**Reviewer Name**:Junming Jin

**Reviewed Name**: Jason white

**Code coverage analysis**:

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Code coverage** | **Proposed test(s) to include** |
| Num::equals(Expr \*o) | 3% | CHECK((new Num(3))->equals(NULL) == false); |
| Variable::equals(Expr \*o) | 5% | CHECK((new Variable("string"))->equals(NULL) == false); |
| Add::equals(Expr \*o) | 3% | CHECK((new Add(new Num(3), new Num(3)))->equals(new Add(new Num(3), new Num(4))) == false); |
| Mult::equals(Expr \*o) | 4% | CHECK((new Mult(new Num(2), new Num(2)))->equals(new Mult(new Num(2), new Num(3))) == false); // |
| Mult::subst(std::string str, Expr \*o) | 4% | CHECK( (new Variable("x"))->subst("y", new Num(2))->equals(new Num(2)) == false); |

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

|  |
| --- |
| The code is commendable, and I was able to attain a remarkably high coverage prior to performing a review. However, it is imperative to acknowledge the requirement for improvement in the approach to writing tests, by taking into account a more comprehensive evaluation of various paths and potential scenarios. This methodology would facilitate in ensuring comprehensive coverage of the software. |
|  |
|  |

Add rows when necessary.