CS 235 Final

Version 0.5

Instructors: K. Seamons and R. P. Burton

December 14 – 18, 2013 (Saturday through Wednesday)

Due in the Lab on Wednesday no later than 6:00 p.m.

Penalty for submitting the final late:

No Credit

Open Book (142 course text and your CS 235 course text only), Open Notes (including your own Lab solutions)

Open Secondary Storage Device: yours only

Open Laptop: if you wish

Open Course Website and www.cplusplus.com, but no other Internet resources (no Google)

Closed Neighbor (and everyone is thy neighbor)

**\*Instructions\***

(Please read carefully)

1. This test consists of a C++ programming problem with optional extra credit. Read and understand the statement of the problem completely before beginning to design, code, and test. As part of your design, consider the test cases that will establish the correctness of your solution. Test your solution thoroughly before submitting it.
2. Produce a solution, which consists of your C++ code, with a comment at the beginning of each file (both .h and.cpp) which includes your name, your CS 235 section, and “CS 235 Fall 2013 Final Exam.” Upload your completed project by compressing the files and submitting them through Learning Suite with TA assistance. If a packet is not collected by a TA upon submission, you will not be graded and will therefore receive no credit for the exam (ouch!). Attribute any code taken from or based on other sources (excluding the course texts and the course websites). Attributed code copied from or based heavily on outside sources is worth half credit. Unattributed code copied from or based heavily on outside sources is worth no credit.
3. Understanding the problem correctly is part of the examination. If something seems unclear, ask a CS 235 TA for clarification. You may pose questions to the CS 235 TAs via the Help Queue. However, the TAs generally are not permitted to answer questions related to design, C++ implementation, debugging, or testing.
4. Prior to submitting your final, score it using the attached scoring sheet (this will help you maximize your points and will help us grade your exam accurately).
5. Your solution packet must all be stapled together before it will be accepted by a TA.
6. You must receive a 75% on the final in order to receive an A grade in the course.
7. You will be given 5 points extra credit if your self-grading score is within 5 points of the TA Total score. You will lose 5 points if your self-graded score is more than 10 points different than the TA Total score.
8. When you are finished, submit your exam via Learning Suite with the assistance of a TA. There will be no timestamps accepted for the final. Incorrect files submitted for any reason will result in a 0 on the final exam.
9. Sign the grading sheet (1) to certify that you have reviewed your posted scores for the semester and alerted a CS 235 TA in writing to any discrepancies, (2) to request that your final be graded, and (3) to certify that no unfair information related to the final has been received by you, either directly or indirectly, and that none will be conveyed by you. If we discover that you cheated or assisted someone in cheating, intentionally or unintentionally (including accidentally), your score for the final may (and probably will) be rand() % 1.

We’re serious.

And He Huffed and He Puffed and…

**Requirements**

Develop an encoding/decoding program for any number of characters that can perform all of the operations described below. This will be completed by creating a Huffman tree (see section 8.6 of the textbook).

To complete this exam, you must extend the HuffmanInterface. This is a high level interface that allows us to enforce some needed functionality while still giving you freedom to complete the exam comfortably. You may use any additional methods or classes as you see fit and this is encouraged. Do not modify the interface in any way. You must create a class called HuffmanTree that implements the HuffmanInterface. You will not be able to pass off this exam if you do not implement the interface or if you name your class differently. This interface will allow us to automate and expedite the pass-off process.

Part 1 - Make the Encoding

* Implement the createTree and getEncoding method(s) of the HuffmanInterface
* Determine the frequency of occurrence of each character, including punctuation marks and whitespace characters, by scanning some sample text. Sample text is passed in as a parameter
* Determine a minimal Huffman encoding using the “Rules for Your Tree” in the Requirement Notes below

Part 2 - Encode

* Implement the encodeMessage method of the HuffmanInterface. Your encoded messages should contain 1's and 0's and nothing else

Part 3 - Decode

* Implement the decodeMessage method of the HuffmanInterface.

Extra Credit (5 points)

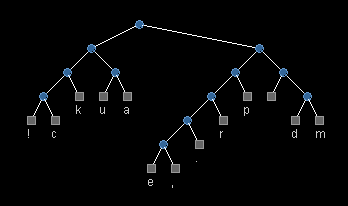
* Store the Huffman Encoding of a representative text as a binary file and as an ASCII file. Compare the memory requirements of the ASCII file and the binary file.

**Clarifications & Constraints**

* Your program needs to be “bullet-proofed”. You need to prepare for all possible input, even input which is not specifically mentioned in the specifications. This requires you as the programmer to anticipate what problems could arise from invalid input and to handle them appropriately.
* A test driver will be used to grade the exam; however it will not be available for student use. You should test your code thoroughly before submission.
* You are expected to create your own main function for testing purposes, but you will not be required to submit the file containing the main function.

**Rules for Your Tree**

* Here is a sample text (brackets not included): [muckduck! muuuud draumak pmdap dp ucmr, paumeu dpapkam. mka. rpkuakdap pamk aprckpku.arp pruak, dcd ckap! r! p! .!mrp]. Here is a visual tree created from the encoding of the message (the node that appears blank on the right side of the tree is for the space character):



* Several different sample text files will be used to grade your solution. For your convenience when testing, we have provided two files: a message “message.txt” and the encoding of that message “encMessage.txt” using the tree generated from the sample text above.
* For any given text, there could be multiple Huffman trees that are all optimal. This could lead to problems if two parties want to exchange messages that are encoded with trees generated using the same sample text. If each party generates a different, but still optimal, tree, they would end up with nonsensical decoded messages. To avoid this problem for your solution, we will use a set of arbitrary, but necessary, rules that will specify how certain aspects of the tree should be handled.
  + The left branch of a node will use a 0 in the encoding and the right branch will use a 1.
  + When combining two nodes to make a new node in your tree, the node with the lower frequency should be placed on the left and the node with the higher frequency should be on the right.
* In situations where there are duplicate frequencies, a few rules need to be used to create the ordering of your priority queue. Original characters should come before combined nodes. When ordering two original characters, the ordering of the ASCII character set should be used. When ordering two combined nodes, the older node should come first.

**Final Scoring Sheet** (A physical copy of this page must be turned in to the TAs)

Printed Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Professor Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section #:\_\_\_\_\_

Student Grading TA Grading

\_\_\_/ 50 pts \_\_\_/ 50 pts – Correctly implements the createTree(string message); method

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_/ 50 pts \_\_\_/ 50 pts – Correctly implements the encodeMessage(string toEncode); method

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_/ 40 pts \_\_\_/ 40 pts – Correctly implements the decodeMessage(string toDecode); method

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_/ 10 pts \_\_\_/ 10 pts – Coding Style

\_\_\_/ 5 pts \_\_\_/ 5 pts – Neat Code: correct indentation, comments as needed, helpful variable names

\_\_\_/ 5 pts \_\_\_/ 5 pts – No debugging cout statements or any print/pause statements in submitted files

**\_\_\_/ 5 pts \_\_\_/ 5 pts – Extra Credit**

\_\_\_/ 5 pts \_\_\_/ 5 pts – Binary/ASCII representation

\_\_\_/ 150 pts \_\_\_/ 150 pts – Total (before late penalties)

\_\_\_/ 150 pts – Final Score

Student to TA Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Received By: TA Name) (Date Received) (Late) (Graded By: TA Name) (Date Graded)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, 2013

(Student Signature)\* (Date)

\*By signing above, you agree that you agree to all of the expectations of this exam.