Code Quality: Treat Warnings as Error

Programmers often treat warnings very liberally when compared to errors.

Errors

- At least in statically typed languages like Java, C#, C++, the compiler reports errors when it finds something syntactically incorrect.
    - There are runtime errors — even statically typed languages have those, but more so languages that are dynamically typed have runtime errors.
- Compiler time errors — cannot even complete the compilation if there are compile time errors.
    - They block us from moving forward. So, we are forced to find and fix those errors.
    - Often, compilers report errors even at the slightest hint of syntactical deviation, like missing a semi-colon.
- We often deal with errors simply because we *have* to.
- Unfortunately, programmers tend to treat warnings much more lightly.

When & How

- Turn on the flag on the compiler that would treat warning as errors.
    - This way, the complier won’t allow us to move forward until we have fixed those warnings.

Deal with it Pragmatically

- The warnings may be from a layer of code not in our control. They may be from some third-party library/api.

    - We can turn off the warning on those specific libraries or marking them with annotations or attributes, whatever is available in the language.
- There are times when they may be a lot of warning in the code, especially in legacy code. Turning on the flag to treat warning as errors may be overwhelming — the warnings/errors may be too much for us to deal with in a reasonable amount of time.