should be on separate lines at the top of the file.

**6. Spaces**

* Number of spaces should be same while declaring the variable.

**7. Comments**

* Use comments to explain **why**, not **what** the code is doing.

**8. Docstrings**

* Use triple quotes (""") to describe what a function or class does.

**Que 2 : Indentation, comments, and naming conventions in Python.**

**1. Indentation in Python**

**Meaning:**

Indentation refers to **spaces at the beginning of a line** of code.  
In Python, **indentation is mandatory** — it shows which code belongs inside loops, functions, if statements, etc.

**Rules:**

* Use **4 spaces per indentation level**
* **Never mix tabs and spaces**

**2. Comments in Python**

**Meaning**

Comments are **notes** written in your code to explain what it does.  
They are **ignored by Python** when the program runs.

**Types of Comments:**

1. **Single-line Comment** – starts with #
2. **Multi-line Comment / Docstring – uses triple quotes (''' or “"")**

**3. Naming Conventions in Python (PEP 8)**

✅ **Good naming makes your code easy to read and understand.**

|  |  |  |
| --- | --- | --- |
| **Element** | **Naming Style** | **Example** |
| Variable/function | lowercase\_with\_underscores | total\_marks |
| Class name | CapitalizedWords | StudentData |
| Constant | ALL\_UPPERCASE | MAX\_SIZE |
| Private variable | \_starts\_with\_underscore | \_hidden\_value |

**Que 3 : Writing readable and maintainable code.**

**1. Use Clear Names**

* Correct way: total\_marks, student\_name
* Incorrect way : x, a1

**2. Follow PEP 8**

* Use **4 spaces** for indentation
* Keep lines under **79 characters**

**3. Write Comments**

* Use # to explain logic
* Use """ """ for functions

**4. Use Functions**

Break big code into small, reusable functions.

**5. Avoid Repeating Code**

Use loops or functions instead of writing the same code multiple times.

**6. Handle Errors**

Use try-except to catch and handle errors.

**7. Keep Code Organized**

Use proper file/folder names and structure.

**Example:**

def greet(name):

"""Greet the user by name."""

print("Hello,", name)

greet("Krishna")