Clannad episode 14 discussion

This side story evokes exactly the same feelings of clannad's common path: I felt calm but still full of silly situations and, best of all, Sunohara. Although this particular story has not achieved anything new and was not outstanding, I can see how many of us may like it as a nostalgic story, because almost all the characters appear, and a short story condenses both happy, peaceful moments; hectic, funny and disturbing, dramatic. You could even say that this episode is something like CLANNAD in a nut nut, so despite how simple it was and how it wasn't too much originality, I can't bring myself to hate in this episode. That's why I gave him 3. Some of the issues I saw were, apart from the lack of originality, the fact that part of the drama with Akio in recent days was almost too forced and its ending was all too obvious. However, it can be argued that it was the creation of some kind of comedy, but in my opinion, if it were, this idea is quite weak in the value of comedy. In addition, the end of the episode is a bit too sudden (if you do not take into account the fact that the next episode follows immediately). And, even though it's not just a disadvantage, I still missed some good old baseball games that we can find in key works. While you'd think I had a fairly negative opinion about this side story, Meeting furukawa bakers certainly had some funny gags like Fuko's call, a cappella singing YO-SHI-NO!, and even a mention to our beloved Saitou (probably Jet Saitou). While it's not too ambitious, this story has definitely made me smile a few times, and that's something i won't deny. What I hate about the argument is that it always interrupts the discussion. -G.K. Chesterton We continue moyatori controversy program, and this time, I'm actually on time. This week's quick deals are on nontrivial issues, how I approach controversy, whether my style fosters discussion on polarizing topics, and how to find a balance between being clear and achieving the depth needed to adequately cover the topic. In an interesting coincidence, I'll be able to do a live demonstration of the approach I'm taking this week's Controversy, which is exciting: a few evenings ago, the order of precedence, the order in which arithmetic expressions are evaluated based on operandy, started trending on Twitter after a user submitted the phrase 6 ÷ 2 (1+2) and claimed that such a phrase was insolubility due to ambiguity. In this post, I will step through the rationale for the answer, approach it from the point of view of computer science and examine the problem in the context of Moyatori to reflect on how my experience and style, affects the way I solve problems (and thus how to deal with controversial topics). The aim of this it is to give readers a sense of how I can ensure my posts can cover the details that I try and at the same time remain accessible to readers from different backgrounds. In today's discussion, the versatility of my style will be put to the test as I try to explain why the approach I have taken towards solving the semi-viral phrase is appropriate. This post is a little long, so for the Moyatori controversy, I'll answer questions here, elevator-style pitch, for visibility: I write very almost as much as I used to for my magazine publications and thesis. My blogging voice varies: in paragraphs, I try to be neutral and stick to what I see in a given job. In character captions I have a little more fun and I speak more freely. My bias depends on what I write. If I write about anime, I will be biased: my goal is fun, and therefore I try to emphasize the positives. I will continue to try to examine all sides of the arguments where possible and mention any advantages they may have. My approach in 2) ensures that it covers both sides, although I will spend more time presenting evidence of the site I am on. With that in mind, I changed perspectives because I've read more before; a well-written piece can and convinced me to see other perspectives beforehand. I use visualization and lean heavily on my screenshots/character captions to help me. The words themselves are dense and can be very dry, but signing an image can help people gauge my reaction to something easier. With answer hints for people in a hurry, I can almost delve into the example of how to deal with controversy. Before I do this, I would like to extend a special thank you to Rose of Wretched and Divine and Nabe-chan from Geek Nabe for encouraging me to do something more exciting and fun on this post. The issue of order In mathematics, operator precedence (or order of operations) is a method used to determine which arguments are executed first in a given expression. Such principles exist precisely to eliminate ambiguity: in general, mathematics is a discipline that deals with quantity, structure, space and change. To ensure consistency, students are taught a set of rules to help them calculate the value of expressions earlier in their education: parentheses are calculated first, then exponents, and then multiplication or division occurs, with subsection and last addition. This is a relatively simple set of rules that provides students with an understanding of operator precedence, and when performing an order of operations, a single-line expression such as 6 ÷ 2 (1+2) can be evaluated as follows: The explanation is quite simple: we first perform a subexpress phrase in parentheses. Total 1 and 2 is 3. Subsequently, using the operator, which states that multiplication and division takes place in the order in which the we rate 6 ÷ 2. This results in 3 as well. The resulting expression, 3 x 3, is calculated as nine. The only assumption is that we are dealing with a single-line operation, and here comes the alleged ambiguity: are we to interpret the phrase 6 ÷ 2 (1 +2) as a rational expression? Let's say we interpret the original statement as rational. Then, the evaluation yields the following: One seemingly simple expression, represented in a different way, gives a completely different result when evaluating through. This is where the crunch of the controversy lies: which of these ultimate values are correct? Basic mathematics is, after all, deterministic: given that we maintain consistent parameters, the result should always be the same. In this case, ambiguity has seemingly created a situation in which mathematics is open to interpretation. To remove this ambiguity, mathematicians define a so-called grouping symbol in which some symbols, such as a horizontal fractional line, form distinct groups that can be resolved. In the case of our expression in this case, assuming it is rational (a / b for all integers a, b and b ≠ 0), the numerator and denominator can be treated as a group. The counter is calculated separately from the denominator, and the final value is the quotient of the numerator and denominator. The horizontal fractional line is critical: its presence clearly indicates that we have two groups of operations, and as such the end value, 1, is correct, provided that we actually had a horizontal fractional line. However, because the expression 6 ÷ 2 (1 + 2) was displayed on a single line, there is no grouping symbol. Mathematics is about being explicit and there is no symbol to explain that 6 ÷ 2 (1 + 2) was in fact a rational expression with counter 6 and denominator 2 (1 + 2). As such, if there is a rational expression, 6 ÷ 2 (1+2) is a simple, one-line expression to be evaluated from left to right, exactly as it seems. Swift Solution To test this, I implemented a simple evaluator in three different languages: Python, Java and Swift. The full code is below and I invite readers to give it a whirlwind. Python code can be copied to a .py file and run using the python command at the command line (e.g. python MathSolver.py), while in the case of Java, the code will have to be copied to a .java file, compiled using javac (e.g. Javac MathSolver.java), and then run. Finally, I write Swift code for playgrounds, which provides a way to run Python-style code without having to open a new Xcode project. By the way, everything I use has the complexity of time and space O(1): it works at a fixed time, and everything is guaranteed to stop after evaluating expressions. Python Example def solve(): result = 6 / 2 * print(result) solve() Java Example of public class MathSolver { public static void main(String[] args) { int result = solve(); System.out.println(result); } static int solve() { int result = 6 / 2 * (1+2); return result; } } Swift Example func solve() { let result = 6 / 2 * (1 + 2) print(result) } solve() In each language, the result was the same: the expression 6 ÷ 2 (1 + 2) is calculated 9. The reason this happens is that the compiler (for Java and Swift) or interpreter (in Python) reads the expression as a single-line expression and takes precedence over the operator to calculate the final result. Compilers typically don't make decisions about more complex operators, and in languages like Java, Swift, or Python, they simply evaluate expressions from left to right. It is the case that different programming languages can interpret the same expression differently depending on how it was designed, so the programmer is responsible for writing out the expressions themselves in the correct order if they are intended to make the expression have a specific meaning. This is not the fault of the engineers who designed the above-mentioned programming languages: the purpose of a high-level programming language is simply to enable the programmer to provide instructions to the computer in a legible manner (low-level languages or assembly languages, provide much greater control at the expense of readability and ease of use), and as such programming languages will interpret the instructions exactly as they are provided. As such, if someone were to indicate that 6 ÷ 2 (1 + 2) was in fact a rational expression, it would be necessary to write a poem as 6 ÷ (2 (1 + 2): additional brackets eliminate any ambiguities and clearly indicate that we intend to treat 2 (1 + 2) as the denominator. Not surprisingly, all three Python, Java and Swift rate 6 ÷ (2 (1 + 2)) to 1. Below I will give you an example of what clean code looks like if you were to write a 6 ÷ rating function (2 (1 + 2)) in Swift. Although less concise, each line clearly indicates its functionality. Swift Example of good practice in readability func solvecrational() { let counter = 6 let denominator = 2 * (1 + 2) let the result = counter / denominator print (result) } solve() And further remarks that a good programmer will not only unseal expressions on one line: readable code is crucial for maintenance and extensibility. While this may require more lines and come at the expense of breech, readability has many benefits. In this example, to make it clear, beyond any debate, that I was dealing with a rational expression, I would like to express a rational expression with separate variables for each numerator and denominator, and then return the quotient. Otherwise, if no other context is provided, the compiler does not make any assumptions about the and just evaluates it. In the example above, I have not passed any parameters or done any work to evaluate any rational expression. It's quite simple, but beyond the scope of this discussion. Computer programs are only as good as the developers who write them, and as such, a faulty program is inevitably the responsibility of the person who writes the code: if someone wanted 6 ÷ 2 (1 +2) to evaluate to 1 in a programming language that supports similar to Swift, Java or Python, it is expected that, at least, add additional parentheses to clearly indicate their intentions; otherwise, what the programmer has on his hands is a semantics error in which the program produces an output contrary to what was expected because the said programmer did not properly communicate his intentions. Handling the controversy Elegantly I have now roughly defined how I achieved my solutions, my understanding of how an alternative solution is achieved and provided justification as to why the alternative solution requires a huge subjective leap that makes it incorrect in its current context. After posting this on Twitter, I was originally met with resistance. From explaining why the computer is not correct, to ill-formed evidence and even ad hominem attacks, it turned out that people insisted that their calculations were correct, often times without proper explanation. The best effort was to try to prove that 6 ÷ 2 (1 + 2) = 1, where the person worked on the way to the solution, but had the opposite error in his reasoning. A reverse error occurs when, given $\forall x, P(x) \rightarrow Q(x)$, q(a) is assumed, which gives them P(a). In the solution attempt, the person here assumed that from, they had Q(a) (i.e. 6 ÷ 2 (1 + 2) = 1), therefore they had P(a) (6 ÷ 2 (1 + 2) = 6 ÷ 2 [(1 + 2)]) as well. This is an incorrect way of reasoning (one that I informally call supposing what you want to prove is true). I have counter-evidence using one of the more entertaining forms of reasoning, called proof of contradiction. In this case, 6 ÷ 2 (1 + 2) = 1, so that 6 ÷ 2 (1 + 2) = 6 ÷ [2 (1 + 2)]. By using algebra to simplify both sides, we end up with the following contradiction: Because it is clear that 1/6 is not equal to 6 (which is absurd if it were true), we have established our contradiction, and therefore the assumption is false. I have shown that these two expressions are different and therefore not equivalent. Proof of contradiction is one of the most fun and powerful forms of reasoning, and this is a pure, elegant way of showing my point of view. Other arguments were that Apple engineers are wrong that I live a sad life... trying to work out this problem like 8 years old, but it [sic] actually much cooler than I litteraly [sic] changed the equation... [because I'm] trying to seem smart, but it took pre calc to see that you're wrong. On Twitter, I am not able to respond to those that are due to restrictions on character and formatting, but here where I have all the space I need, specifically and definitely showed in not one, but two different ways, that there is no ambiguity here. Some results are due to how clearly their expression is defined, and there is one reality: if someone does not have an eye for details and pays attention to things, they will inevitably get results that seem correct, but are in fact wrong. Such errors can be fatal in the system: individual errors may not seem like such a big deal, but in a complex system with multiple equations and multiple calculations, the propagation of errors means that the more errors, the worse the system will perform. I'll leave it up to the people above to decide if it makes me as the equivalent of 8 years old who apparently didn't take the pre-bill. The takeaway lesson is therefore not that mathematics is open to interpretation, because there may be ambiguity, but rather that in mathematics and computer science there are specific symbols and syntax that is used to define semantics. The compiler, not a human being, does not deduce the meaning of the user and as such, it is the responsibility of the writer to explain their meaning. Therefore, the whole debate on Twitter served only to illustrate one thing: that there is a nontrivial belief that mathematics can be considered liberal art: open to debate, semantic and interpretation. The origins of this belief are outside the scope of this document, but I believe that the reason why mathematics and science in general can be counted on is that it is a discipline dependent on quantitative means. Unlike liberated art, where personal origins, beliefs and other intangible elements are at stake, there are certain things that simply cannot be argued with in mathematics: these laws and thesis form an undeniable framework that can be used to solve more complex, sophisticated problems. If these foundations cannot be agreed, there would be no basis for seeking increasingly exciting solutions to solving difficult and important problems. A short version of this post, referring to issue 6 ÷ 2 (1+2), is as follows: the answer is 9, and I have outlined my reasons why this is true. Given what I presented above, I can conclude that 6 ÷ 2 (1 + 2) = 9 with quod erat demonstrandum (equivalent to the mathematics of microphone drops; more formally it is Latin for what was to be demonstrated). Personally, I think the limited number of Twitter characters really prevents discussions like these, and that may be one of the reasons why it seems to be happening every other day now: without enough space to establish context, things get misunderstood more often. There is another comment I have about this post; it took a total of about an hour to forumate the ideas and then an additional four hours to put it actual post together, including equations and code snippets; the hard part was formatting the post, so everything looked nice to readers. With this post in the books, I'm going to go back to doing what I do best: relax with the anime community on Twitter and get ready for Canon, which we're again, as well as blasting the bad guys in The Division 2, which I'm sure my opponents would find more enjoyable than if I were to give them a personalized demonstration of what school graduates makes their enemies. In this week's controversy, I presented a discussion that roughly indicates how I would solve the polarizing question outside the anime realm. In this case, I believe that I am not biased in examining both sides of the argument to conclude that I have done so. In something like computer science or mathematics, bias can be harmful, so I always try to make sure that I understand what the other side is saying and why they say it: there are circumstances where I'm sure of my position, and there are others where the other side is correct because of a misunderstanding because of me. I do not believe in defending something wrong and I can be persuaded to accept the other party's statements if there is a well-founded and evidence-based explanation. In anime discussions, things are more subjective, and I admit that I can become attached to my own theories. But even here, I don't see any other perspective: it's about being in the anime community to see what others are saying. The advantage of this is the opportunity to see what are the experiences of different nations through how they watch, and enjoy their anime. Finally, Moyatori's discussion raises the question of how to ensure people walk away with a good idea of what I'm saying, even if they don't venture into the arcane realm and start drawing terminology and phrases from my old textbooks. The solution to this problem is quite simple: I use many screenshots and it seems to me to be a little more casual in the accompanying subtitles. Here in the drawing captions, I crack jokes, explain myself with more conversational terms, and use screenshots as a context for what I say. In this post in particular, I used visual examples to demonstrate my thought process (i.e. show my work). I believe that throughout the life of this blog, there is one that has allowed me to cover various topics, from anime, games, movies and all the aristocracy themes in between without creating inconsistencies. Inconsistency.