

# **IT314 - Software Engineering**

**Prof : Saurabh Tiwari**

**Group 9 : Pull Panda**

(AI-powered Pull Request Reviewer)

---

## **Static Analysis Report**

# **W-Series (Warnings)**

- These indicate risky, non-ideal, or potentially incorrect code patterns.

## **W0621 – Redefining Name From Outer Scope**

- Meaning: A variable or function argument shadows a name from an outer scope.
- Cause: Using the same variable name as a global or parent scope variable.
- Fixes:
  - Rename the inner variable.
  - Rename the outer variable if needed.
  - If overriding on purpose, disable the warning for that line (not preferred).

## **W0719 – Catching Too General Exception**

- Meaning: Using a very broad exception like “except Exception”.
- Cause: Overly generic error handling.
- Fixes:
  - Catch a specific exception type
  - Catch a tuple of specific exceptions.
  - Add logging and re-raise if broad catch is unavoidable.
  - Disable warning only if required for framework-level code.

## **W0718 – Catching BaseException**

- Meaning: Using “except BaseException”, which catches system-level interrupts.
- Cause: Handling exceptions too broadly.
- Fixes:
  - Replace with “except Exception”.
  - Catch specific exception types.
  - Handle KeyboardInterrupt and SystemExit separately if required
  - Disable the warning for special use cases.

## **W0311 – Bad Indentation**

- Meaning: Indentation does not follow the 4-space PEP8 rule.
- Cause: Mixing tabs and spaces or using incorrect spacing.
- Fixes:
  - Convert all indentation to 4 spaces.
  - Enable auto-formatting in your editor.
  - Use a formatter tool to fix indentation.

## **W0611 – Unused Import**

- Meaning: A module was imported but never used.
- Cause: Dead code, leftover imports, or unused side-effect imports.
- Fixes:
  - Remove the unused import.
  - If import is for side effects, alias it with an underscore and comment it.
  - Disable warning for very rare intentional cases.

## **W0613 – Unused Argument**

- Meaning: A function argument is never used inside the function body.
- Cause: API signatures, callback requirements, or oversight.
- Fixes:
  - Remove the unused argument.
  - Prefix the argument with an underscore to show it is intentional.
  - Add documentation explaining why it is unused.

- Disable warning when required for interface compliance.

## **W0702 - No Exception Type Specified**

- Meaning: Using a bare “except” without specifying an exception type.
- Cause: Trying to catch all exceptions blindly.
- Fixes:
  - Replace with “except Exception”.
  - Catch specific error types.
  - Include logging to avoid silent failures.
  - Use bare except only for very safe cleanup code.

## **W0612 - Unused Variable**

- Meaning: A variable is assigned a value but never used.
- Cause: Leftover debugging code or unnecessary assignment.
- Fixes:
  - Remove the unused variable.
  - Rename it to an underscore if needed only for assignment side effects.
  - Use the variable if it was mistakenly omitted.
  - Disable warning for special debugging scenarios.

## **W1514 - Missing Encoding Argument in Open**

- Meaning: The ‘open()’ function is used without specifying the file encoding.
- Cause: Relying on OS default encoding.
- Fixes:
  - Add an explicit encoding such as UTF-8.
  - Use binary mode if encoding is irrelevant.
  - Define a project-wide standard for encoding.

## **W1309 - F-string With No Placeholders**

- Meaning: An f-string is used even though it contains no variables.
- Cause: Typing habit or copy-paste error.
- Fixes:
  - Remove the f-prefix.
  - Add a placeholder if it was intended.

- Disable only if generating template strings dynamically.

### **W4903 – Raise Statement Outside Exception Handler**

- Meaning: A “raise” is used with no active exception to re-raise.
- Cause: Misplaced or incorrect re-raise.
- Fixes:
  - Raise a specific error with a message.
  - Move the raise statement inside an exception block.
  - Ensure that re-raises occur only within an except block.

## **C-SERIES (CONVENTION)**

- **These are PEP8 style or formatting issues.**

### **C0114 – Missing Module Docstring**

- Meaning: The file has no description at the top.
- Fixes:
  - Add a descriptive module-level docstring.
  - Disable the rule if the file is extremely small and self-explanatory.

### **C0411 – Wrong Import Order**

- Meaning: Import grouping does not follow PEP8 order.
- Fixes:
  - Reorder imports: standard library, third-party, then local.
  - Use “isort” or another auto-import sorter.
  - Separate groups with a blank line.

## C0116 – Missing Function Docstring

- Meaning: A function does not have a docstring.
- Fixes:
  - Add a docstring explaining what the function does.
  - Disable only for trivial one-liner functions.

## C0301 – Line Too Long

- Meaning: A line exceeds the maximum allowed length (typically 100–120 characters).
- Fixes:
  - Break the line into multiple lines.
  - Continue expressions across multiple lines using parentheses.
  - Increase the line limit only if your team agrees.

## C0303 – Trailing Whitespace

- Meaning: Extra spaces appear at the end of a line.
- Fixes:
  - Remove trailing whitespace.
  - Enable auto-formatting to avoid it.

## C0325 – Unnecessary Parentheses

- Meaning: Parentheses used where they are not required.
- Fixes:
  - Remove unnecessary parentheses.
  - Keep only where needed for grouping complex conditions.

## C0304 – Missing Final Newline

- Meaning: File does not end with a single newline.
- Fixes:

- Add exactly one newline at the end of the file.
- Set editor to automatically add final newline.

## C0103 – Invalid Naming Style

- Meaning: Names do not follow proper naming conventions.
- Fixes:
  - Rename variables to snake\_case.
  - Rename classes to PascalCase.
  - Rename constants to UPPER\_CASE.
  - Disable only when the naming is required by external API or specification.