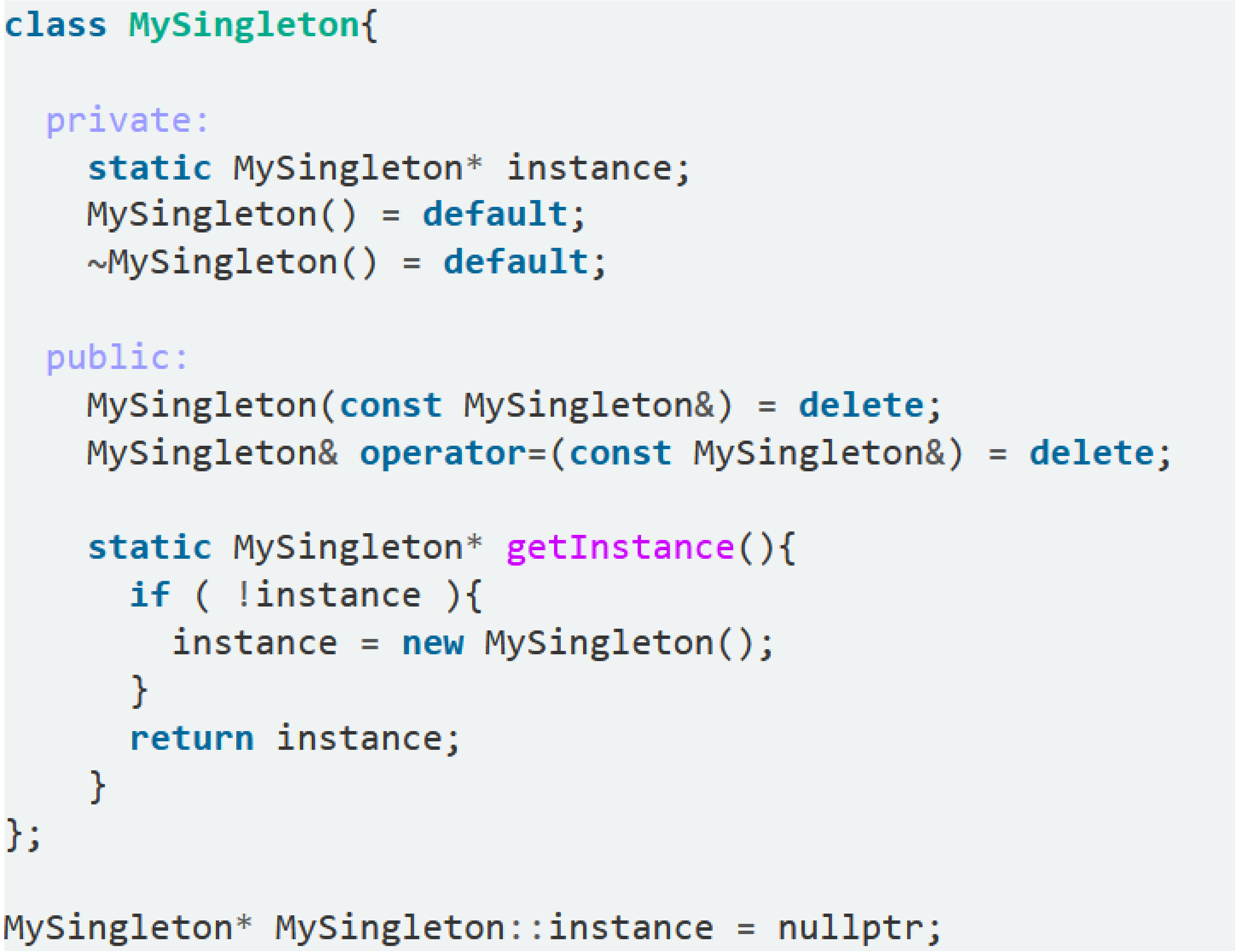
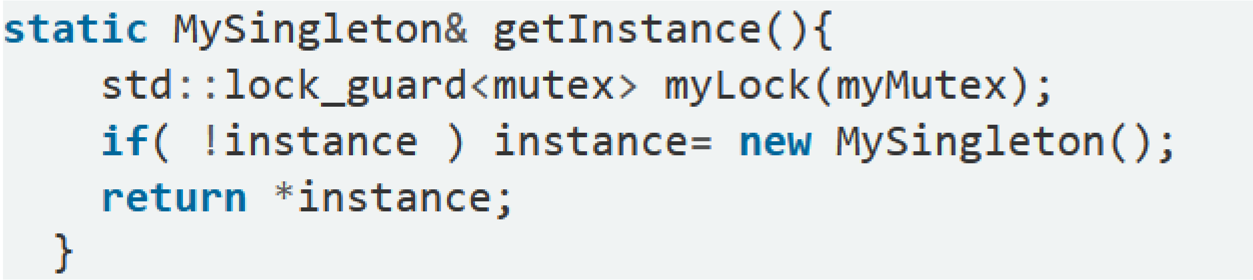
Double-Checked Locking Pattern



```
class MySingleton{

  private:
    static MySingleton* instance;
    MySingleton() = default;
    ~MySingleton() = default;

  public:
    MySingleton(const MySingleton&) = delete;
    MySingleton& operator=(const MySingleton&) = delete;

    static MySingleton* getInstance(){
      if ( !instance ){
        instance = new MySingleton();
      }
      return instance;
    }
};

MySingleton* MySingleton::instance = nullptr;
```

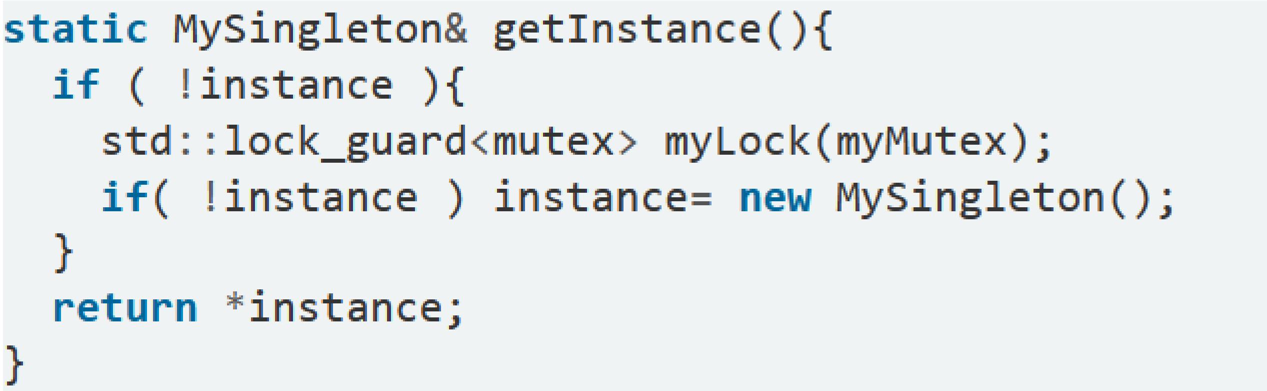
Double-Checked Locking Pattern

- Thread-Safe Singleton



```
static MySingleton& getInstance(){
    std::lock_guard<mutex> myLock(myMutex);
    if( !instance ) instance= new MySingleton();
    return *instance;
  }
```

- Double-Checked Locking Pattern by Douglas Schmidt



```
static MySingleton& getInstance(){
  if ( !instance ){
    std::lock_guard<mutex> myLock(myMutex);
    if( !instance ) instance= new MySingleton();
  }
  return *instance;
}
```

Double-Checked Locking Pattern is broken