

# CIS 36A :: LAB 12 - Input/Output

**Student Name:**

## **Task 1: Definitions & Concepts**

**Instructions:** Briefly answer the questions below.

Exercise 1: Byte streams are more convenient when handling binary bytes, where as character streams are obviously more convenient for ASCII characters

## **Task 2: Understanding Programming**

Intentionally left blank.

## **Task 3: Programming Exercises**

Instructions: Use any IDE to write and execute below exercises from the textbook. Attach Snipping photos of your source code and execution of the code in the console. Make sure to create separate files for each exercise.

**Chapter Examples:** Follow the lectures and do the programs with the instructor. Share the screenshots of the two below:

## CopyFile.java

```
*/
import java.io.*;
class CopyFile {
    Run | Debug
    public static void main(String[] args){
        int i;
        FileInputStream fin = null;
        FileOutputStream fout = null;
        // First, make sure that both files has been specified.
        if(args.length != 2) {
            System.out.println(x: "Usage: CopyFile from to");
            return;
        }
        // Copy a File.
        try {
            // Attempt to open the files.
            System.out.println(x: "Opening files");
            fin = new FileInputStream(args[0]);
            fout = new FileOutputStream(args[1]);
            do {
                i = fin.read();
                if(i != -1) fout.write(i);
                System.out.println(x: "Writing files");
            } |
            while(i != -1);
        }
        catch(IOException exc) {
            System.out.println("I/O Error: " + exc);
        }
        finally {
            try {
                if(fin != null) fin.close();
            } catch(IOException exc) {
                System.out.println(x: "Error Closing Input File");
            }
            try {
                if(fout != null) fout.close();
            } catch(IOException exc) {
                System.out.println(x: "Error Closing Output File");
            }
        }
    }
}
```

```
PS G:\Java 1\CIS36A\Lab11> javac CopyFile.java
PS G:\Java 1\CIS36A\Lab11> java CopyFile input.txt output.txt
Opening files
Writing files
PS G:\Java 1\CIS36A\Lab11> 
```

## ReadLines.java

```
1 // Read a string from console using a BufferedReader.
2 import java.io.*;
3
4 class ReadLines {
5     public static void main(String[] args)
6         throws IOException {
7         // create a BufferedReader using System.in
8         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
9         String str;
10        System.out.println(x: "Enter lines of text.");
11        System.out.println(x: "Enter 'stop' to quit.");
12        do {
13            str = br.readLine();
14            System.out.println(str);
15        } while (!str.equals(anObject: "stop"));
16    }
17 }
```

```
PS G:\Java 1\CIS36A\Lab11> & 'C:\Program Files\
eodor Obukhov\AppData\Roaming\Code\User\workspac
Enter lines of text.
Enter 'stop' to quit.
test
test
123
123
stop
stop
PS G:\Java 1\CIS36A\Lab11> 
```

## TRY THIS 11-1 A File Comparison Utility - Share your test runs only

```
G:\Java 1\CIS36A\Lab11>clear
'clear' is not recognized as an internal or external command,
operable program or batch file.

G:\Java 1\CIS36A\Lab11>java CompFiles CompFiles.java temp
Files are the same.

G:\Java 1\CIS36A\Lab11>java CompFiles CompFile.java temp
Files differ.

G:\Java 1\CIS36A\Lab11>
```

## Chapter Exercises:

### Exercise 7:

```
import java.io.*;
class ReplaceSpaces {
    Run | Debug
    public static void main(String[] args) {
        byte i[];
        FileInputStream fin = null;
        FileOutputStream fout = null;
        if (args.length != 2) {
            System.out.println(x: "Usage: CopyFile from to");
            return;
        }
        try {
            System.out.println(x: "Opening files");
            fin = new FileInputStream(args[0]);
            fout = new FileOutputStream(args[1]);
            i = new byte[fin.available()];
            System.out.println(x: "Writing files");
            fin.read(i);
            String temp = new String(i);
            int location = temp.indexOf(str: " ");
            String x = temp.substring( location, location + 3 );
            String replacement = "-";
            byte replacementBytes[] = replacement.getBytes();
            i[location] = replacementBytes[0];
            i[location + 1] = replacementBytes[1];
            i[location + 2] = replacementBytes[2];
            fout.write(i);
        }
        catch (IOException exc) {
            System.out.println("I/O Error: " + exc);
        }
        finally {
            try {
                if (fin != null)
                    fin.close();
            } catch (IOException exc) {
                System.out.println(x: "Error Closing Input File");
            }
            try {
                if (fout != null)
                    fout.close();
            } catch (IOException exc) {
                System.out.println(x: "Error Closing Output File");
            }
        }
    }
}
```

b11 > ≡ output.txt

1 test-123

```

import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
public class ReplaceSpacesChar {
    Run | Debug
    public static void main(String[] args) throws IOException {

        FileReader inputStream = null;
        FileWriter outputStream = null;

        try {
            inputStream = new FileReader(args[0]);
            outputStream = new FileWriter(args[1]);

            int c;
            while ((c = inputStream.read()) != -1) {
                if (c == ' '){
                    c = '-';
                }
                outputStream.write(c);
            }
        } finally {
            if (inputStream != null) {
                inputStream.close();
            }
            if (outputStream != null) {
                outputStream.close();
            }
        }
    }
}

```

Exercise 8:

```

1 test-123

```

## Exercise 18:

```
Lab11 > J NoTryWithResources.java > ...
1  import java.io.*;
2
3  class NoTryWithResources {
4      Run | Debug
    public static void main(String[] args) {
5          // First make sure that both files have been specified.
6          if(args.length != 2) {
7              System.out.println(x: "Usage: NoTryWithResources From To");
8              return;
9          }
10
11         try(FileInputStream fin = new FileInputStream(args[0]); FileOutputStream fout = new FileOutputStream(args[1])) {
12             if(fin != null && fout != null) {
13                 int c = fin.read();
14                 fout.write(c);
15             }
16         }
17         catch (IOException exc) {
18             System.out.println(x: "IOException: program halted.");
19         }
20     }
21 }
22
```

## Exercise 19:

```
Lab11 > J IfIntegerCheck.java > ...
1  import java.io.*;
2  public class IfIntegerCheck {
3      Run | Debug
    public static void main(String[] args) throws IOException{
4          InputStream is = new ByteArrayInputStream(new byte[] { 0, 1, 2, 3, 4, 5, 6 });
5          ByteArrayOutputStream buffer = new ByteArrayOutputStream();
6
7          int nRead;
8          byte[] data = new byte[4];
9
10         while ((nRead = is.readNBytes(data, off: 0, data.length)) != 0) {
11             System.out.println("here " + nRead);
12             buffer.write(data, off: 0, nRead);
13         }
14
15         buffer.flush();
16         byte[] targetArray = buffer.toByteArray();
17         for (int i : targetArray){
18             if (i == -889275714){
19                 System.out.println(x: "yes");
20                 break;
21             }
22         }
23     }
24 }
25
```

## Optional Exercises:

### Exercise 22: