Rational implementation using functions:

```python
def rational(n, d):
    def select(name):
        return n if name == 'n' else d
    a = select('a')
    b = select('b')
    return a / b

def numerator(x):
    return x('n')

def denominator(x):
    return x('d')
```

List comprehensions:

```python
[<map exp> for <name> in <iter exp> if <filter exp>]
```

List & dictionary manipulation:

```python
[b for a in b if a]  # a and b are lists
```
Recursive description:
- A tree has a root label and a list of branches.
- Each branch is a tree.
- A tree with zero branches is called a leaf.

Relative description:
- Each location is a node.
- Each node has a label.
- One node can be the parent/child of another.

Recursive decomposition:
- E.g., count_partitions(6, 4)

Recursive calls:
- count_partitions(6, 3)
- count_partitions(2, 4)

Built-in instance function:
- is_leaf: returns True if a class has a single instance attribute.

Assignment statements with a dot expression on their left-hand side affect:
- If the object is an instance, then assignment sets an instance attribute.
- If the object is a class, then assignment sets a class attribute.

Python object system:
A new instance of that class is created.
- The __init__ method of the class is called with the new object as its first argument (named self), along with any additional arguments provided in the call expression.

Class Account:
- __init__(self, account_holder):
  - self.balance = 0
  - self.account_holder = account_holder
- deposit(self, amount):
  - self.balance = self.balance + amount
  - return self.balance
- withdraw(self, amount):
  - if amount > self.balance:
    - return ‘Insufficient funds’
  - self.balance = self.balance - amount
  - return self.balance

Function call:
- all arguments within parentheses
- class 'function'
- class 'method'
- dot expression
- <name>

The <expression> can be any valid Python expression.
The <name> must be a simple name.

To evaluate a dot expression:
- Evaluate the <expression> to the left of the dot, which yields the object of the dot expression.
- Evaluate the <name> on the right, which yields the value of attribute <name>.

Assignment statements with a dot expression on their left-hand side affect the object for that dot expression:
- If the object is an instance, then assignment sets an instance attribute.
- If the object is a class, then assignment sets a class attribute.