

Programming Paradigms

Programming Paradigms

➤ **Imperative (Non-Declarative) Programming**

- Fortran, C, Cobol, Pascal
- Object-Oriented Programming
 - C++, Java, C#

➤ **Declarative Programming**

- Functional Programming
 - ML, Lisp, Haskell, Scheme, F#
- Logic Programming
 - Prolog (Sicstus, SWI, GNU, YAP, Ciao)

Imperative Programming

- How to Solve, rather than what to solve
- Requires the programmer to specify an algorithm to be run
- Sequence of statements
- Makes the algorithm explicit and leaves the goal implicit

For example ...

Imperative Programming (Contd.)

For example ...

```
int Function (int n) {  
    int t = 1;  
    while (n > 0) {  
        t = t * n;  
        n = n - 1;  
    }  
    return t;  
}
```



Computes the factorial of n

Declarative Programming

- What to Solve, rather than how to solve
- Requires the programmer to specify just the problem, the language compiler figures the algorithm
- Sequence of definitions (functions or predicates)
- Makes the goal explicit and leaves the algorithm implicit

For example ...

Declarative Programming (Contd.)

For example ...

$\text{fac}(0) = 1$
 $\text{fac}(n) = n * \text{fac}(n-1)$

Computes the factorial of n



SEND + MORE = MONEY Puzzle

- Each letter represents a unique digit from 0 to 9.
- Two letters cannot represent the same digit.
- What digit each letter represents to satisfy the simple equation below?

Solution:

	S	E	N	D
+	M	O	R	E
<hr/>				
M	O	N	E	Y

	9	5	6	7
+	1	0	8	5
<hr/>				
1	0	6	5	2

SEND + MORE = MONEY

A Declarative CLP Program:

solve(Digits) :-

 Digits = [S,E,N,D,M,O,R,Y],

 Digits :: [0..9],

 alldifferent(Digits),

 1000*S + 100*E + 10*N + D

 + 1000*M + 100*O + 10*R + E

 #= 10000*M + 1000*O + 100*N + 10*E + Y,

 labeling(Digits).

? – solve(X)

 X = [9,5,6,7,1,0,8,2]

Declarative Vs Imperative Programming

Algorithm = Logic + Control

- Imperative programming needs both logic & control whereas Declarative programming needs just logic, it figures control on its own.
- Declarative programming is a higher level programming paradigm than Imperative.

Then why Declarative Programming is not so popular?

- Declarative programming is not as efficient as Imperative programming as it needs to figure out the control part of the algorithm on its own.
- Imperative programming is more closer to the popular (Von Neumann) architecture of a computer, whereas Declarative Programming is independent of the architecture, so not as optimized.

Applications of Declarative Programming

- Artificial Intelligence, Machine Learning
- Knowledge Representation, Semantic Web
- Deductive Databases (DataLog)
- Modeling and Simulation
- Verification and Validation
- Game Development
- Resource Allocation & Scheduling
- Many more ...