Reviewer Name: Tlakkireddy

Reviewed Name:Tina Chen

Code coverage analysis:

Method Name	Code coverage	Proposed test(s) to include
Add::equals(Expr * e)	42	
Add :: Add(Expr *numberOne, Expr *numberTwo)	64	
Add :: interp()	8	
Add :: has_variable()	2	
Add :: subst (string var, Expr* randExpr)	3	
Multiply::equals(Expr * e)	33	
Multiply :: Multiply (Expr *numberOne, Expr *numberTwo)	40	
Multiply :: interp()	5	
Multiply :: subst (string var, Expr* randExpr)	0	CHECK((new Multiply(new Variable("x"), new Number(7))) ->subst("x", new Variable("y")) ->equals(new Multiply(new Variable("y"), new Number(7))));
Multiply:: has_variable()	3	
Number :: equals(Expr * number)	84	CHECK_FALSE((new Number(7))->equals(new Variable("x")));
Number :: Number (int value)	158	
Number :: interp()	24	
Number :: has_variable()	6	
Number :: subst(string var, Expr* randExpr)	3	
Variable :: Variable (string val)	21	
Variable :: equals(Expr *var)	6	CHECK_FALSE((new Variable("x")) -> equals(new Number(7)));
Variable :: interp()	1	
Variable :: has_variable ()	2	
Variable :: subst(string var, Expr* randExpr)	6	CHECK((new Variable("a")) ->subst("b", new Add(new Variable("c"),new Number(9))) ->equals(new Variable("a")));

Thoughts / suggestions to improve the code or the tests:

The test overall is complete; we just need to pay more attention to the cases when the variable passes in is not the same type.

Add rows when necessary.