**Doxygen Commenters' Thread**

1. **Commenting Guidelines Summary**

**@brief**: A brief summary of the function or class.

**@param**: Details of function parameters.

**@return**: Explanation of what the function returns.

**@warning**: Highlights any important warnings.

**@note**: Provides additional information or notes on usage.

**@todo**: Indicates future work or improvements needed.

1. Function Documentation



**Explanation**:

@brief: to provide a concise summary,

@param: tags for each parameter

@return: to describe the function's output.

@note: for additional information

@warning: to highlight a potential issue

1. Class Documentation



**Explanation**:

@brief:  to provide an overview of the class’s purpose.

@param, @return, @note, @warning: to provide critical usage information

1. Discussion Topics

**a). For Complex Algorithms:**

When documenting functions with complex logic, it is important to break down the algorithm step by step in the comments.

(use @note to explain the approach and assumptions made)

**b). Edge Cases and Performance Considerations:**

Discuss when and how to document edge cases using @warning and @note tags.

(Include examples of how to indicate potential performance impacts)

**c). Consistency Across the Project:**

Maintaining a consistent style in comments is crucial for readability. Developers are encouraged to share their methods for ensuring uniformity across large codebases.

1. Best Practices

**a). Keep Comments Concise:**

Use @brief for short summaries and expand only when necessary.

**b). Use Complete Sentences:**

This improves readability and ensures clarity.

**c). Avoid Over-Commenting:**

Comments should provide value; avoid stating the obvious, such as commenting on simple variable assignments or loops unless they have specific importance.

1. Feedback and Collaboration

**a). Interactive Comment Review Sessions:**

Developers can participate in bi-weekly sessions to review and discuss code comments. These sessions focus on improving clarity, consistency, and coverage of the comments.

**b). Suggestions and Improvements:**

Developers are encouraged to propose new commenting methods, or to refine existing ones, by opening discussion threads or submitting code patches.

**c). Contribution Guidelines:**

Ensure all new code contributions adhere to the established Doxygen guidelines. Use the provided templates for consistency.