# *COSC 3100 – Data Structures II*

## Assignment 9
## Deadline November 13, 2023

In a file named 'HuffmanStocks.txt', on the course webpage, there is information for a list of Stocks that are managed by a company. The number of Stocks listed in the file is indicated at the beginning. In addition to each Stock (name, symbol, price), there is also a 'frequency' that indicates the chance of a particular Stock being sold. When a Stock is sold, the company has to inform many people around the world, so they want this process to be an efficient as possible. Each Stock has a frequency between 0 and 1, and the total of the Stock frequencies is 1.

```
10
American Express Co
AXP
118.67
0.1
Apple Inc
AAPL
120.3
0.15
…
```

*This program does not require any additional classes to be created, just the Stock class and the functions listed below, and then calling these functions from 'main'.*

Program requirements:

a) A 'Node' structure should be created to include 'double frequency' as a member.

b) The following function should be written to read the data from the 'HuffmanStocks.txt' file, create and populate the arrays 'stocks' and 'freqs', and return the number of stocks that were contained in the file:

**int readStocks(const string& fileName, Stock*& stocks,**
**double*& freqs);**

c) The Huffman Tree is constructed based on the data in 'stocks' and 'freqs'. The following function takes these arrays as arguments, and the size of the arrays, creates the Huffman tree and returns a pointer to the root of the tree:

**Node<Stock>* createHuffmanTree(Stock stocks[], double freqs[], int size);**

Within this function, an array of 'Node<Stock>*' named 'elements' should be dynamically allocated and used to support the creation of the tree. 'elements[0]', which will be a pointer to the root of the tree, should then be returned at the end of the function.

d) The following function will print an array of values, and the Stock identified at the end of the path:

**void printArray(int path[], int pathLen, Node<Stock>* leaf);**

e) The following function will display the path to each Stock:

**void printPaths(Node<Stock>* node, int path[], int pathLen, int lr);**

f) A 'string' variable called 'sentence' should be constructed in 'main'. This variable should contain several '0' and '1' characters, constructed based on the code determined for each Stock in the Huffman tree. The following function should then be called to print out the **names** of the Stocks than are contained in 'sentence':

**void displayStockList(Node<Stock>* rootPtr, const string& sentence);**

Note that the first argument is the root of the Huffman tree.

**THE DEPARTMENT STANDARDS FOR "STYLE GUIDELINES" SHOULD BE FOLLOWED IN ALL CODE.**