**Reviewer Name**: Yijun Zhan

**Reviewed Name**: Tailang Cao

**Code coverage analysis**:

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Code coverage** | **Proposed test(s) to include** |
| bool Add::equals(const Expr\* other) const | 87%-97% | CHECK( (new Add(new Num(2),new Num(3)))->equals(new Add(new Num(2),new Num(3)))==true );  CHECK( (new Add(new Num(2),new Num(3)))->equals(new Add(new Num(3),new Num(2)))==false );  CHECK( (new Add(new Num(-2),new Num(-3)))->equals(new Add(new Num(-2),new Num(-3)))==true );  CHECK( (new Add(new Num(-2),new Num(0)))->equals(new Add(new Num(-2),new Num(0)))==true ); |
| bool Mult::equals(const Expr\* other) const | 97%-100% | CHECK( (new Mult(new Num(2),new Num(3)))->equals(new Mult(new Num(3),new Num(2)))==false );  CHECK( (new Mult(new Num(2),new Num(3)))->equals(new Mult(new Num(2),new Num(3)))==true );  CHECK( (new Mult(new Num(-2),new Num(-3)))->equals(new Mult(new Num(-2),new Num(-3)))==true );  CHECK( (new Mult(new Num(-2),new Num(0)))->equals(new Mult(new Num(-2),new Num(0)))==true );  } |

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

|  |
| --- |
| 1. Increasing Code Coverage:   For Add::equals, the code coverage of between 87% and 97% suggests a certain portion of conditions or branches within the method that are not being tested. Consider adding more tests that covers the edge cases as well as exceptional cases. |
| 2.Test Cases Completeness:  Ensure that the tests cover all possible equivalence classes and boundary conditions. This often includes testing with zero, negative numbers, and the maximum and minimum possible values.  Add tests for null pointers or invalid arguments to ensure the method handles such cases gracefully, if applicable. |
| 3.Test Documentation:  You can add more description for each of your test that clarifies what aspect of the method is the case testing, as well as putting test with the similar purposes into same group. This increase the readability and makes it easier for future adjusting when required. |