```cpp
//
//  FibonacciSquenceGenerator.cpp
//  Asignment2
//
//  Created by Luan Nguyen on 15/4/2024.
//

#include <stdio.h>
#include "FibonacciSequenceGenerator.h"
#include <string>
#include <cassert>
#include <cstddef>



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

// Get Sequence ID
const std::string& FibonacciSequenceGenerator::id() const noexcept{
    return fID;
}

// Get current Fibonacci number
const long long& FibonacciSequenceGenerator::operator*() const noexcept{
    return fCurrent;
}
// Converse type to bool
FibonacciSequenceGenerator::operator bool() const noexcept{
    return hasNext();
}


void FibonacciSequenceGenerator::reset() noexcept{
    fPrevious = 0;
    fCurrent = 1;
} // Reset the prev and current values

bool FibonacciSequenceGenerator::hasNext() const noexcept{
    long long lnext = fPrevious + fCurrent;
    return lnext>=0; // Return True if the next value is positive

}
void FibonacciSequenceGenerator::next() noexcept {
    // Perform the Fibonacci sequence calculation
    long long lnext = fPrevious + fCurrent;

    // Precondition assertion to guarantee no negative values
    assert(lnext >= 0);

    // Update previous and current values
    fPrevious = fCurrent;
    fCurrent = lnext;
}
```