Include guards

ALICE forbids `#pragma once` (can cause problems with symlinks)

Used by flesnet and tracking code

remainder of CbmRoot: various include guards

proposed by ALICE:

```
#ifndef <project>_<path>_<classname>_H_
```

E.g.:

```
#ifndef CBM_ONLINE_DETECTOR_STS_HITFINDER_H_
```

Naming

Variables:

- regular variables: lower case
- Compile time constants: CamelCaseCapitalized
- class members: prefixed with f (ALICE uses m instead)
- struct members: no prefix

Naming

- Function names: Start with lower case?
- namespaces: underscore and lower case: `some_namespace`
- Enum: CamelCaseCapitalized (like compile time constants)
- Class Names: CamelCaseCapitalized

Documentation

Doxygen comments: Use /// prefix

Currently (not formatted by clang, easy to mess up):

```
/**
 **
 **
 **/
```

```
/**
 *
 *
 **/
```

Mandatory documentation for files, . . . ?

Indentation

don’t indent namespace content (currently the case in CbmRoot)

```
namespace cbm {

  class SomeClass {
   public:
    SomeClass() = default;
  };

}
```

```
namespace cbm {

class SomeClass {
 public:
  SomeClass() = default;
};

}
```

control statements

4 possible formats for single-statement ifs:

Short statements:

```
if (someTest) functionCall();
if (someTest) { functionCall(); }
```

Long statements:

```
if (someTest)
  aVeryVeryVeryVeryLongStatement(foooooooooooobar);
if (someTest) {
  aVeryVeryVeryVeryLongStatement(foooooooooooobar);
}
```

control statements

Proposed change: Always wrap line and add braces

consistent with ALICE style and block easier to extend

only 1 format possible

```
if (someTest) {
  functionCall();
}
```

```
if (someTest) {
  aVeryVeryVeryVeryLongStatement(foooooooooooobar);
}
```