

Design and distribution

This is a requirements design document, the requirements have already been specified in the problem statement.

The whole project is to be executed in c++ programming language along with unit tests.

A main driver program will evaluate the test cases based on assertions.

The flow diagram of the INTAL_C is as follows:-

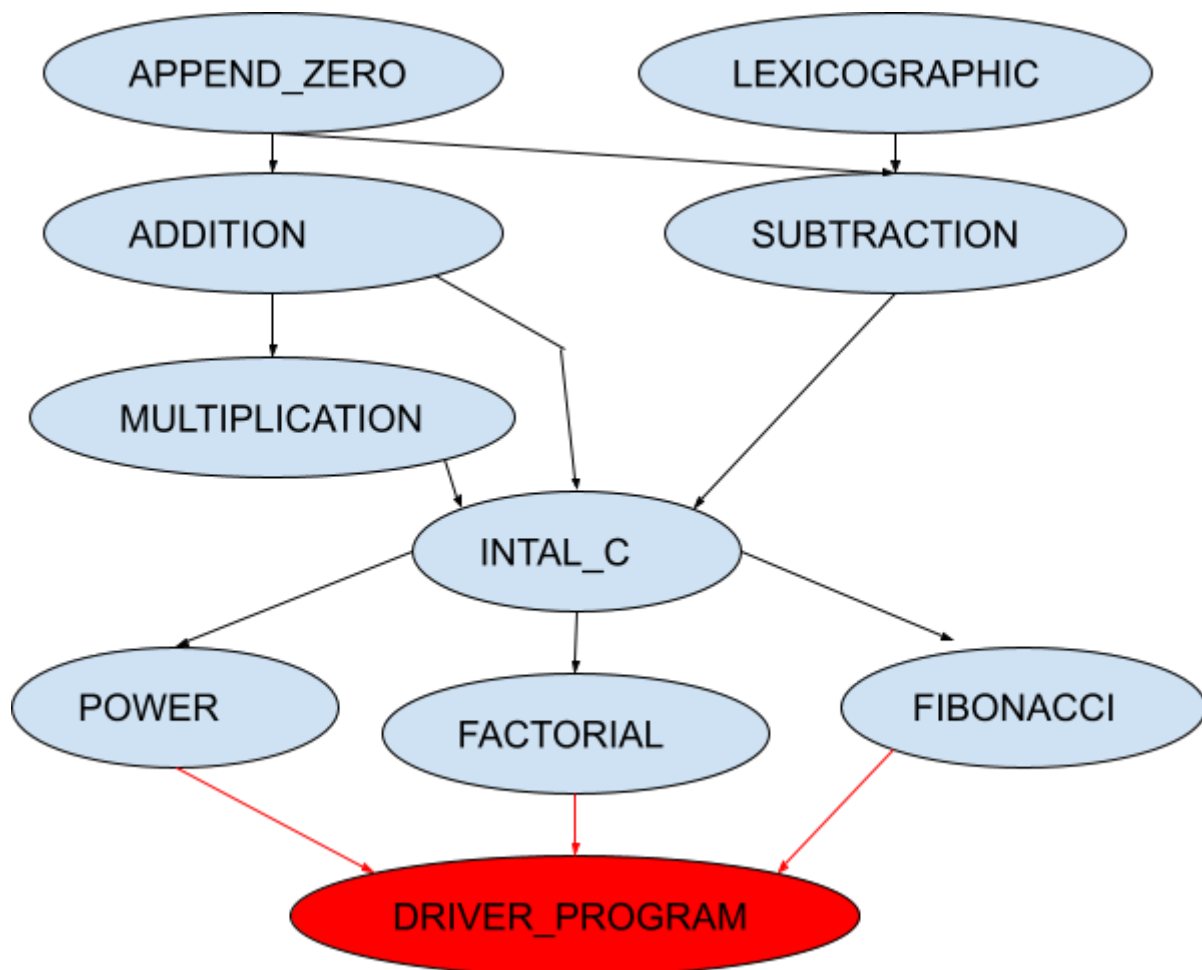


Figure 1

Each arrow highlights an inheritance hierarchy. Proper emphasis is given on as much abstraction which can be achieved in every class along with single responsibility principle.

The last three arrow lines are not inheriting but the three classes are imported for testcase run as with the main driver program being highlighted in red.

Some key features

The main emphasis is based on computations on three cases.

- Operation with other INTAL_C object
- Operation with any number represented as string.
- Operation with any number represented as a long long int.

The INTAL_C custom made library has many operators being overloaded hence standard relational operators of **<, > and ==** can be used with other INTAL_C datatype object with strings and even with long long int.

Assignment operator is also overloaded where an INTAL_C datatype object can be assigned a value from another INTAL_C datatype object, a string or even a long long int datatype.

The main operations high in the classes **ADDITION, SUBTRACTION and MULTIPLICATION** are also done using overloaded operators **+, - and *** along with all the three cases mentioned earlier and it can be used just like the integer primitive datatype in C++.

Highlights of the three main classes inheriting from INTAL_C:-

- The INTAL_C library can compute powers to a large exponent, one of the test cases computes to a power of 132. Moreover it computes the power in logarithmic time.
- The INTAL_C library can compute factorials to a large extent, one of the testcase asserts to a correct result for 100 factorial.
- The INTAL_C library can compute fibonacci numbers to a large computations.