Swinburne University of Technology

Faculty of Science, Engineering and Technology

ASSIGNMENT COVER SHEET

Subject Code:	COS30008
Subject Title:	Data Structures & Patterns
Assignment number and title:	2 - Iterators
Due date:	Monday, 22 April, 2024, 10:30
Lecturer:	Dr. Markus Lumpe

Your name: Dang Khoa Le **Your student id:** 103844421

Marker's comments:

Problem	Marks	Obtained
1	40	
2	70	
Total	110	

Extension certification:

This assignment has been given an extension and is now due on ____________

Signature of Convener:_________________________

```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;
}
```

```cpp
#include "FibonacciSequenceIterator.h"
#include <cassert>

FibonacciSequenceIterator::FibonacciSequenceIterator(const FibonacciSequenceGenerator&
aSequenceObject, long long aStart) noexcept
    : fSequenceObject(aSequenceObject), fIndex(aStart - 1) {}

const long long& FibonacciSequenceIterator::operator*() const noexcept {
    return *fSequenceObject;
}

FibonacciSequenceIterator& FibonacciSequenceIterator::operator++() noexcept {
    ++fIndex;
    fSequenceObject.next(); // Advance the sequence generator
    return *this;
}

FibonacciSequenceIterator FibonacciSequenceIterator::operator++(int) noexcept {
    FibonacciSequenceIterator temp = *this;
    ++(*this);
    return temp;
}

bool FibonacciSequenceIterator::operator==(const FibonacciSequenceIterator& aOther) const
noexcept {
    return fSequenceObject.id() == aOther.fSequenceObject.id() && fIndex == aOther.fIndex;
}

bool FibonacciSequenceIterator::operator!=(const FibonacciSequenceIterator& aOther) const
noexcept {
    return !(*this == aOther);
}

FibonacciSequenceIterator FibonacciSequenceIterator::begin() const noexcept {
    return FibonacciSequenceIterator(fSequenceObject, 1);
}

FibonacciSequenceIterator FibonacciSequenceIterator::end() const noexcept {
    return FibonacciSequenceIterator(fSequenceObject, 93); // The end position is at Fibonacci
number 93
}
```