COS30008 Semester 1, 2024 Dr. Markus Lumpe

Swinburne University Of Technology

Faculty of Information and Communication Technologies

ASSIGNMENT COVER SHEET

Subject Code: COS30008
Subject Title: Data Structures & Patterns
Assignment number and title: 3 – Design Patterns and 12 Bit I/O
Due date: May 13, 2024, 10:30
Lecturer: Dr. Markus Lumpe

Your name: Avery Flannery **Your student id:** 104416957

Marker's comments:

Problem	Marks	Obtained
1	138	
Total	138	

Extension certification:

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

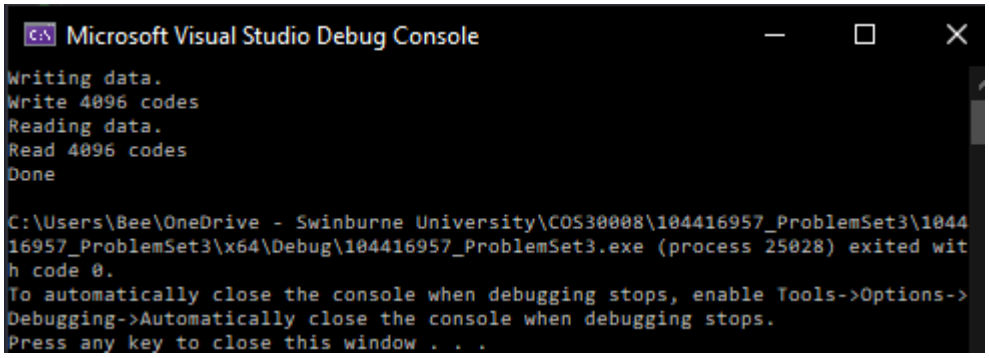
Signature of Convener:_____

Problem 1

```
//COS30008 - 1044169576 - Avery Flannery
//Problem Set 3 - Design Pattern and 12-bit I/o


#include "ifstream12.h"
#include <cassert>

// Reset function to reset the input stream
void ifstream12::reset() {
    // Reset buffer to all zeros
    for (size_t i = 0; i < fBufferSize; i++) {
        fBuffer[i] = std::byte{ 0 };
    }
    // Reset byte count, byte index, and bit index
    fByteCount = 0;
    fByteIndex = 0;
    fBitIndex = 7;
}

// Function to fetch data from the input stream
void ifstream12::fetch_data() {
    reset(); // Reset input stream
    fIStream.read(reinterpret_cast<char*>(fBuffer), fBufferSize); // Read data
from the input stream
    fByteCount = fIStream.gcount(); // Get the number of bytes read
}

// Function to read the next bit from the input stream
std::optional<size_t> ifstream12::readBit() {

    if (fByteCount == 0) {
        fetch_data();
    }

    if (eof()) { // If end of file is reached
        return std::nullopt;
    }

    std::byte lByte = (fBuffer[fByteIndex] & (std::byte{ 1 } << fBitIndex));
    fBitIndex--; // Move to the next bit

    if (fBitIndex < 0) {
        fByteCount--;
        fByteIndex++;
        fBitIndex = 7;
    }

    return std::to_integer<size_t>(lByte) != 0;
}

// Constructor
ifstream12::ifstream12(const char* aFileName, size_t aBufferSize) {
    fBuffer = new std::byte[aBufferSize]; //
    fBufferSize = aBufferSize;

    reset();

    open(aFileName);
}
```

```cpp
// Destructor
ifstream12::~ifstream12() {
    close();
    delete[] fBuffer;
}

// Function to open a file
void ifstream12::open(const char* aFileName) {
    assert(!isOpen());

    if (aFileName) {
        fIStream.open(aFileName, std::ifstream::binary);
    }
}

// Function to close the file
void ifstream12::close() {
    fIStream.close();
}

// Function to check if a file is open
bool ifstream12::isOpen() const {
    return fIStream.is_open();
}

// Function to check if the input stream is in good state
bool ifstream12::good() const {
    return fIStream.good();
}

// Function to check for end of file
bool ifstream12::eof() const {
    return (fByteCount == 0);
}

// Overloaded input operator to read 12 bits from the input stream
ifstream12& ifstream12::operator>>(size_t& aValue) {
    aValue = 0;
    for (int i = 0; i < 12; i++) {
        auto bit = readBit();

        if (bit == std::nullopt) {
            break;
        }
        else if (bit == 1) {
            aValue += (static_cast<size_t>(1) << i);
        }
    }

    return *this;
}
```

Output


Microsoft Visual Studio Debug Console
Writing data.
Write 4096 codes
Reading data.
Read 4096 codes
Done
C:\Users\Bee\OneDrive - Swinburne University\COS30008\104416957_ProblemSet3\104416957_ProblemSet3\x64\Debug\104416957_ProblemSet3.exe (process 25028) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .