**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

For each of the below questions, write a short sentence or two to express (in your own words) your answer. Keep the answers short, but use complete, correct, English sentences.

If it helps to clarify the questions, feel free to mentally prefix all the questions with the phrase "According to the video…"

1. After you’ve watched all the videos, please answer this question:  
   Of all the videos that you watched, if you could pick one video to be re-recorded by the instructor outside of class which would you choose? Why?  
   (Keep in mind the recording outside of class will omit any pauses from the instructor answering student questions, have less hemming and hawing, etc, and generally be more concise)

|  |
| --- |
| < Write your answer here > |

# **VIDEO: File I/O (Input) In C#**

## File I/O In C#

1. What does “I/O” mean?

|  |
| --- |
| < Write your answer here > |

1. What is **input**? Specifically, where is the data starting out, and where does the data move to?

|  |
| --- |
| < Etc. > |

1. What is **output**? Specifically, where is the data starting out, and where does the data move to?

|  |
| --- |
|  |

1. What are the three steps that you need to do in order to interact with the data contained within a file?

|  |
| --- |
|  |

1. If we want to move, or delete, or rename (etc.) an entire file, do we need to do the above three steps?

|  |
| --- |
|  |

1. Give an example of how the operating system can optimize I/O when **reading data out of a file**.

|  |
| --- |
|  |

1. Give an example of how the operating system can optimize I/O when **writing data to a file**.

|  |
| --- |
|  |

1. What does the word ‘directory’ mean?

|  |
| --- |
|  |

1. What is a path?

|  |
| --- |
|  |

1. How do absolute paths differ from relative paths?

|  |
| --- |
|  |

1. What is the program’s “current working directory”?

|  |
| --- |
|  |

1. When writing a relative path, what is different about the beginning of the path?

|  |
| --- |
|  |

1. Why is it important to tell Visual Studio what the current working directory is?

|  |
| --- |
|  |

1. When you run the program is run (WITHOUT setting the current working directory) what happens?  
   What does this tell you?

|  |
| --- |
|  |

1. How do you open up the data file in Visual Studio?

|  |
| --- |
|  |

## Manipulating Files

1. For all things that we’re going to talk about, what do you need to put at the top of your C# source code file?  
   **NOTE: This is also needed for the code that reads information out of the file (not just for deleting/renaming/etc)**

|  |
| --- |
|  |

1. What does the backslash character ( **\** )do, in general?  
   What does **\\** mean?

|  |
| --- |
|  |

1. What is a static method?

|  |
| --- |
|  |

1. Why is it useful for File.Exists to be a static method?

|  |
| --- |
|  |

## Reading Data From A File

1. Give a brief, intuitive description of what are binary files?

|  |
| --- |
|  |

1. When you open up a text file in Visual Studio (or Notepad, etc), what do you see?

|  |
| --- |
|  |

1. Give some examples of text files

|  |
| --- |
|  |

1. How can you check what data is in a text file?

|  |
| --- |
|  |

1. Both File.OpenText and new StreamReader accept a parameter. What is that parameter?

|  |
| --- |
|  |

1. Give an example of what might happen if your program don’t close a file that it has opened.

|  |
| --- |
|  |

1. What does the **using(...) { ... }** construct do for you?

|  |
| --- |
|  |

1. What is a ‘verbatim’ string? How do these differ from normal C# strings?

|  |
| --- |
|  |

1. What happens when your program tries to open a file that doesn’t exist?

|  |
| --- |
|  |

1. What is an exception?

|  |
| --- |
|  |

1. Describe a typical way for a program to react to an exception being thrown

|  |
| --- |
|  |

## Parsing Data In A File

1. What does it mean to ‘parse’ a file?

|  |
| --- |
|  |

1. What is a token?  
   What are they separated by? (Give specific examples of whitespace)

|  |
| --- |
|  |

1. List all the tokens on the following line:  
   **1 “John Smith”**

|  |
| --- |
|  |

1. Why are fixed-width / monospaced fonts useful when looking at text data files (and programming language files)?

|  |
| --- |
|  |

1. What does the ReadLine() command do?

|  |
| --- |
|  |

**It looks like I lost the drawings from slide 23 (“File Input Answer”) – sorry about that!**(I think that the program is simple enough that you should be able to follow along anyways)

1. What is “leading whitespace”?  
   What is “trailing whitespace”?

|  |
| --- |
|  |

1. What TWO THINGS happen when you give Int32.TryParse a string that contains a valid integer?

|  |
| --- |
|  |

1. How do Int32.TryParse, Double.TryParse, and Int16.TryParse differ?

|  |
| --- |
|  |

**Note that even though there aren’t a lot of questions about the example programs you should study these programs closely. They’re valuable because they can both explain how to use these concepts (and code) to solve problems and they provide useful models to follow when creating your own programs.**

1. If we want our program to read all the lines in the file (regardless of how many lines there are), how do we set up the program?  
   (You can answer this by briefly describing the differences between this program and the prior program)

|  |
| --- |
|  |

1. What does ReadLine() return when there’s nothing else in the file?

|  |
| --- |
|  |

1. If we want our program to ignore non-numeric tokens, how do we set up our program?  
   (You can answer this by briefly describing the differences between this program and the prior program)

|  |
| --- |
|  |

1. What is a ‘delimiter’ character?

|  |
| --- |
|  |

1. What does the .Split() method do?

|  |
| --- |
|  |

1. What does the ReadToEnd() method do?

|  |
| --- |
|  |

1. What is a downside to reading the entire file into memory all at once?

|  |
| --- |
|  |

# VIDEO: File I/O (Output) In C#

1. When talking about file output, where is the data going to?

|  |
| --- |
|  |

1. We’re going to be using StreamWriter objects to write to files, using several methods.  
   Which methods are these, and where have you seen (and used) these methods before?

|  |
| --- |
|  |

1. When printing to a file, will the output immediately be written to the file on the disk?  
   If not, where is that information being stored?

|  |
| --- |
|  |

1. When will the information be written to the disk?

|  |
| --- |
|  |

1. C# provides a way to ensure that the Dispose() method is called. Copy the example code from the slides and put in a brief comment explaining what each line does.

|  |
| --- |
|  |

1. Why will we NOT be using the C# using construct in these slides?

|  |
| --- |
|  |

1. If you create a new StreamWriter object (in order to write data to a file) and the file does NOT exist, what happens?

|  |
| --- |
|  |

1. If you create a new StreamWriter object (in order to write data to a file) and the file DOES exist, what happens?

|  |
| --- |
|  |

1. How can you check that your program has successfully written the data to the file?

|  |
| --- |
|  |

1. For the ‘Hours’ question, put in a quick explanation of what each part means:  
   *123 Kim 12.5 8.1 7.6 3.2*  
   E.g., explain what the ‘123’ means, what the ‘Kim’ means, etc.

|  |
| --- |
|  |

1. In the answer to the Hours problem, explain what this line does, and how it works:  
   *while( (sLine = input.ReadLine() ) != null )*

|  |
| --- |
|  |

1. Why is it necessary to check if count is zero?

|  |
| --- |
|  |