**Name: Jason Truong**

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…"

* 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)

|  |
| --- |
| I'm not a critical person; and it also worked, I have nothing of critique to say - my apologies. |

**VIDEO: How To Use My Videos**

* When viewing the videos in your web browser, where are the video-playback controls located?

|  |
| --- |
| Along the lower side bar side of the screen |

* List out at least three controls that you’ll find on the web page, and what each one does.

|  |
| --- |
| A thumb to drag along the video playback bar to scrub through the video and pick a place to start and stop on, a fullscreen button to enter fullscreen mode that will be replaced with a minize button once entered, and an audio controller to adjust the volume |

* How can you download the .MP4 video file (so that you can watch it in a media player program on your local computer)?

|  |
| --- |
| Download the videos through the lesson page and on OneDrive somewhere along the top should give you an option to download the video |

* List out at least three features that the VLC Media Player has, and what each one does.

|  |
| --- |
| An audio controller to adjust the volume, a hotkey system to quickly go for actions that would need to accessed through a menu or a mouse; all with just a press of the button, a thumb to drag along the video playback bar to scrub through the video and pick a place to start and stop on |

**VIDEO: What is a project?**

* How many files might a typical program be made of?

|  |
| --- |
| around dozens and dozens of files |

* What can you think of a project as?  
  What is the primary purpose of a project?

|  |
| --- |
| A bucket; storing a bunch of the files into one place |

* Give some examples of different types of files that you might store inside a project?

|  |
| --- |
| source code files, image files, maybe even video files |

* What is a Visual Studio Solution?

|  |
| --- |
| A further bucket; which can contain multiple smaller buckets, projects. |

* Give some examples of different things that a Solution might contain.

|  |
| --- |
| Installer projects, help file projects, other projects and such |

* THIS IS REALLY IMPORTANT:  
  When you’re working with a Project/Solution in Visual Studio, WHICH FILE SHOULD YOU OPEN?

|  |
| --- |
| Open the project file, .cs file, or the solution files |

* What problem will you run into if you open a C# file directly?

|  |
| --- |
| C# won't have all its neccessary files to run correctly. |

**VIDEO: How to create a simple console application**

* Briefly explain two separate ways to start the process of creating a new project.

|  |
| --- |
| At the top left of VS, select 'FILE', select 'New', select 'Project', |

* If you’re working at school and you have trouble getting your program to compile and run on the H: (network) drive, where should you try saving the project?

|  |
| --- |
| The C: Drive |

* How do you tell Visual Studio to display line numbers?

|  |
| --- |
| Along the top select 'Tools', then select 'Options', a new menu pops open, there find 'Text Editor', go into 'All Languages', underneath the sub-categories select 'General', then a box with text that says 'Enable Line Numbers' - yeah that one, click it. |

* What is the difference between “Start **With** Debugging” and “Start With**out** Debugging”

|  |
| --- |
| Starting without debugging with ask you to press any key before closing the program. Starting with debugging will stop on the red mark you can make along the line numbers, but will close the program if no mark is to be found. |

* How does Visual Studio indicate that your file has a compile-time/syntax error?

|  |
| --- |
| Syntax errors are with a red squiggly underline  A compile error is with the number line and can be found in the 'Error List' window that should open when started |

**VIDEO: How to download and use a simple console application**

* What is the key thing to do after you’ve downloaded the .ZIP archive?

|  |
| --- |
| Extract the files out of the .ZIP archive |

* If you don’t extract the files from the .ZIP archive and instead you open the C# file from within Visual Studio anyways (while the file is still inside the .ZIP archive) what problem will you run into?

|  |
| --- |
| VS won't read the other files in the folder to run the program |

* How do you get Visual Studio to display the Solution Explorer?

|  |
| --- |
| Along the top click 'View', then click 'Solution Explorer' |

**VIDEO: How Basic console I/O ("Everything you need to know for this class, and nothing more”)**

* What does Console.WriteLine do?

|  |
| --- |
| It prints text in a program window |

* How does Console.Write differ from Console.WriteLine?

|  |
| --- |
| .WriteLine puts itself on a new line to print its text, .Write doesn't |

* How does one produce a line of output in Java (and potentially in C#). Assuming that int x = 3; int y = 7; has been declared, list the code here:

|  |
| --- |
| int x = 3;  int y;  y = 7;  Console.Write("x: " x + " y: " + y); |

* What is the better way to print out variables in C#. Assuming that int x = 3; int y = 7; has been declared, list the code here:

|  |
| --- |
| Console.WirteLine("x: {0} y: {1}", x, y}; |

* When printing out variables, what does {0} refer to? {1}?

|  |
| --- |
| It reads it similarly to spaces in an array and takes the number in the first position {0} as x, which has been established as 3.  Then it continues to the second position {1} as y, which has been established as 7.  Then prints both |

* Before getting input from the user what should the program first do?

|  |
| --- |
| Prompt the user to input something and store it into a String variable |

* What is the line of C# code that will get whatever the user has typed?  
  (Make sure that your code stores that input into a variable)

|  |
| --- |
| string thisCouldBeAnything  thisCouldBeAnything = Console.ReadLine(); |

* What is the line of C# code that will convert the input from text into a integer?

|  |
| --- |
| Int32.TryParse(szInput, out x); |

* If the user types in a non-integer value, what will the value of **out x** be?

|  |
| --- |
| x = 0 as a result |

* What is the C# source code can you use to check if the user actually typed in an integer (and display a message either repeating that value, or telling the user that they didn’t type a number in)?

|  |
| --- |
| if (thisCanBeAnyVariable.TryParse(thisCouldBeAnything, out x); == true){  Console.WriteLine("The number you typed is: {0}", x);  }  else {  Console.WriteLine("You did NOT type a number!");  } |

* What is the C# source code that will attempt to convert user input into a real number (into a double value)?

|  |
| --- |
| double d;  if(Double.TryParse(thisCouldBeAnything, out d) == true) {  Console.WriteLine("The numbuh you typed is: {0}", d);  }  else {  Console.WriteLine("You did NOT type a number!");  } |

**VIDEO: Expression Evaluation (Order of operations)**

* Describe in your own words what the first thing that we do when evaluating an expression:

|  |
| --- |
| establish each variable |

* We then repeatedly do two steps.  
  What is the step 1? What is step 2?

|  |
| --- |
| figure out which operator goes next, keep track of the type |

* When you see a number like 3.0, what is it’s data type?

|  |
| --- |
| double |

* When you see a number like 3 (without the .0 / without anything after the decimal point), what is it’s data type?

|  |
| --- |
| integer |

* How do we figure out which operator goes next?

|  |
| --- |
| Follow the heirarchy of operations in the reference table for programming, then run Order of Operations for calculations. RESPECT WHAT TAKES PRECENDECE |

* Once we’ve identified which operator will be evaluated next,   
  what are the three steps in actually doing / evaluating an operator?

|  |
| --- |
| Figure out which operator goes next inside the parentheses  Convert Data Types  "Do" that operator |

* In the precedence table that was built for you, which operator goes first?

|  |
| --- |
| Parentheses |

* What does ‘left to right associativity’ mean?

|  |
| --- |
| When come across a set of operations, leftmost to rightmost takes priority |

* How is the assignment operator unusual?

|  |
| --- |
| It goes right to left instead of traditional lef to right |

* In the expression that gets evaluated in the video, what “operator” goes first?

|  |
| --- |
| (30.0[double] \* x[int] + y[double] \* 2[int]) |

* Within that thing that gets evaluated first, what is the first operator that we evaluate?

|  |
| --- |
| 30.0[double] \* x[int] |

* What is it safe to convert an integer into a double?

|  |
| --- |
| Yes, its safe to convert an int into a double because both are 32 bits in size |

* Can you put an assignment operator inside a larger expression?

|  |
| --- |
| Yes |

* In the video you saw many steps needed to evaluate the expression.   
  Does the computer actually do all these steps, or are these just for teaching purposes?

|  |
| --- |
| There are 7 steps and the computer takes each one into account. Its able to be utilized efficiently in debuggers to make coding less awful; and code more efficient |