**Name: Patrick O’Brien**

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)

|  |
| --- |
| All the videos with spontaneous coughing can be corrected. An example would be the first video for If/Else statements. |

**VIDEO: If, If/Else**

1. If you can’t remember the syntax for something in C#, you can try whatever you would have done in what other language (and it will probably (but not always) be right)?

|  |
| --- |
| Much of the logic is the same, but the syntax can differ highly between languages. |

1. Copy down the example of the if statement from the video (the one involving x, y, and printing “Yippee!”). For each part of the if statement put in a note about the syntax (just like the video does). This means that you should have a note about “if” being lowercase, etc.

|  |
| --- |
| Static void Main(string[] args)  {  Int x = 10;  Int y = 20;  If ( x < y)  {  Console.WriteLine(“Yippee!”);  }  } |

1. Why is it useful for us to move rightwards 1 tab stop for all the statements that are “within” the if statement?

|  |
| --- |
| To clearly define the statement from the rest of the code. Also it’s good practices for when using languages like python which considers tab stops in logic. |

1. What is a “block statement”? Why is it useful?

|  |
| --- |
| When you wrap everything in closed brackets {} so that it exacts all as one statement. |

1. For the following program, EXACTLY what will be printed and why?

int x = 100;

int y = 20;

if( x < y )

Console.WriteLine(“Yippee!”);

Console.WriteLine(“Yippee!”);

Console.WriteLine(“Yippee!”);

Console.WriteLine(“End Of program”);

|  |
| --- |
| It would not print in C# since the statement does not have a block to close multiple lines. |

1. How would you check if two variables are the same in C#?  
   How would you check if two variables are NOT the same in C#?  
   How would you check if one variable is less than another variable in C#?  
   How would you check if one variable is less than or equal to another variable in C#?

|  |
| --- |
| 1. By comparing them in a boolean check using the == operator. 2. By comparing them with the != operator 3. By comparing them with the < operator 4. By comparing them with the <= operator |

1. Can you put an “else” clause onto anything other than an “if” statement?

|  |
| --- |
| No |

**VIDEO: Switch**

1. In light of the fact that we will NOT be doing a lot of the ‘warming up’ exercises from classes like BIT 115 (i.e., no find-and-fix errors in the homework assignments, no tracing, etc, etc), what does the instructor HIGHLY recommend that you do for each new concept that you want to use BEFORE you try to use it in something like assignment 1?

|  |
| --- |
| Practice it |

1. Instead of using a switch statement, how could we use if/else statements to accomplish the same goal?

|  |
| --- |
| You can accomplish the same logic as a switch using if/else statement. However it much less efficient than a switch statement. |

1. Copy the example switch statement from the video (when it only has case 1 and case 2), then (briefly, intuitively) explain what will happen when the user userChoice has the value 2.

|  |
| --- |
| Switch(userChoice)  {  Case 1:  Console.WriteLine(“You typed 1”);  Break;  Case 2:  Console.WriteLine(“You typed 2”);  Break;  }  You typed 2 will be printed in the console. |

1. How can you have a switch statement do the same actions for when userChoice is either 0 or 1, WITHOUT duplicating code?

|  |
| --- |
| Using a default statement or a follow-through switch statement. However, a follow through switch statement is not supported by all versions C#. |

1. In C#, what data type do you typically use switch statements on?

|  |
| --- |
| Integers and strings |

1. What limitations are there on the values after the word **case**? Can you use variables (like **case x:**)? Can you specify ranges (like **case 1 – 10:** )?

|  |
| --- |
| Anything that you can figure out during compile time you can put in for case. For instance, 0+4 or 1-10. Basic operators. |

1. How do you catch the situation where none of the (other) cases match the variable that you’re switching on?

|  |
| --- |
| The default case. If nothing else matches then it will default and run the code in the default case. |

1. Is the break statement required at the end of every case (in C#), and if so, why?

|  |
| --- |
| No, follow-through statements are allowed in many languages. It is only required at the end of a statement. If the case has no statement it can continue to the next case without a break. |

**VIDEO: Integer Division**

1. Around the 2:20 mark the video explains how to do integer division. Briefly explain (in your own words) how to do integer division:

|  |
| --- |
| You can divide two integers using the / operator. So you could do int x = 20 int y = 200 and then Double result = x / y such as in the video. The final result is converted to the nearest integer. |

1. What is the result of dividing x by y? How/why do you get this result?

|  |
| --- |
| The result is 0, you get this because it simplifies to the nearest int. |

1. What triggers integer division?

|  |
| --- |
| When a int and a double are used when dividing. |

1. If you wanted to divide x by 100 (in order to get, say, a percentage of the form .2 (instead of the form 20, as in 20%) ), what result would you get and why?

|  |
| --- |
| Going from smaller to larger would result in 0 |

1. Why is it ok for the compiler to automatically convert an int into a double?

|  |
| --- |
| No information is lost, and accuracy can only be increased in the case for future problems. |

1. In C#, when does real division happen?

|  |
| --- |
| If there a decimal involved it will trigger the real division. |

1. How should you choose what data type each of your variables should be?

|  |
| --- |
| Your desired result should reflect the variables you use it achieve it. Usually you will be moving up from less accurate to more accurate variables. |

1. What will the expression **x / (double) r** be (around the 8:30 mark)

|  |
| --- |
| Make a copy of r that is a double and take x and divide it by the copy. 20 / 100.00 |

1. What are the two names for the process demonstrated in the previous question?  
   (Hint: both name begin with “type”, as in “type \_\_\_\_\_\_\_\_”)

|  |
| --- |
| Type cast, type conversion |

1. What is one common symptom of accidentally doing integer division?

|  |
| --- |
| Having a number being rounded, thus losing accuracy. If you have a double somewhere in the division process. |

**VIDEO: Modulus (How does it work, mechanically)?**

1. In C#, when you’ve got a literal number with a decimal part (such as 10**.0** ), what data type is it?

|  |
| --- |
| A double |

1. What problem does the following line of code have? After explaining what the problem is then re-write it so that it does not have that problem.  
   float x = 10.2;

|  |
| --- |
| Float num = 10.0f; |

1. Complete this sentence: “The modulus operator (or the \_\_\_\_\_\_\_\_\_\_ operator) gets me the \_\_\_\_\_\_\_\_\_ of doing integer division”

|  |
| --- |
| Modulo remainder |

1. In C#, what symbol is used for the modulus operator?

|  |
| --- |
| % |

1. What is the result of 21 % 10, and why?

|  |
| --- |
| 1 🡪 10 goes into 20 twice and the remainder is 1 |

1. What is the result of 17 % 4, and why?

|  |
| --- |
| 1 🡪 4 goes into 16 4 times and the remainder is 1 |

**VIDEO: Effective usage of the modulus operator**

1. One example of where the modulus / remainder operator is useful is figuring out how many items are left over after you pack equal-sized containers full of items. The video uses the specific example of packing 15 eggs into egg cartons (where each carton can contain up to 12 eggs).   
   Explain how to find out how many cartons are completely filled up, then explain how to find out how many eggs are left over after packing that 1 egg carton completely full:

|  |
| --- |
| How many cartoon completely filled = 15 / 12 = 1 carton  How many cartoons left after packing = 15 % 12 = 3 remaining eggs |

1. How can the idea from the prior question be applied to figure out exactly which bit (which slot) to examine in the third integer? How do you know you want to look at the third integer?

|  |
| --- |
| Using bitwise operators to look for the specific slot.  Dividing it by the amount of bits per integer? If you’re asking for 3 bits specific bit=3  Then you’d add 3 to the original even division. I’m not sure I understand the question exactly. |