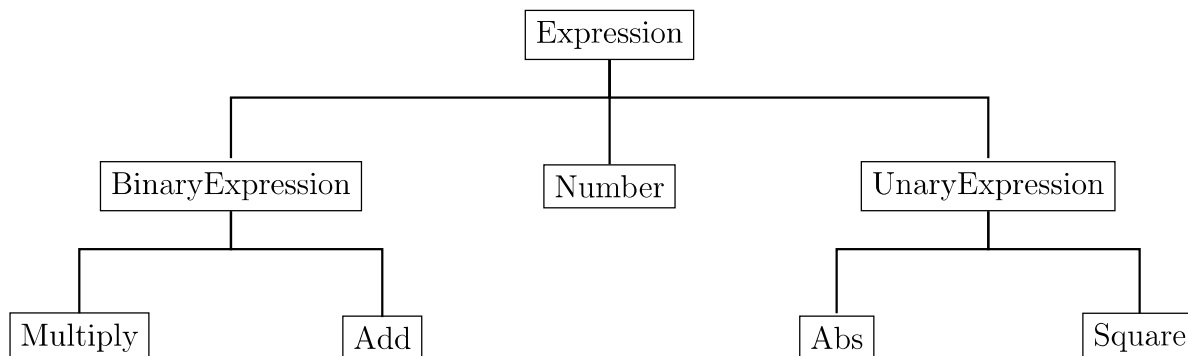


2 Part : Expression class hierarchy

50%

In this exercise you will implement storing and evaluation of simple arithmetic expressions using inheritance. Expressions are either multiplication, addition, absolute value, square, or just an integer value. The hierarchy of expressions is shown in the figure below.



Number represents expressions that are integer values, **Multiply** represents multiplication of two Expressions, **Add** represents sum of two Expressions, **Abs** represents the absolute value of an Expression, and **Square** represents the square of an Expression.

All expressions should support the following functions:

- **type** : returns the name of the class,
- **print** : returns a string representation of the entire expression, as shown below,
- **evaluate**: returns the integer result of evaluating the expression.

The result of the function **print** should be as follows.

- For an object of class **Multiply** with sub-expressions **subexp1** and **subexp2**:
(subexp*subexp)
- For an object of class **Add** with sub-expressions **subexp1** and **subexp2**:
(subexp+subexp)
- For an object of class **Number** with value **n**:
n
- For an object of class **Abs** with sub-expression **subexp**:
Abs(subexp)
- For an object of class **Square** with sub-expression **subexp**:
Square(subexp)

Above, the sub-strings **(,), *, +, Abs, Square** are part of the output format, and do not depend on the contents of the expression. You should not add spaces or new lines.

As in Part 1, you should add the **virtual** keyword where appropriate and then implement the functions where they are most appropriate. **Expression**, **BinaryExpression**,

and `UnaryExpression` could be abstract classes but all other classes should be concrete classes. You should add the necessary member variables and function implementations. The skeletons of the classes are less complete than in previous assignment, as you now have to choose where to implement each function, where add the `virtual` keyword, etc.

A detailed description of each function is given in the file `Expression.h`.