1. The HuffmanNode class does not contain a next method because HuffmanNodes do not relate to one another though the idea of next. The “next to elements are the two children, left and right. So to refer to the “next’ element would require calling one of them instead.
2. I used ArrayList because of the random access ability. When I built my tree I would sort it to find specific characters and because of how I stored my data I could access the frequency of different characters but referencing their character. Although a linkedlist may have saved time when I insert new HuffmanTrees and sort the list in my build tree class. But I also just like ArrayList better too.

Operation:

There is a main method in this program that can be used to test this code, but changing the filename in this method it can be checked.

To run step by step would be done by:

1. Creating an Encoder object
2. Putting the data in a map: toMap(filename)
3. Turn map into an array: makeArray(map)
4. Build tree from the array: buildTree(array)
5. Make to code from the array: makeCodeMap(tree)
6. Open a file to print to:
7. Print the code map to the file: printCode(newMap)
8. Some work is done inside HuffmanEncoding
9. Close the files

The space saved it at the bottom of the Encoded file. The encoding table precedes the code.