CMPUT 274

bitio.py Quick Reference

Topics Covered:
- BitReader
- EOFError
- BitWriter

BitReader

- BitReader(input_stream)
  - Creates an instance of the BitReader class
  - This object will read from the input_stream
- readbit()
  - Reads the next bit in the input_stream, and returns it as a 1 or 0
- readbits(n)
  - Reads n bits and returns them as the sequence of bits evaluated as an integer

BitReader example

```
import bitio

with open('simple.txt', 'rb') as fin:
    mybitreader = bitio.BitReader(fin)

    # read in a byte, one bit at a time
    for i in range(8):
        my_bit = mybitreader.readbit()
        print(my_bit, end = "")
    print()

    # read in a byte all at once
    my_byte = mybitreader.readbits(8)
    print(my_byte)
```

How to Read to End of File?

- EOFError is raised when there are no more bits to read from the input_stream

- We can catch and handle this exception gracefully

```
import bitio

with open('simple.txt', 'rb') as fin:
    mybitreader = bitio.BitReader(fin)
    end_of_file = False

    while not end_of_file:
        try:
            bit = mybitreader.readbit()
            print(bit, end = "")

        except EOFError:
            end_of_file = True
```

BitWriter

- BitWriter(output_stream)
  - Creates an instance of the BitWriter class
  - This object will write to the output_stream
- writebit(bit)
  - If bit is True, writes 1 to the output_stream
  - If bit is False, writes 0 to the output_stream
- writebits(integer_value, n)
  - Writes the n least significant bits of integer_value to the output_stream, starting with the most significant of these bits

BitWriter (cont.)

- flush()
  - Forces any bits waiting in buffer to the output_stream
  - ALWAYS call when finished writing to write any partial bytes to output_stream
  - Any incomplete bytes are automatically padded with extra 0s in least significant bits

BitWriter example

```
import bitio

seq1 = "01101000"
seq2 = [ord('E'), ord('L'), ord('L'), ord('O')]

with open('message.txt', 'wb') as fout:
    mybitwriter = bitio.BitWriter(fout)

    for single_bit in seq1:
        if single_bit == '1':
            mybitwriter.writebit(True)
        elif single_bit == '0':
            mybitwriter.writebit(False)
    mybitwriter.flush() # don’t forget at the end!

    for single_byte in seq2:
        mybitwriter.writebits(single_byte, 8)
    mybitwriter.flush() # don’t forget at the end!
```