

## CS 1150 Design Notebook Required Sections

### Step 1: Problem Statement

This assignment will take in 3 files. One file will be used as the key and this key is used to decode the other 2 files and find the parrots' hidden message. This code needs 2 object classes, one for decoding and one for storing indexes needed to find the message. The KeyGridElement class will take in 2 numbers, one for row, the other for column and store them. In the other class, DecodingMachine it will take the key fill it as a 2D array and print it and will also use the key to index it will values from the 2 message files, storing it in an iterator. At the end the code will fill 2 queues with the index from the message files and take them to the decoding method and will display the 2 secret messages.

### Step 2: Understandings

- What I Know:
  - Objects
  - ArrayList, Queues
  - File Reading
- What I Don't Know:
  - Iterators

### Step 3: Pseudocode

Main:

- Create 3 File Object
- Create 3 Scanner Taking in Files
- Create a DecodingMachineObj
  - Takes In dimensions for 2D Array from keyGrid File
- Create an arraylist and fill it with characters from keyGrid File
- Change ArrayList into an iterator
- Call fillKeyGrid Method
  - Take the arraylist iterator and fills 2D array in DecodingMachine
- Call printKeyGrid Method
  - Prints 2D array
- Create two normal queues
- Call fillQueue Method
  - Fills queues with KeyGridElement Objects that store character index in 2D array
- Change queues to be an iterator
- Store and call decodeMessage Method
  - Used 2D Array and takes characters from index found with message files and store it in an ArrayList iterator

DecodingMessage:

- Takes in KeyGridElement Iterator
- Loops until the last object in given KeyGridElement Iterator, use hasNext() method
- Index 2D array will values within the KeyGridElement objects, use getter methods
- Add character found to decoded message variable
- Return a Character Iterator – Decoded Message

### Step 4: Lesson Learned

I coded the decodeMessage method as

secretMessage.add(keyGrid[msgIterator.next().getRow()][msgIterator.next().getColumn()]) and it errored because here I was making it do next twice in one loop meaning it errors when it reaches the other next because the loop only checks when it ends. This also would mean the index I was getting were wrong, to fix this I made a variable iterator2DIndex to hold the next object in msgIterator, making it so it only did next once for one loop and this worked.

## Step 5: Code

```
//package cs1450;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
/*
Isaiah Hoffer
CS1450 (M/W)
4/10/25
Assignment 8
This assignment will take in 3 files. Two files, the messages, have numbers that are indexes for a 2D
Array and the keyGrid file gives you the decoding
message used to figure out what the two messages say. This assignment will use iterators from
arraylists and queues. At the end display the deciphering key
and the two messages deciphered.
*/
public class HofferIsaiahAssignment8 {

    public static void main(String[] args) throws FileNotFoundException {

        //Creating Files
        File messageOneFile = new File("Message1.txt");
        File messageTwoFile = new File("Message2.txt");
        File keyGridFile = new File("KeyGrid.txt");

        //Opening Files For Reading -- Scanner
        Scanner readMessageOne = new Scanner(messageOneFile);
        Scanner readMessageTwo = new Scanner(messageTwoFile);
        Scanner readKeyGrid = new Scanner(keyGridFile);

        DecodingMachine decodingMachineObj = new DecodingMachine(

readKeyGrid.nextInt(), readKeyGrid.nextInt());

        //Creating ArrayList
        ArrayList<Character> keyGridList = new ArrayList<>();

        //Filling ArrayList
        while(readKeyGrid.hasNext()) {

            keyGridList.add(readKeyGrid.next().charAt(0));

        } //While
```

```
//Creating An Iterator For The ArrayList
Iterator<Character> keyGridIterator = keyGridList.iterator();

//calling fillKeyGrid Method
decodingMachineObj.fillKeyGrid(keyGridIterator);

//Displaying Key Grid Used To Decode Method
//Pretext
decodingMachineObj.printKeyGrid();

//Creating Queues To Store Secret Message
Queue<KeyGridElement> messageOneQueue = new LinkedList<>();
Queue<KeyGridElement> messageTwoQueue = new LinkedList<>();

fillQueue(messageOneQueue, readMessageOne);
fillQueue(messageTwoQueue, readMessageTwo);

//Creating Iterators From Queues
Iterator<KeyGridElement> messageOneIterator = messageOneQueue.iterator();
Iterator<KeyGridElement> messageTwoIterator = messageTwoQueue.iterator();

//Calling DecodingMessage Method And Store Returned Iterator
Iterator<Character> decodedMessageOne =
decodingMachineObj.decodeMessage(messageOneIterator);
Iterator<Character> decodedMessageTwo =
decodingMachineObj.decodeMessage(messageTwoIterator);

//Displaying First Message -- Decoded
System.out.printf("\n\nFirst Message Decoded:\n");
System.out.printf("-----\n");
while(decodedMessageOne.hasNext()) {

    System.out.printf("%c",decodedMessageOne.next());
}

//While

//Displaying Second Message -- Decoded
System.out.printf("\n\nSecond Message Decoded:\n");
System.out.printf("-----\n");
while(decodedMessageTwo.hasNext()) {

    System.out.printf("%c",decodedMessageTwo.next());
}

//While

//Closing Scanner
readMessageOne.close();
readMessageTwo.close();
readKeyGrid.close();

}

} //main

public static void fillQueue(Queue<KeyGridElement> queue, Scanner messageFile) {
```

```

        while(messageFile.hasNext()) {

            int row = messageFile.nextInt();
            int column = messageFile.nextInt();

            queue.offer(new KeyGridElement(row, column));
        }//While

    }//FillQueue Method
} //class

//Finds Location Of A Character From The Key Grid
class KeyGridElement {

    //Private Data
    private int row; //How Many Rows 2D Array Will Have
    private int column; //How Many Columns 2D Array Will Have

    public KeyGridElement(int row, int column) {

        //Initalizing Variables
        this.row = row;
        this.column = column;

    }//KeyGridElement Constructor

    //Getter For Row
    public int getRow() {

        return row;
    }//getRow Method

    //Getter For Column
    public int getColumn() {

        return column;
    }//getColumn Method

} //KeyGridElement Class

//Decode Message From Files
class DecodingMachine{

    char[][] keyGrid;
    int numRows;
    int numColumns;

    public DecodingMachine(int numRows, int numColumns) {

```

```

        //Initalizing Variables
        this.numRows = numRows;
        this.numColumns = numColumns;

        this.keyGrid = new char[numRows][numColumns]; //2D Array

    }//DecodingMachine Constructor

    public void fillKeyGrid(Iterator<Character> charIterator) {

        for(int row = 0; row < numRows; row++) {

            for(int column = 0; column < numColumns; column++ ) {

                keyGrid[row][column] = charIterator.next();
            }//For Column
        }//For Row

    }//FillKeyGrid Method

    //Decode Message
    public Iterator<Character> decodeMessage(Iterator<KeyGridElement> msgIterator) {

        ArrayList<Character> secretMessage = new ArrayList<>();

        //Goes Through Message File Indexes, Returns Character At Index Found Within
        while(msgIterator.hasNext()) {

            //Gets Next Object To get Indexes From
            KeyGridElement iterator2DIndex = msgIterator.next();

            secretMessage.add(keyGrid[iterator2DIndex.getRow()][iterator2DIndex.getColumn()]);
        }//While

        //Changing ArrayList To Iterator
        Iterator<Character> secretMessageIterator = secretMessage.iterator();

        return secretMessageIterator;
    }//decodeMessage Method

    //Prints Decoded Message
    public void printKeyGrid() {

        //Pretext
        System.out.printf("Decoding Machine's Key Grid: \n");
        System.out.printf("-----");

        //Printing Characters From 2D Array
        for(int row = 0; row < numRows; row++) {

```

Object

```
        System.out.printf("\n");
        for(int column = 0; column < numColumns; column++) {

            System.out.printf("%c",keyGrid[row][column]);

        }//For Column

    }//For Row

} //printKeyGrid Method
} //DecodingMachine Class
```