#define **is a preprocessor directive**: it is used to generate the eventual C++ code **before it is handled to the compiler** that will generate an executable. Therefore code like:

```
for(int i = 0; i < 54; i++) {
  #define BUFFER_SIZE 1024
}
```

is **not** executed 54 times (at the preprocessor level): the preprocessor simply runs over the for loop (not knowing what a for loop is), sees a define statement, associates 1024 with BUFFER_SIZE and continues. Until it reaches the bottom of the file.

You can write #define everywhere since the preprocessor is not really aware of the program itself.

Sure this is possible. The #define is processed by the preprocessor before the compiler does anything. It is a simple text replacement. The preprocessor doesn't even know if the line of code is inside or outside a function, class or whatever.

By the way, it is generally considered bad style to define preprocessor macros in C++. Most of the things they are used for can be better achieved with templates.