**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: Passing an object to a method (objects as parameters)**

(It may be best to watch this video in the ‘fullscreen’ mode)

The video starts with an overview of the program that we’ll be using in this video; it’s great review up until about the 2:30 mark.

1. What is the overall goal of passing an object to a method?

|  |
| --- |
|  |

1. Within this particular program, what is meant by the “leftmost” point?

|  |
| --- |
|  |

1. What pattern do you follow to pass a pair of points into a method (such as PrintLeftMost)?

|  |
| --- |
|  |

1. On the call**ing** side, how do you pass (say) the **origin** and **near** points into the PrintLeftMost method?

|  |
| --- |
|  |

1. Why do we call PrintLeftMost(near, far) then immediately call PrintLeftMost(far, near)

|  |
| --- |
|  |

**VIDEO: Objects are passed (as a parameter) by reference**

1. In the example project the program first prints out the starting value of the **near** point, then calls the MovePointLeftwards command, then prints out the ending value of the **near** point.  
   What is this code trying to test / what question will this program help us answer?

|  |
| --- |
|  |

1. What does “pass by reference” mean?

|  |
| --- |
|  |

1. What does “pass by value” mean?

|  |
| --- |
|  |

1. What are some of the advantages of passing something (an object, an array) by reference?  
   What are some of the disadvantages/dangers of passing something (an object, an array) by reference?

|  |
| --- |
|  |

**VIDEO: Return an object from a method (objects as return values)**

(Note that the new material starts around the 2:00 mark – you can skip ahead to that point if you want)

1. Why is it bad to have your constructor ask the user for input?

|  |
| --- |
|  |

1. How do you declare (in C#) that your method will be producing/returning a Point object?

|  |
| --- |
|  |

1. How do you tell C# exactly which Point object you wish to return when the method is executing?

|  |
| --- |
|  |

1. Why will you get an error when attempting to call pt2.Print() prior to the pt2 = ptMaker.CreatePoint(); line?

|  |
| --- |
|  |

1. What is one very common error that you absolutely do not want to do?  
   Also make sure to explain briefly and intuitively why this is bad.

|  |
| --- |
|  |

1. When the CreatePoint method returns **localPoint**, what is it actually returning?

|  |
| --- |
|  |

1. What is the term for a block of memory that has no references pointing to it?

|  |
| --- |
|  |

**VIDEO: Class with an array of Double values inside it (GradeTracker)**

1. What is the major difference between the array that we previously looked at, and what we’re going to look at today?

|  |
| --- |
|  |

1. When creating a new GradeTracker object, what will it do (array-wise)?

|  |
| --- |
|  |

1. What will the PrintGrades method do?

|  |
| --- |
|  |

1. How do you declare an instance variable that is an array (of, say, doubles)? Where do you put that declaration?

|  |
| --- |
|  |

1. Why do you NOT need to do the numGrades = 0; line within the constructor?

|  |
| --- |
|  |

1. Does the declaration (from the previous question) actually create the array of **double**s?   
   If not, what line of code actually creates the array (since line numbers aren’t being displayed in the video you can just copy the line of code into this quiz):

|  |
| --- |
|  |

1. Briefly, intuitively summarize how the AddGrade method will work.

|  |
| --- |
|  |

1. In the PrintGrades method, why does the for loop iterate until i reaches numGrades instead of until I reaches grades.Length? What will this print out to the screen?

|  |
| --- |
|  |

1. In the example video, can you use the syntax **gt[0]** to access the grade in slot 0 within the GradeTracker object referred to by the **gt** variable?  
   (Why would it be bad to allow this?)

|  |
| --- |
|  |

(Again, make sure that you watch the remainder of the video so that you can develop an intuitive understanding of how the computer executes this code)

|  |
| --- |
| **Class with an array of Double values inside it (GradeTracker): Additional Demo Video**  There are no viewing quiz questions for this video.  This video explains the Histogram class which should save you time when working on the exercise that uses this class. |

**VIDEO: Class with an array of Object References inside it (PointTracker)**

1. In this video we’re going to create an array of references to objects that is located where?

|  |
| --- |
|  |

1. What does the PointTracker.Print method do? How many points will it print out the very first time that it’s called?

|  |
| --- |
|  |

1. Why is it very convenient to use the following line of code? How might it be superior to creating a local variable to hand to the Add method?  
   pt.Add( new Point( 10, 10 ) );

|  |
| --- |
|  |

1. In the following line of code, where should you NOT put a semi-colon?  
   (There are a lot of places where you shouldn’t put the semi-colon – make sure you list the one explained in the video because it’s a very common mistake that people make)  
   pt.Add( new Point( 10, 10 ) );

|  |
| --- |
|  |

(Again, make sure that you watch the remainder of the video (after the 8 minute mark) so that you can develop an intuitive understanding of how the computer executes this code)

**VIDEO: Class with an array of Object References inside it WITH NULL VALUES (PointTracker)**

1. What will this video deal with?

|  |
| --- |
|  |

1. What is different about this video?

|  |
| --- |
|  |

1. How does the Print method differ from the one shown in the prior video?   
   What effect will this have on the running program?

|  |
| --- |
|  |

1. Briefly, intuitively summarize (in your own words) how the new AddGrade method will work when you add the object referred to by the **origin** variable here:

|  |
| --- |
|  |