1. Enumerated types are like lists, but not considered lists right? Correct.

Also, are objects usually the things we make like enumerated types, classes, methods.

A method is not an object. Every variable in C# is an object, but remember there are value-type objects and reference-type objects. Value-type objects are accessed directly by their value. Reference-type objects are simply pointers to the value – like pointing a mirror at something.

And what else would there be? There are namespaces, data types, variables and methods – that’s everything! There are an infinite number of data types, since we can create any data types we like: enums, structs, classes, and delegates. So far we have learned all except classes.

1. Why don't we need “static” in our class when we make enumerated types and structures? static is only used for methods and variables, not data type definitions. As we delve into structures, I can finally really show what static does.
2. When we create a bunch of source files I know that we use “partial” to be able to use the code within the other source files, but does the keyword “static” have to be there as well? Yes, because a class must either be static or not. You can’t have a single class defined as both static and non-static.
3. When we make new arrays the way we usually declare them is “string[] myArr = {values}” or “string[] myArr = new string[]”. I used strings specifically in this example, but when we make the array we have to stay consistent with the data type in terms of the second declaration example I used? Absolutely. Otherwise you will get a syntax error.
4. Which data types can we usually not use for an array? Can we use enumerated types, structures, and other objects within arrays as well? Every datatype can be used as an array.
5. The foreach loop can’t edit arrays, so how come it allowed us to add values to an array like we did in the example of copying elements from one array to another during office hours on Wednesday? Wouldn't that technically be editing an array (empty in this case)?

You can edit the array in a foreach() loop, but you cannot edit the variable that is used to read through the array: foreach( int thisInt in myIntArray )

You cannot change thisInt, but you can add and remove items from myIntArray while in the loop.

1. Spaces don't count as characters in C# right? Spaces are characters.
2. Does wrong scope count as a compile time error or run time? Compile time. The code won’t compile if the variable is not in the correct scope. For example:

{  
 {  
 int myInt = 0;  
 }  
 myInt = 42; // myInt is not in scope, the code won’t compile  
}

1. Are arrays stored in a block of memory as soon as we declare it or do we have to initialize it as well? And the values within that array are stored right next to each other within memory? int[,,,] myIntArray = int[10,10,10]   
   a block of 1,000 ints is created all initialized to 0. The values are all next to each other in memory.
2. I had questions from the arrays application from A-V (Each question followed by a “\*”), I made it so you could just write 10A,10B,etc. [myIGME-201/IGME 201 classwork/Week 2/Arrays\_new at main · kashahmed04/myIGME-201 (github.com)](https://github.com/kashahmed04/myIGME-201/tree/main/IGME 201 classwork/Week 2/Arrays_new) (Please let me know if the link is wrong)
3. Is initializing something to Int32 the same thing as saying int? Yes
4. Do we usually set the "points" for the goto (Ex. start or end) by any name and what are the capitalization rules for it? Labels don’t have any capitalization rules that I’ve seen. I usually see them camelcase.
5. For the math questions program, did you say that when we implement division it was preferable to convert the operands or the convert final answer into a double at one time?

You need to convert the operands, because if they are 2 ints then your answer will be an int regardless of converting the answer to a double.

1. For the number sorting application, in our first for each loop, why would we want to go back to the start (back to the start of the application) if one of the user's responses was not a valid number? Could we have just instead skipped that value and moved on within the loop by using “continue”? Why did we have to do that?

We didn’t have to, but I didn’t want the user to lose any data.

1. Why is it that we have to put a Console.ReadLine() right after a Console.WriteLine and not be able to write some code in between then put a Console.ReadLine()?

Normally you want to prompt the user so they know what to enter. It’s not clear what your question refers to.

1. I had written some questions for PE4 under the assignments section, and I got a grade back for the assignment, but my questions were not answered so I was wondering if you could answer those whenever you were free?

I completely forgot, thanks for the reminder!

1. For goto statements do we usually not put () after it and just put wherever we want to do or is there a case where we may need to put () around the goto?

Parentheses are option for goto, return and several other statements. I like putting parens around returns.

1. Do we not need a break statement if we use goto within a switch statement because when I said break, the font usually got darker meaning that it would not go to that statement.

You do not need the break in that case, but I would always add it in case the code changes in the future and it wasn’t clear whether it should fall through to the next case or not.

1. Say for example we had i = 5 and j = 6 if we made i = j, would the 5 associated with j be thrown out completely or would we still be able to use the original value of j somehow?

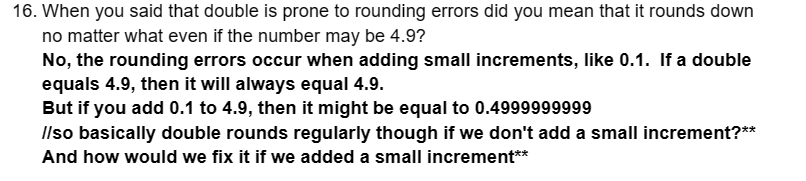
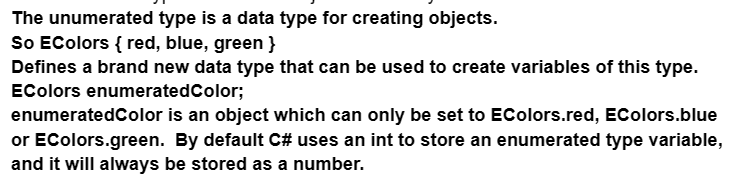
If you make i = j, then i will equal 6 and the original value of i is lost forever.

1. Are exams cumulative?

Yes

1. Did you mean “enumerated” here?

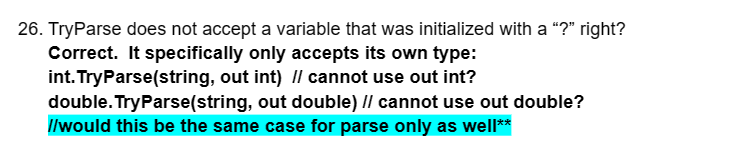
Yes



We fix it by always using Math.Round() when incrementing doubles

For implicit casting it's basically from less precise (8 bits) to more precise (16 bits) because it can hold all the data from the smaller 8 bits. But for explicit, we have to have a try catch and a checked to make sure if we lose data (ex. Going from 16 bits (more) to 8 bits (less) and some of the 16 bits are lost)?? I was still confused about this topic especially the idea of explicit casting and how to do it within a try catch.

I need a specific question here.





No, because you don’t pass an “out” variable into Parse.

1. For the keyword private, can stuff within the namespace (for example methods) have access to it even though its within another code block within the namespace (ex. if it's in the class program)?

Private variables and methods can be accessed within the same class, but not from a different class.

1. When we use “object” to pass as a datatype is it from a class we created or just a keyword to pass in anything into the variable?

Object is the most generic datatype and can be set to any variable.

1. If we have a break in the nested loop does it not exit the while loop and just the inner one?

A break in a nested loop will only leave the inner loop.

1. When we convert a string to an array the string is not lost right and the array and string are 2 different things?

The string is not touched or changed. Yes, the array and string are separate objects.

1. Can for each technically be used with strings since they are an array of characters. What else can they be used for?

foreach() can be used for any collection of data – arrays, strings, Lists, SortedLists, Dictionary, etc

1. Whitespace does not count as a character in c# right that's why it does not count in the length of a string

Whitespace counts in the length of a string. If s=” “ (5 spaces) then s.Length = 5

1. For the number sorting, why would we want to go back to the start if the user enters something that's not a number? Could we have just done a continue statement and go back to the loop condition and skip that element within the array for our index incrementer (the first loop that counts how many elements are in the unsorted array)?

We could do that, but then the user might lose data if they happened to type a number incorrectly.

1. For the number sorting, for the very last method about decreasing the size of the unsorted array don't we have to have a return statement to set the array of the unsorted array to the new array we just made so the value actually gets passed in?

We pass the unsorted array as a reference variable and the method changes the unsorted array at the last line of the method. Why did we return the lowest number in the find lowest value loop but not this one? Because we have passed in the number we want to remove. What if we want to sort in descending order?

1. Why do we need the while true within the last statement in the math questions because it breaks regardless of what we type if its not “y” and there's no continue so can we just not include the loop?
2. For the \n part basically we say if there is a \\n (\n character itself in the text file), then we replace it with an actual \n space character so it would be an actual space now instead of \n itself (mad libs homework)? We are replacing “\\n” with “\n” the actual newline character.
3. How can we define an array of ecolors from favcolorandnumberr? EColors[] aColors = new EColors[8];
4. Does split make a copy of the string into an array

Yes, it doesn’t change the original string at all.

1. For the number sorting, if it calls the method with ints instead of doubles for find lowest value, how could it do that because we initialize the array as a double so the elements can be doubles only then?

It will choose the method that matches the array type. Note there are 2 methods with the same name.

1. When we make our “markers” for our goto statements, what are the capitalization rules for it

There are no rules, just that camelCase looks better

1. In the math quiz can’t we just put \n instead of an empty writeline statement why did we use that instead? They both do the same thing
2. No matter how many “and” statements we have, if one of them is not true then it evaluates to false? Yes
3. No matter how many “or” statements we have if one is true then the whole thing is true?

Yes

1. In terms of the “and” condition within the loop to check difficulty in the math questions application, it only checks the first value right? So how does it know to stay in the loop?



1. For this image, if we add the range and if it's 10 for example, it would add the 10 in this case and not get a random number between it and then add? (math quiz application)
2. If we have a try catch and other stuff below it do we need a continue in the catch or will it go back up immediately?

It will not go back up immediately.

1. Why would we want to split in mad libs if it's already split and we replaced the \n with a space (the split we had before the loop for the specific the user chooses)?

We replace \n with the newline character, not a space. It isn’t already split.