**Reviewer Name**:

**Whitney Kenner**

**Reviewed Name**:

**Mack Tawa**

**Code coverage analysis**:

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Code coverage** | **Proposed test(s) to include** |
| Num::equals(Expr \*e) | return false; line coverage | Testing num expression equality where they are not equal |
| Num::has\_variable() | return false;  line coverage | Testing if a num has a variable |
| Var::has\_variable() | return true;  line coverage | Testing if a vraible has a variable |
| Var::equals(Expr \*e) | return false;  line coverage | Testing variable equality where the result is false |
| Var::interp() | throw std::runtime\_error( "no value for variable" );  line coverage | Testing that trying to interpret a variable throws an error |
| Add::has\_variable() | return (this->lhs->has\_variable() || this->rhs->has\_variable());  line and branch coverage | Testing that an add expression has a variable. This should be tested for both the left hand and right hand side with and without variables (all 4 branches) |
| Mult::has\_variable() | return (this->lhs->has\_variable() || this->rhs->has\_variable());  line and branch coverage | Testing that an mult expression has a variable. This should be tested for both the left hand and right hand side with and without variables (all 4 branches) |
| Mult::equals(Expr \*e) | return false;  line coverage  branch coverage | Test that a multiplication equality returns false. Test this for both sides of the mult expr |
|  |  |  |

**Thoughts / suggestions to improve the code or the tests**:

|  |
| --- |
| Make sure to test false return statements for each method |
| For equality methods be sure to check all branching options |
|  |

Add rows when necessary.