CS-207: Programming II
Spring 2016
Northeastern Illinois University
Research Lab: XOR Encryption
Due to D2L on Tuesday, 04/19/16 by 9:00 a.m.

Research Lab Goal

The goal of this research lab is to encrypt a file using the XOR encryption algorithm while using exception handling techniques described in your course. In addition, students will become more familiar with reference the Java 8 docs when working with new API classes. Please note that his lab is due to D2L by the above date and time and there will be **not** be an extended deadline to resubmit.

Getting Started

1. Create a new Java class named `XOREncryption` that has the `main` method.
2. You need one specific Java package imported in order to work with `File` objects. What is it? Import this package before your class definition.
3. Download the `data.txt` file from the NeededFiles.zip file and save it in the same folder as your Java file. Look at the data in this file - do you recognize it?
4. Create a .txt file named `key.txt` that has the following text: `This is a key`
5. Compile your code.

Creating and Using File Objects

1. Each file that you read from needs to be represented by a `File` object.
2. Create a `File` object for each of your files.
3. You will need to be able to read from each of your `File` objects using the `FileInputStream` class.
4. Find the constructor for the `FileInputStream` class in the Java 8 docs that takes a `File` object as a parameter. What type/name of the exception does it throw? Why do you need to know this before you create a `FileInputStream` object?
5. Create a `FileInputStream` object for each of your `File` objects.
6. Handle the `FileNotFoundException` by printing out `"No file."`
7. Compile your code.

Reading From a File

- The `read` method of the `FileInputStream` class takes an initialized byte array as a parameter.
- Create and initialize two byte arrays. How do you know the `length` of the arrays? (Hint: Look at the lecture!)
- The `length` method of the `File` class returns a `long`. However, in order to declare and create your arrays, you need an `int`. Cast the result of calling the `length` method to an `int`.
- Look at the Java 8 docs for the `read` method. What does this method return?
- Look at the Java 8 docs the `read` method again. Does it throw any exceptions, and if so, what is the name/type of the exception?
- Read from your `data.txt` and `key.txt` files and save the results to `int` variables.
- Handle the `IOException` by printing out `"Error reading file."`
- Compile your code.

Implementing the Encryption Algorithm

- To implement the algorithm, you need to iterate over the byte array created by reading from the `data.txt` file.
- For each byte in that array, you should do the following:
  - Find the index of the key byte array by taking the for-loop counter `mod` the length of the key byte array.
  - Encrypt the byte: (byte at for loop counter) XOR (key_byte at index found using `mod`)
  - Assign the encrypted byte back to the byte array (for the `data.txt` file).
- Hint: Look at the lecture to find the operator for XOR.
- Compile your code.

Writing the Encrypted Data

- Find the constructor for the `FileOutputStream` class in the Java 8 docs that takes a `String` as a parameter. What type/name of the exception does it throw and when does it throw it? Do you need to write another `catch` block for this exception?

- Create a `FileOutputStream` object and pass in the `String` `"data_encrypt.txt"` as a parameter to the constructor.
- Next, we need to write the encrypted byte array to using the `FileOutputStream` object and the method `write`.
- What type of parameter does the `write` method take and what does it return?

- Write the encrypted byte array.
- Close both of the `FileInputStream` objects and the `FileOutputStream` object.
- Compile your code.

Testing Your Code

- Run your code. A file named `data_encrypt.txt` should appear in the same location as your Java file.
- Open the `data_encrypt.txt` file and verify that the contents of the file look like nonsense.
- In your `XOREncryption.java` file, change the file that you are reading from to be `"data_encrypt.txt"` and the file that you are writing to be `"data_decrypt.txt"`. Do not change your `key.txt` file!
- Run your code. A file named `data_decrypt.txt` should appear in the same location as your Java file. This file should be exactly the same (with the same formatting) as your `data.txt` file.
- Try different keys and different text files to see how your encryption works!
- Submit only your `XOREncryption.java` file to the Research Lab #3 Dropbox on D2L.