Reviewer Name: Mina Akbari
Reviewed Name: Rylie Byers

Code coverage analysis:

Method Name	Code coverage	Proposed test(s) to include
Num::equals	100%	No additional tests needed, already fully covered.
Num::interp	100%	Check the interpretation of a Num object with a value of 0.
Num::has_variable	100%	No additional tests needed, already fully covered.
Num::subst	100%	Test the subst method of a Num object, so that it returns a new Num object with the same value.
Add::equals	70%	Test case for nullptr
Add::interp	100%	Test the interpretation of an Add expression containing variables.
Add::has_variable	100%	No additional tests needed, already fully covered.
Add::subst	100%	Test to replace variables with other expressions/nums
Mult::equals	70%	Test case for nullptr , negative values, zero values
Mult::interp	100%	No additional tests needed, already fully covered.
Mult::has_variable	100%	No additional tests needed, already fully covered.
Mult::subst	0%	Test substitution of variables in both operands. Test substitution in Mult instances where the variable to be substituted is present in only one of the operands.
Var::equals	80%	Test case nullptr
Var::interp	0%	Exception handling: Test a case where Var::interp() throws an exception.
Var::has_variable	100%	No additional tests needed, already fully covered.
Var::subst	100%	No additional tests needed, already fully covered.

Thoughts / suggestions to improve the code or the tests:

Great job on writing tests! Your testing approach is clear and thorough.