DTS Lab 8  
bit streams

Contents

[Objective 2](#_Toc358034706)

[Prototypes 2](#_Toc358034707)

[Desired Output 4](#_Toc358034708)

[Submission 5](#_Toc358034709)

# Objective

Today you will construct two classes which will allow the user to stream individual bits to and from a file. Place all your code for both classes in a file called *BitStream.h*.

# Prototypes

BitOstream public interface:

/////////////////////////////////////////////////////////////////////////////// Function : constructor  
// Parameters : filePath - the path of the file to open for output  
// headerChunk - a chunk of data to be written at the   
// beginning of the file  
// numberOfBytes - the number of bytes of header information   
// to write out  
/////////////////////////////////////////////////////////////////////////////  
BitOStream(const char\* filePath, const char\* headerChunk = NULL, int numberOfBytes = 0)

/////////////////////////////////////////////////////////////////////////////  
// Function : destructor  
/////////////////////////////////////////////////////////////////////////////  
~BitOStream()

/////////////////////////////////////////////////////////////////////////////  
// Function : insertion operator  
// Parameters : bits - a vector containing some number of 1's and 0's to   
// stream out to the file  
// Return : BitOStream& - the stream (allows for daisy-chaining insertions)  
////////////////////////////////////////////////////////////////////////////  
BitOStream& operator<<(vector<int> &bits)

/////////////////////////////////////////////////////////////////////////////  
// Function : close  
// Notes : closes the file stream - if remaining bits exist, they are written  
// to the file with trailing 0's. if no remaining bits exist,   
// simply close the file  
/////////////////////////////////////////////////////////////////////////////  
void close()

BitIStream public interface:

/////////////////////////////////////////////////////////////////////////////  
// Function : constructor  
// Parameters : filePath - the path of the file to open for input  
// headerChunk - a chunk of data to be read from the   
// beginning of the file  
// numberOfBytes - the number of bytes of header information   
// to read in  
/////////////////////////////////////////////////////////////////////////////  
BitIStream(const char\* filePath, char\* headerChunk = NULL, int numberOfBytes = 0)

/////////////////////////////////////////////////////////////////////////////  
// Function : destructor  
/////////////////////////////////////////////////////////////////////////////  
~BitIStream()

/////////////////////////////////////////////////////////////////////////////  
// Function : extraction operator  
// Parameters : bit - store the next bit here  
// Return : BitIStream& - the stream (allows for daisy-chaining extractions)  
/////////////////////////////////////////////////////////////////////////////  
BitIStream& operator>>(int &bit)

/////////////////////////////////////////////////////////////////////////////  
// Function : eof  
// Return : true if we are at the end of the file, false otherwise  
// Notes : should only return true when we have given the user every byte  
// from the file and every bit from the buffer  
/////////////////////////////////////////////////////////////////////////////  
bool eof()

/////////////////////////////////////////////////////////////////////////////  
// Function : close  
// Notes : close the file  
/////////////////////////////////////////////////////////////////////////////  
void close()

# Desired Output

Compile and run your code with the DTSLab8.cpp file provided via FSO. Your console output should match the following block identically:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\* LAB 8: \*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
\*\*\* TEST 1 \*\*\*  
Writing a Sample Bit Stream (No Header)...  
35 ed 80   
  
\*\*\* TEST 2 \*\*\*  
Writing a Sample Bit Stream (WITH Header)...  
41 42 43 35   
  
\*\*\* TEST 3 \*\*\*  
Reading a Sample Bit Stream (No Header)...  
10101011 11001101 11101111   
  
\*\*\* TEST 4 \*\*\*  
Reading a Sample Bit Stream (WITH Header)...  
11101111

# Submission

To submit the lab assignment:

1. Clean, build, and run DTSLab8.cpp with your BitStream.h file in Visual Studio (debug mode).
   1. clear up any warnings you encounter.
   2. verify that your output is correct by comparing it to the lab document's Desired Output section, line-by-line.
   3. ensure there are no memory leaks.
2. Submit the "BitStream.h" file via FSO.