Reviewer Name: Rylie Byers

Reviewed Name: Mina Akbari

Code coverage analysis:

Method Name	**Code coverage**	**Proposed test(s) to include**
AddExpr::equals(Expr * e)	Covers variables, positive numbers, positive numbers and variables,	Do one with negative numbers, zeros
MultExpr::equals(Expr* e)	Covers variables, positive numbers, positive numbers and variables,	Do one with one negative number, and do one with two negative numbers, zeros
VarExpr::equals(Expr* e)	Positive numbers, varable and positive numbers,	Do one with capitalization, and different strings like “@!x&”
AddExpr::interp()	Positive numbers, positive and variables, variables, zeros	Negative numbers
MultExpr::interp()	Negative negative, positive , variables,	Do a negative and positive one
VarExpr::interp()	Throw test, numbers, variables	Do one with capitalization, and different strings like “@!x&”
AddExpr::has_variable()	Variables, numbers and variables,	Do one with capitalization, and different strings like “@!x&”,zeros
MultExpr::has_variable()	Variables, numbers and variables,	Do one with capitalization, and different strings like “@!x&”, zeros
VarExpr::has_variable()	Variables, numbers and variables,	Do one with capitalization, and different strings like “@!x&”, zeros
AddExpr::subst(**const** std::string& varName, Expr* substreplacement)	Positive, positive and variables	Negative, zeros
MultExpr::subst(**const** std::string& varName, Expr* replacement)	Positive, positive and variables	Negative, and zeros
VarExpr::subst(**const** std::string& varName, Expr* replacement)	Variables, positive numbers, null ptr	Do one with capitalization, and different strings like “@!x&”, zeros

NumExpr::interp()	Positive, negative, zero, edge case	Pretty good coverage
NumExpr::has_variable()	Zero, positive, negative, decimals,	Pretty Good Coverage.
NumExpr:: subst(**const** std::string& varName, Expr* replacement)	Positive, variables	Zeros
NumExpr::equals(Expr* e)	Positive, positive and variables	Negative and zeros

Thoughts / suggestions to improve the code or the tests:

There are a lot of good tests. Already has super good coverage. Just incorporate more 0 and negative tests.
The organization by method for tests makes it easy to look at and find certain tests.

Add rows when necessary.