Union/ Heterogeneous Data Structures

```
union IntDouble
{  int ival;        // public members
   double dval;
};
```

```
IntDouble x;
x.ival= 9; x.ival= x.ival + 5; ...        // x as int variable
x.dval= 34.56;   // overwrite int value
x.dval = x.dval + 3.4 // x as a double
x.ival= 5;  .... x.dval+ 2.7 ..   // disaster
```

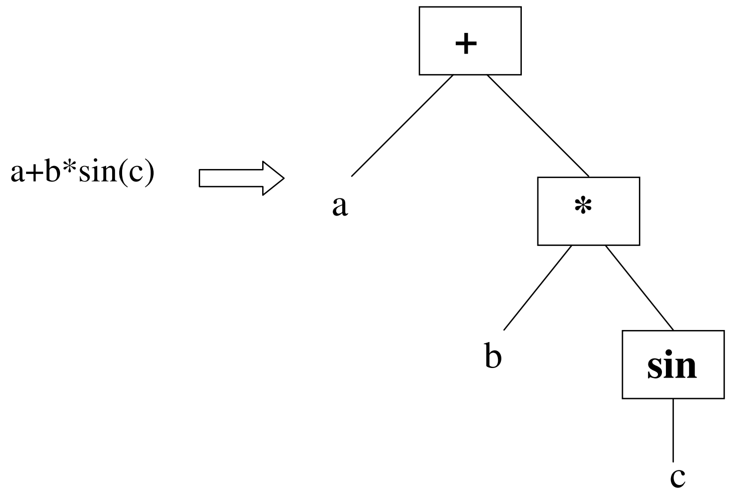
- A union object is a struct object that stores all its data-members at the same location (offset zero relative to the beginning of the object).
- A union object holds only one type of value at a time.
- A union object is given enough space to hold the largest of its data members.
- The member notations x.ival and x.dval are used not to get different offsets, but to get different interpretations of data stored at the same location.
- It is your responsibility to access a union object correctly to retrieve the most recently assigned value.

Symbolic expressions

Terminal symbols are:

+, -, *, /,sin, cos, exp, log, (,), variables, constants

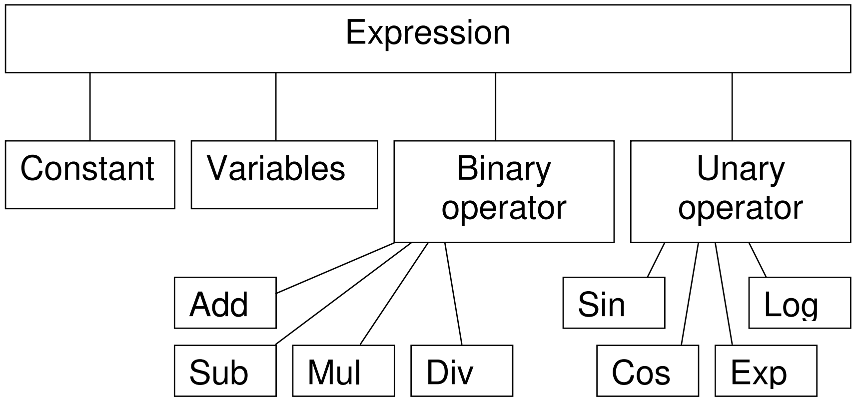
Consider writing a symbolic expression as a tree:



Every terminal symbol is part of the expression, the operators have either one or two arguments.

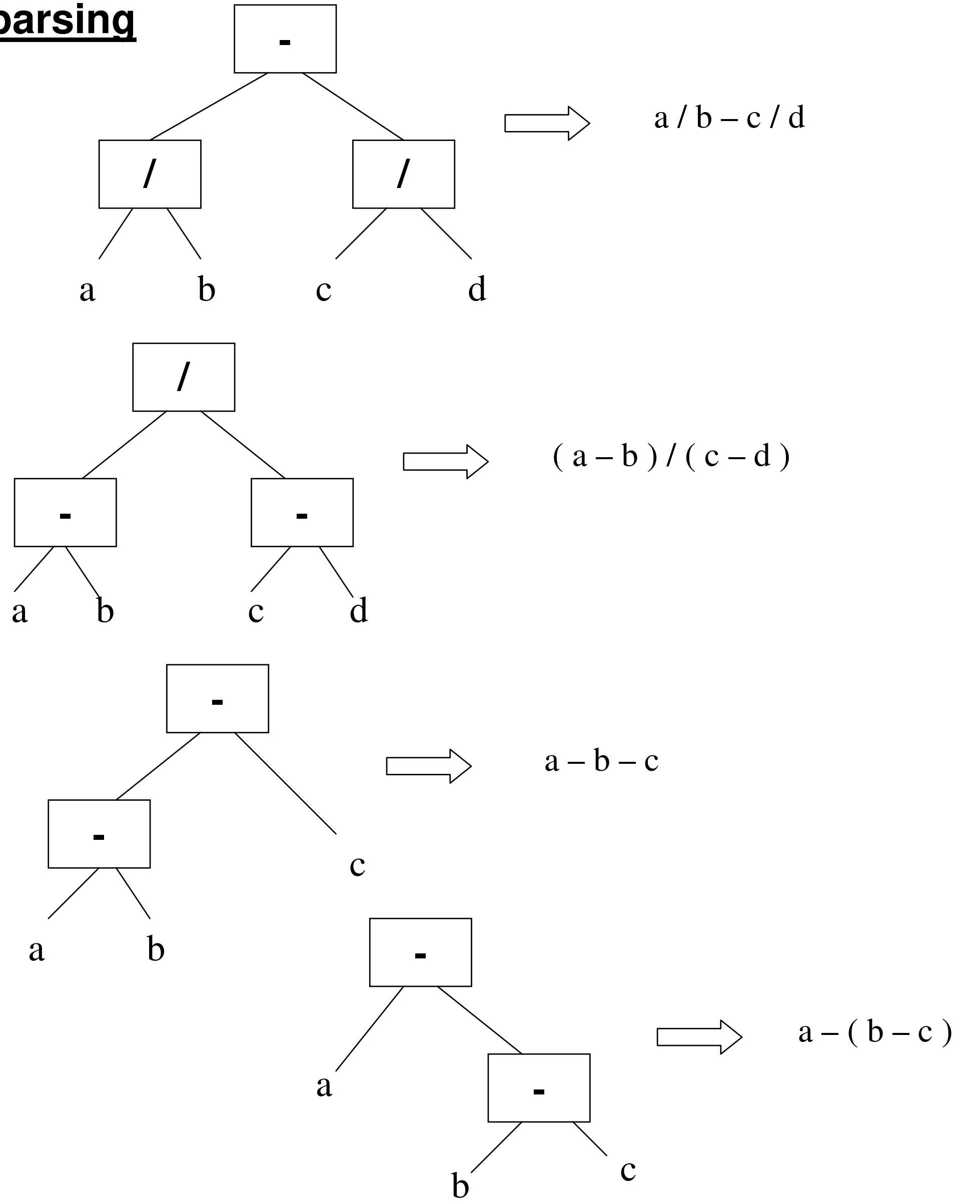
Symbolic expressions

One way of making inheritance:



Common for all expressions is that they can be evaluated, shown, differentiated and so fourth, such methods should be declared in an abstract base class, “Expression” in the above.

Unparsing



Compare change in operator priority and placement of parenthesis.