Everywhere?!

Let's make a general outline about the appropriate use of [[nodiscard]].

There's a paper that might be a "guide" P0600R0 - [[nodiscard]] in the Library.

The proposal didn't make into C++17 but was voted into C++20. It suggests a few places were the attribute should be applied.

For existing API's:

- not using the return value always is a "huge mistake" (e.g., always resulting in resource leak)
- not using the return value is a source of trouble and can happen easily (not obvious that something is wrong)

For new API's (not been in the C++ standard yet):

- not using the return value is usually an error.

Here are a few examples where the new attribute should be added:

- `malloc()` / `new` / `allocate` - expensive call, usually not using the return value is a resource leak
- `std::async()` - not using the return value makes the call synchronous, which might be hard to detect.

On the other hand such function as `top()` are questionable, as "not very useful, but no danger and such code might exist"

It's probably a good idea not to add `[[nodiscard]]` in all places of your code but focus on the critical places. Possibly, as mentioned before, error codes and factories are a good place to start.

Lastly, we'll discuss alternatives to this attribute.