“Guidelines for C Programming”

- **Consistent Formatting:**

  Maintain consistent indentation, spacing, and naming conventions throughout the code base.

- **Meaningful Names:**

  Use descriptive names for variables, functions, and other identifiers to enhance code clarity and understanding.

- **Commenting Document:**

  Code logic, algorithmic decisions, and any non-trivial sections with clear and concise comments.

- **Modularization:**

  Divide code into smaller, cohesive modules or functions, each responsible for a specific task or functionality.

- **Error Handling:**

  Implement robust error-checking and handling mechanisms to ensure proper program behavior in unexpected situations.

- **Optimization:**

  Prioritize clear and readable code over premature optimization, optimizing performance only when necessary and backed by profiling data.

- **Function Documentation Template:**

  Use the following template to document functions in your C code:

```
/*---------------------------------------------------------------------------
 NAME:              BlinkLed (int wait)
 DESCRIPTION:       Blinked the Led at GPIO25

 INPUTS:            int wait, Led on-off delay in ms
 OUTPUTS:           void

 PROCESS:           turn on an LED for wait ms and turn it off for wait ms
 AUTHOR:            Aamir Farooqui.
 DATE:              5/10/2024
 MODIFIED BY:
 COMMENTS:
 VERSION:           V1.0

-------------------------------------------------------------------------------*/
```

Adhering to these guidelines promotes code quality, readability, and maintainability, ultimately contributing to the success of C programming projects.