```cpp
#include "FibonacciSequenceGenerator.h"
#include <stdexcept>

FibonacciSequenceGenerator::FibonacciSequenceGenerator(const std::string& aID) noexcept :
fID(aID), fPrevious(0), fCurrent(1) {}

const std::string& FibonacciSequenceGenerator::id() const noexcept {
    return fID;
}

const long long& FibonacciSequenceGenerator::operator*() const noexcept {
    return fCurrent;
}

FibonacciSequenceGenerator::operator bool() const noexcept {
    return hasNext();
}

void FibonacciSequenceGenerator::reset() noexcept {
    fPrevious = 0;
    fCurrent = 1;
}

bool FibonacciSequenceGenerator::hasNext() const noexcept {
    // Check for overflow by comparing with the maximum representable value
    return fCurrent <= (std::numeric_limits<long long>::max() - fPrevious) && fCurrent >= 0;
}

void FibonacciSequenceGenerator::next() noexcept {
    long long next = fCurrent + fPrevious;
    fPrevious = fCurrent;
    fCurrent = next;
}
```