

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: _____ **Your student id:** _____

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: _____

```
1  #include "ifstream12.h"
2  #include <cassert>
3
4  void ifstream12::reset() {
5      for (size_t i = 0; i < fBufferSize; i++)
6          fBuffer[i] &= std::byte{ 0 };
7      fByteCount = 0;
8      fByteIndex = 0;
9      fBitIndex = 7;
10 }
11
12 void ifstream12::fetch_data() {
13     fIStream.read(reinterpret_cast<char*>(fBuffer), fBufferSize);
14     fByteCount = fIStream.gcount();
15     fByteIndex = 0;
16     fBitIndex = 7;
17 }
18
19 std::optional<size_t> ifstream12::readBit() {
20     if (fByteCount == 0) {
21         if (fIStream.eof()) {
22             return std::nullopt; // Return no value if EOF is reached
23         }
24         fetch_data();
25     }
26
27     std::byte lByte = fBuffer[fByteIndex] & (std::byte{ 1 } <<
28         fBitIndex);
29     size_t lBitValue = std::to_integer<size_t>(lByte);
30     fBitIndex--;
31     if (fBitIndex < 0) {
32         fBitIndex = 7;
33         fByteIndex++;
34         fByteCount--;
35     }
36
37     if (lBitValue == 0) {
38         return 0;
39     }
40     else {
41         return 1;
42     }
43 }
44
45 ifstream12::ifstream12(const char* aFileName, size_t aBufferSize) :
46     fBuffer(new std::byte[aBufferSize]), fBufferSize(aBufferSize),
47     fByteCount(0), fByteIndex(0), fBitIndex(7) {
48
49     if (aFileName) {
50         open(aFileName);
51     }
52 }
```

```
53
54 ifstream12::~ifstream12() {
55     close();
56     delete[] fBuffer;
57 }
58
59 void ifstream12::open(const char* aFileName) {
60     assert(!isOpen());
61
62     if (aFileName) {
63         fIStream.open(aFileName, std::ifstream::binary);
64     }
65 }
66
67 void ifstream12::close() {
68     assert(isOpen());
69     fIStream.close();
70 }
71
72 bool ifstream12::isOpen() const {
73     return fIStream.is_open();
74 }
75
76 bool ifstream12::good() const {
77     return fIStream.good() && (fByteCount > 0 || !fIStream.eof());
78 }
79
80 bool ifstream12::eof() const {
81     return fByteCount == 0;
82 }
83
84 ifstream12& ifstream12::operator>>(size_t& aValue) {
85     aValue = 0;
86     for (size_t i = 0; i < 12; i++) {
87         auto bitOpt = readBit();
88         if (!bitOpt) {
89             break;
90         }
91         if (bitOpt == 1) {
92             aValue += (1 << i);
93         }
94     }
95
96     return *this;
97 }
```