Doc. No.: Pxxxx
Project: Programming Language - C++ (WG21)
Author: Andrew Tomazos <andrewtomazos@gmail.com>
Date: 2019-11-21
Audience: EWGI
Target: C++23

Proposal of constpub

We propose a new decl-specifier called **constpub**.

constpub is short for "**const** in **pub**lic".

constpub may only be applied to the declaration of a non-static data member of a class type that is declared in the public section of its class:

```
constpub int x; // ERROR: not in class definition

class C {
 public:
   constpub int y; // OK
 private:
   constpub int z; // ERROR: not in public section
};
```

It makes it so that the data member is const when accessed from a public context, but is non-const in a private context. (Formally the underlying object is non-const but the id-expression in a public context produces a const reference to it.)

For example:

```
struct S {
  constpub int i;

  void f() {
    std::cout << i; // OK: private read
    i = 42; // OK: private write
  }
};

int main() {
  S s;
  std::cout << s.i; // OK: public read
  s.i = 42; // ERROR: public write
}
```

This is an extremely common thing to want to do. A class designer often uses access control in order to maintain some invariant with respect to a data member, so wants to prevent public modification, but wants to allow public read access.

The way this is handled today without this feature is by placing the data member in the private section and then providing a public accessor:

```
  // TODAY
  struct S {
    const int& get_i() const { return i; }
   private:
    int i;
  }

  int main() { std::cout << S().get_i(); }

  // PROPOSED
  struct S {
    constpub int i;
  }

  int main() { std::cout << S().i; }
```

As can be seen constpub represents a significant saving on boilerplate given how common this intent is desired to be expressed. It also avoids the naming collision between the data member and the accessor which would otherwise need to be resolved by some naming convention (accessors have get_ prefix, or members have m_ prefix, or something like that).