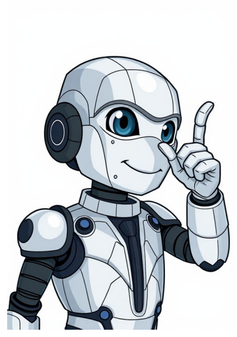
Continue



Go to fitbit r/fitbit • The notations $\cong$ and $\simeq$ are not totally standardized. Both are usually used for "isomorphic" which means "the same in whatever context we are." For example "geometrically isomorphic" usually means "congruent," "topologically isomorphic" means "homeomorphic," et cetera: it means they're somehow the "same" for the structure you're considering, in some senses they are "equivalent," though not always "equal:" you could have two congruent triangles at different places in a plane, so they wouldn't literally be "the same" but their intrinsic properties are the same. I've seen colleagues use both for isomorphic, and some (mostly the stable homotopy theorists I hang out with) will use $\cong$ for "homeomorphic" and $\simeq$ for "up to homotopy equivalence," but then others will use the same two symbols, for the same purposes, but reversing which gets which symbol. The $\approx$ is used mostly in terms of numerical approximations, meaning that the values in questions are "close" to each other in whatever context one is working, and often it is less precise exactly how "close." Topologists also have a tendency to use $\approx$ for homeomorphic. The main take-away from this answer: notation is not always standardized, and it's important to make sure you understand in whatever context you're working. I've never developed in C++. I just have a Visual Studio 2019 (primarily C) project given to me that includes a C++ file, and this file is importing symbols from a library that the linker can't find. I've written an example file that is basically: #include #include extern "C" { int ExampleFunc(int starting_num) { std::vector myvec; for (int i=starting_num; i------ Build started: Project: Example, Configuration: Win64_DLL x64 ------ 1> Creating library C:\git\Cpp\F2\Example_Code\Builds\Win64_DLL\Example.lib and object C:\git\Cpp\F2\Example_Code\Builds\Win64_DLL\Example.exp 1>Example.obj : error LNK2019: unresolved external symbol __imp__invalid_parameter referenced in function "void * __cdecl std::_Allocate(unsigned __int64)" (??$_Allocate@$0BA@U_Default_allocate_traits@std@@$0A@@std@@YAPEAX_K@Z) 1>ProblemFile.obj : error LNK2001: unresolved external symbol __imp__invalid_parameter 1>Example.obj : error LNK2019: unresolved external symbol __imp__CrtDbgReport referenced in function "void * __cdecl std::_Allocate(unsigned __int64)" (??$_Allocate@$0BA@U_Default_allocate_traits@std@@$0A@@std@@YAPEAX_K@Z) 1>ProblemFile.obj : error LNK2001: unresolved external symbol __imp__CrtDbgReport 1>AppCfg\Win32_Cfg\FVCANSing.lib : warning LNK4272: library machine type 'x86' conflicts with target machine type 'x64' 1>node_modules\@some-scope\some\path\htxtrc_vx_near.lib : fatal error LNK1107: invalid or corrupt file: cannot read at 0x3388 1>Done building project "Example.vcxproj" -- FAILED. ========== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped ========== I think a standard library would be something that I don't have to think about and it's packaged with the compile pipeline such that by default, I can just use it and not worry about it. But it's definitely anything that `std::` that produces these linker errors. I don't know how to configure the project to find these standard libraries in Visual Studio 2019 or if that is actually my problem, which is why I'm here. I've searched for this error and the best I'm getting is the `__imp__` part of the linker symbols indicate these are to be linked against a dynamic library but what the linker finds is a static one. However, I don't even know where these libraries live or how to configure them. EDIT I wonder if this is also related to the way extern "C" section is laid out... Maybe it's being done wrong and causing the wrong symbols to be used... EDIT 2 Setting Use Debug Libraries to Yes made the error go away in Project Properties > Configuration Properties > Advanced > Use Debug Libraries. The most common expression is just $\Bbb R\setminus\Bbb Q$. When a single letter is used, in my experience by far the most common is $\Bbb P$, though I have on very rare occasions seen $\Bbb I$. (Note, though, that $\Bbb I$ is also occasionally used to denote $[0,1]$.) If the context were clear enough, you could probably get away with using $\Bbb P$ without comment, but it would be much safer (and more courteous) to define the symbol explicitly. In topological contexts (including descriptive set theory) the irrationals are often denoted by ω^ω (or occasionally $\Bbb N^{\Bbb N}$), since in the topology that they inherit from $\Bbb R$ they are homeomorphic to the product space ω^ω; here no further comment is required. Go to AppleWatch I lost my charger recently and had to buy a new cube for the cord. It’s an apple brand bought at walmart. I’ve never seen this symbol before and I don’t think it’s charging? Please help! Page 2 Go to AppleWatch Perdí mi cargador hace poco y tuve que comprar un cubo nuevo para el cable. Es de la marca Apple, comprado en Walmart. Nunca antes había visto este símbolo y no creo que esté cargando. ¡Por favor ayuda! This is probably the stupidest thing I've ever needed help with, but I've read you can format sign text to different colours using § codes but I cannot for the life of me figure out how to put that symbol into minecraft. I've tried copying it, alt code 21, alt code 0167, they all just fail to put anything on screen. How do I colour my sign text? Further information: I'm playing Java on windows 10 on a PC Edit: In an absolutely BIZARRE combination of button presses, I can get something to happen. If I PASTE the § symbol with Ctrl+V, THEN press Alt+0167, the window will accept my next input as though it's a modifier for the § symbol despite not actually showing the symbol. So if I press CTRL+V, ALT 0, 1,6,7, release ALT, and then press 6, the text I type on the line will be gold. However, when I finish typing the message into the sign, the sign appears with gold text briefly, but then reverts to black text. What in the actual hell? So, in the definition of what is a square root, $\sqrt{x}$ are all numbers y such that $y×y=x$. are there any logical mathematical symbols so that the above definition can be written using logical operators only, and no natural language? Where can I get some introductory or reference material on all such logical symbols? update: I noticed, some time after asking the question that the definition of square root I am giving is wrong. The square root of x is to defined to be the non-negative number y that satisfies $y*y=x$. But the question was about notation, not square roots, so I am leaving it as it stands due to some answers using the supplied (erroneous) definition. I would like to add to Henricus answer as it gives the answer for a integer range but not for ranges of general step sizes, which is something that is very common in programming. The expression $$\{s k \mid k \in [a..b] \}$$ would give you a range from sa to sb with increments of s. So if you wanted to write an expression for the numbers $$\{-3,-2.5, -2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3\}$$ that could be written as $$\{\frac{1}{2} k \mid k \in [-6..6] \}.$$ $\begingroup$ Whats the meaning of this symbol? Its a three dot symbol: ∴ I read a book, im could not find any definition of this symbol. This is about continuum property of the natural numbers and the archimedean property: for some $n\in\mathbb{N}$, $n>B-1$ ∴ $n+1>B$ this should be a proof on the set $\mathbb{N}$ of natural numbers is unbounden above. But I do not understand it. An answer on how the three-dot symbol is what I am out after. Additional explanation of the proof would be nice to know as well, but not needed. $\endgroup$ 2 ∈ (mathematics) means that it is an element in the set of… For eg…x ∈ ℕ denotes that x is within the set of natural numbers. The relation "is an element of", also called set membership, is denoted by the symbol "∈". Writing {\displaystyle x\in A} x\in A means that "x is an element of A". Equivalent expressions are "x is a member of A", "x belongs to A", "x is in A" and "x lies in A". The expressions "A includes x" and "A contains x" are also used to mean set membership, however some authors use them to mean instead "x is a subset of A". Another possible notation for the same relation is {\displaystyle Ai x,} Ai x, meaning "A contains x", though it is used less often. The negation of set membership is denoted by the symbol "∉". Writing {\displaystyle xotin A} xotin A means that "x is not an element of A".