**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: BST: Print**

1. Prior to this point, what technique have we used for all the BST methods?

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

1. Can you (easily) traverse every node in the tree using a loop?
What abstract data type will you need to use?

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

1. What does the externally available, public version of the print method do?
(I.e., what responsibilities does it have?)

|  |
| --- |
|  |

1. Why can't we make the "internal", private version of Print public?

|  |
| --- |
|  |

1. In terms of the "onion model of recursion", what is the work that we're going to do at each level of recursion?

|  |
| --- |
|  |

1. In terms of the "onion model of recursion", what are the two recursive aspects of the Print algorithm?

|  |
| --- |
|  |

1. In the PrintR method, what is the significance of cur.Left not being null?
(Note that the video has the PrintR method mistakenly calling Print – PrintR should be recursively calling Print**R** instead)

|  |
| --- |
|  |

1. In the PrintR method, what is the significance of cur.Right not being null?

|  |
| --- |
|  |

1. In PrintR, why does the Console.WriteLine command (abbreviated at C.WL) go in between the first and second recursive calls?

|  |
| --- |
|  |

**VIDEO: BST: Patterns**

1. What are the three traversal patterns listed on the screen at the start of this video?

|  |
| --- |
|  |

1. What order will the "in-order traversal" traverse all the nodes?

|  |
| --- |
|  |

1. How would you traverse all the nodes in the reverse order (by "reverse", I mean the reverse of the order listed in the prior question)

|  |
| --- |
|  |

1. What is a good mnemonic (memory-aid/trick) to remember where to put the "do work" part of the in-order traversal?

|  |
| --- |
|  |

1. What would be one good usage of the "pre-order traversal" pattern?
Why is the pre-order traversal good for this task?

|  |
| --- |
|  |

1. Given the usage you listed in prior question (for pre-order traversal), why would the post-order traversal be a particularly bad choice for that same usage?

|  |
| --- |
|  |

1. How would you find the smallest value in the tree?
(This is demonstrated in response to a student's question, and it isn't really mentioned what's going on until near the end of the demonstration)

|  |
| --- |
|  |

**VIDEO: Recursive Add**

1. What is the first thing to check for when adding a new value to the BST?
If this is true, what do you do next?

|  |
| --- |
|  |

1. For this implementation, which class is the public AddR method located on? Where is the private, recursive AddR method located?

|  |
| --- |
|  |

1. What are the two basic options (in the recursive AddR method)?

|  |
| --- |
|  |

1. In terms of the "onion model of recursion", what is the small amount of work that can be done immediately? How is the remaining problem a smaller version of the overall problem (of adding a new value to the tree)?

|  |
| --- |
|  |

1. How do we know if we need to add the value to the left of the current node?
Explain this in English AND provide a snippet of C# source code.

|  |
| --- |
|  |

1. If you know that you need to add the node to the left, what are the two cases that might be true?
Explain this in English AND provide a snippet of C# source code.

|  |
| --- |
|  |

1. How can you check for duplicate values (i.e., that the value you want to add is already in the tree)?

|  |
| --- |
|  |

1. What is the one thing you should NOT do if your BST finds a duplicate value in the tree?

|  |
| --- |
|  |