

# Style Guide for RDS

---

This project is based on the [google style guide](#) with some adjustments as well as some additions.

## Changes to the Google style guide

---

1) The indent width is changed from 2 to 4 spaces

## File Headers and define guards

---

### Cpp Files

---

```
/**
 * @file main.cpp
 * @brief The main file for the project
 */
```

### Cpp Files

---

```
/**
 * @file foo.hpp
 * @brief header for foo
 */
#ifndef FOO_HPP
#define FOO_HPP
// code
#endif // FOO_HPP
```

## Basic Format

---

```
// should be used to one line comment
```

```
/**
 * should be used for block comments
 * meaning logical dividers and
 * comments that span multiple lines
 */
```

## Naming

---

- Classes, File names except main will use snake case with the first letter capitalized.
  - ex. Activity\_log
- Variables and Functions will use snake case
  - ex. radio\_selected
  - Manage functions do not follow this.
- Memory constants of any form will be uppercase snake case
  - ex. MENU\_PROMPT

## Function

---

The function comment block is only required for functions which meaning is not obvious.

```
/**
 * @brief computes the area of a rectangle
 * @param[in] width width of the rectangle
 * @param[in] length length of the rectangle
 * @returns area of the rectangle
 */
int area(const int& width, const int& length) {
    return (width * length);
}
```

## Class

---

```
/**
 * @brief Rectangle class
 * Abstraction for a Rectangle
 */
class Rect {
private:
    int height;
    int width;

public:
    Rect(int x, int y) : height(x), width(y) {}
    std::pair<int, int> getRect();
};
```

## Loop

---

```
for (int i = 0; i < 5; i++) {
    cout << "hi";
}
```

## If

---

```
if (x < 5) {
    cout << "foo";
} else {
    cout << "bar";
}
```