Google C++ Style Guide – Cheat Sheet

1. Naming Conventions

Classes / Structs / Enums: PascalCase
Functions / Methods: PascalCase
Variables: snake_case
Constants: k + PascalCase
Namespaces: lowercase
Enum values: PascalCase
File names: lowercase_with_underscores

2. Formatting

• Indent with 2 spaces (no tabs)
• Line length ≤ 80–100 chars

Braces:

```
if (condition) {
  DoSomething();
} else {
  DoOtherThing();
}
```

Pointer & Reference: int* ptr; int& ref = x;

3. Functions and Methods

• Use **const** for non-modifying methods
• Pass large objects by **const reference**
• Use **override** instead of virtual in derived classes

4. Comments

• Use // for regular comments
• Use Doxygen-style for documentation

```
// Computes average value
double ComputeAverage(const std::vector<int>& samples);
```

5. Includes

Include order:
1. Own header
2. C system headers
3. C++ standard headers
4. Other libraries
5. Local headers

Avoid using namespace std; in headers.

6. Misc Rules

• Use **nullptr** not NULL
• Use **enum class** instead of enum
• Prefer **auto** when type is obvious
• Avoid macros, prefer **constexpr**
• Use smart pointers (unique_ptr, shared_ptr)
• Always initialize variables

7. Example

```
// Copyright 2025 Inaz
// Description: Radar tracking backend

#include "entities/track.h"
namespace backend {

class RadarTracker {
  public:
    explicit RadarTracker(int id);
    void Update(const TrackData& data);
    double GetAccuracy() const;

  private:
    int id_;
    double accuracy_;
};

} // namespace backend
```