Subject: Re: Capture division by zero
Posted by jjacksonRIAB on Mon, 25 Jul 2022 01:37:48 GMT
View Forum Message <> Reply to Message

Quote:
The calculation of your bank account balance is infinite

Anyone have an opinion on this one? I'm of the opinion that floats should not be used to represent any kind of currency or bank balance and I tended to replace them in C# with Decimal when I worked for an accounting department. I haven't checked but does U++ have a fixed-point decimal type for dealing with currency and other numbers where float accuracy is not good enough? If not maybe one should be made.

EDIT Just realized I didn't provide a reason why not to use them. What I found is it wasn't NaN-stuff (though that could be a problem), but mainly because floats have a tendency to accumulate errors over successive calculations - that and the fact that you might want Banker's rounding or some other kind of rounding algorithm. Another reason I ditched floats is because they can't produce consistent results on different architectures, different CPUs, different families of CPUs, or even the same CPU. In one particularly egregious case I had a bad video driver produce wildly inaccurate results on a single accountant's machine and I could only guess that some of the computation was being offloaded to the GPU.

Looking further into C#'s decimal type, it appears it uses arbitrary precision floats underneath. There are number of libraries for doing this in C++, some listed here:
https://en.wikipedia.org/wiki/List_of_arbitrary-precision_ar ithmetic_software

Since they are still floats it's not a cure all for the problem of cumulative errors so I was wrong about that. That said, I still think a fixed-point decimal class would be desirable.