# Understanding Programming Paradigms

[Janeve George](http://www.janeve.me/author/janeve-george) Posted On March 20, 2013

## [Full Slides](https://www.slideshare.net/janevegeorge/programming-paradigms?ref=http://www.janeve.me/software-programming/understanding-programming-paradigms)

subtleties

## Preface

My colleague [Nilesh Mevada](http://in.linkedin.com/pub/nilesh-mevada/13/30a/801) and I conducted a session on Programming Paradigms at [Mukt.in](http://mukt.in/) 2008 (in Osmania University Hyderabad). After a great response we talked on the same at [Techfest](http://www.techfest.org/) 2009 ([IIT Mumbai](http://www.iitb.ac.in/)) and [XP Days Indore 2010](http://www.janeve.me/blog/2010/04/xp-days-indore-2010/). We talked about the programming paradigms and why we need too many of them? Why won’t a single programming paradigm and programming language solve all problems?

It was also joyful to see the attendees trying their hands-on various other programming languages and paradigms and solving their problems more tactically after the session. One of the attendee shared his experience with us:

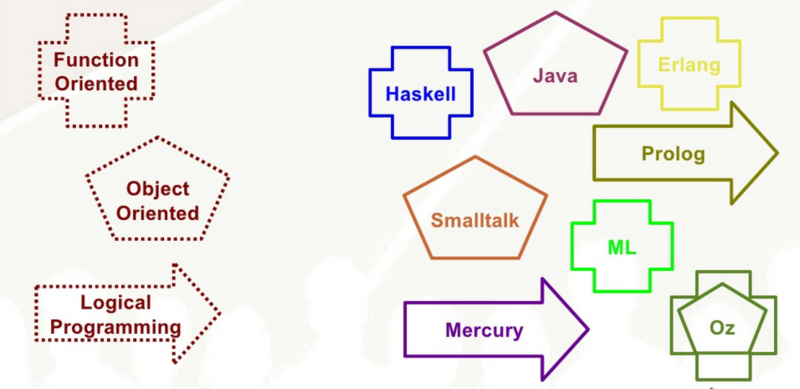
"I want to thank you and DirectI for conducting such a marvelous session and I am very much indebted for that. I fell in love with LISP programming language and found that it was very much ideal for signal processing algos. The approach you have suggested was great and the book you have suggested was really really great." - Chinni Krishna, Mukt.in '08 Session Attendee

|  |  |
| --- | --- |
| 1  2  3  4  5 | "I want to thank you and DirectI for conducting such a marvelous session and I am very much indebted for that.  I fell in love with LISP programming language and found that it was very much ideal for signal processing algos.  The approach you have suggested was great and the book you have suggested was really really great."  - Chinni Krishna, Mukt.in '08 Session Attendee |

We hope that this article will be interesting to you as well.

## Programming Paradigm – Definition

A Programming Paradigm is simply a **fundamental style** of computer programming. It is like the blue print of a construction site. A programming paradigm provides some very basic concepts and patterns which various designers of programming languages adopt and implement.



Programming Paradigms Patterns

In the above illustration we have few programming paradigms on the left which are patterns. The programming languages on the right uses one or more of these patterns to implement their features and functionality.

## Why should we bother about programming paradigms?

### Schools of Bridge Building Scenario

Let’s consider a scenario where there are two schools **School A** and **School B** that teaches their students to build bridge. **School A** teaches their students to build a bridge using wood, while **School B** teaches their students to build a bridge with iron rods. Now that students from both the schools have graduated and ready to build bridges.

Now the mayor of the town decides to build a new bridge and lay down few conditions.

The bridge should be

* Earthquake resistant at x level
* Should not erode
* Low cost
* etc..

Now the students from School A can only think of a solution with wood while students from School B can only think of a solution with iron rods. And these cannot satisfy all the requirements given by the mayor and both fails to come up with an optimal solution.

The problem is not because that they are not capable of building the bridge, but they had spent too much time and focus on the resources used instead of the concept of building a bridge. If they focus a bit more on the concepts and awareness of various other techniques to build a bridge, they would have been capable of coming up with an optimal solution that might include wood, iron and also other substance like concrete.

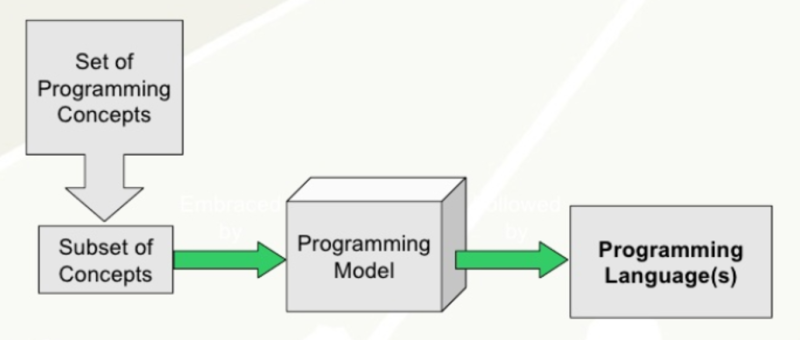
### The Programmers Analogy

Just like the schools of bridge building example, programmers are also equipped with knowledge, tools and techniques in implementing solutions using only the programming paradigms they know (usually only one). A programmer who has implemented projects in OOPs can only think in OOPs. Similarly a person who has implemented projects only in Procedure Oriented Language, can only think of procedures.

By focusing more on the programming concepts and various programming paradigms, it helps in understanding multiple programming languages and how they are different from each other. You will be able to bring innovations to the programming technologies and software architectures.

## Inception of Programming Paradigms

A programming language is evolved through a series of events as depicted in the image below.



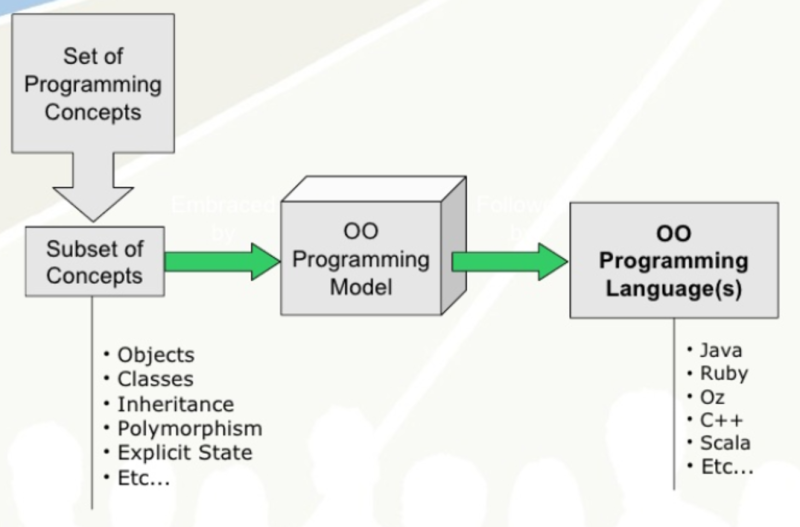
Programming Language Evolution

The first one being the inception of a programming concept. It might be brilliant idea someone came up with or probably a well planned and executed feature in a programming language that is already available. Few examples: Objects, Inheritance, Aspects, Functions, Lazy Evaluation, Variable Scope, Multithreading and concurrency, Dynamic Programming etc…

A subset of these concepts are chosen by the programming language designers who define a **Programming Model**. They then get busy in implementing these features which results in your favorite **Programming Language(s)**. When a particular programming model matures and more and more language implementers start adopting them, it becomes a **Programming Paradigm**.

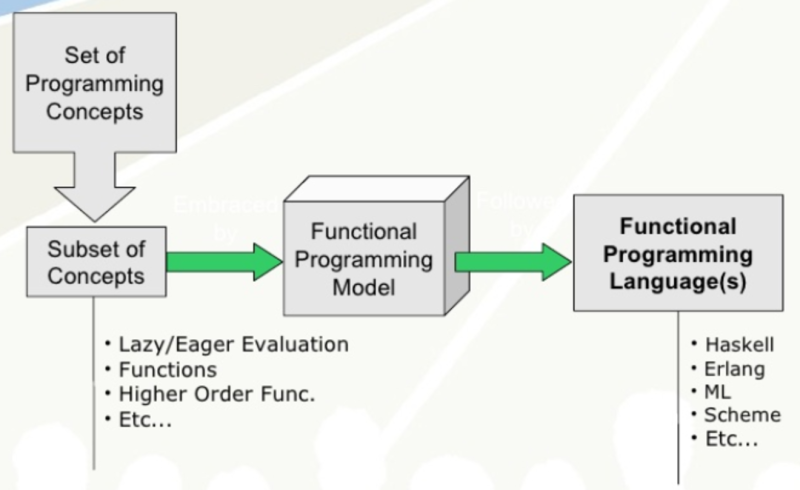
Few examples:

The inception of the Object Oriented Programming Model



Object Oriented Programming Model

The inception of Functional Programming Model



Function Oriented Programming Model

There are few cool programming languages out there that provide a mix of all these popular programming models. For example, Scala supports Functional and Object Oriented concepts, similarly Oz has huge feature list that comes from Procedural, Functional and Object Oriented.

## Popular Programming Concepts

Few very popular and key concepts that you might be already using while you program are:

* Eager Evaluation
* Lazy Evaluation
* Procedures
* Functions
* External State
* Internal State
* Static Typing
* Dynamic Typing
* Concurrency

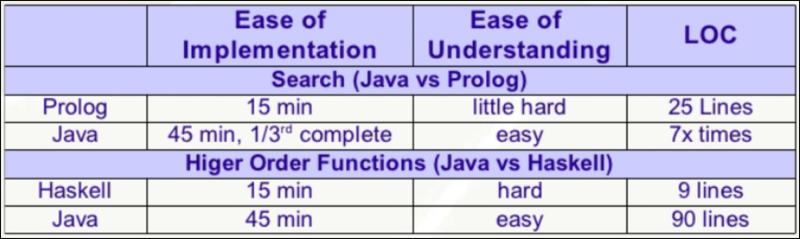
## A real life experiment

We took a problem statement in which given some family details, we had to find if any two people are related. We found that a logical programming language such as prolog was the best choice for such scenarios.

% % Facts % male(hrithik). male(shahrukh). male(salman). male(abhishek). male(akshay). male(aamir). female(diya). female(aishwarya). female(katrina). female(malaika). parent(hrithik,shahrukh). parent(hrithik,salman). parent(hrithik,diya). parent(shahrukh,abhishek). parent(shahrukh,akshay). parent(salman,aishwarya). parent(salman,katrina). parent(salman,aamir). parent(diya,malaika). % % Rules % father(X,Y) :- parent(X,Y), male(X). mother(X,Y) :- parent(X,Y), female(X). grandparent(X,Y) :- parent(X,Z), parent(Z,Y). paternalgrandfather(X,Y) :- father(X,Z), father(Z,Y). sibling(X,Y) :- parent(Z,X), parent(Z,Y). brothers(X,Y) :- sibling(X,Y),male(X),male(Y), \+ (X=Y). % % Queries % ?- mother(diya,malaika) % Outupt: yes ?- parent(salman,aamir) % Outupt: yes ?- parent(salman,malaika) % Outupt: no

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 | %  % Facts  %    male(hrithik).  male(shahrukh).  male(salman).  male(abhishek).  male(akshay).  male(aamir).  female(diya).  female(aishwarya).  female(katrina).  female(malaika).  parent(hrithik,shahrukh).  parent(hrithik,salman).  parent(hrithik,diya).  parent(shahrukh,abhishek).  parent(shahrukh,akshay).  parent(salman,aishwarya).  parent(salman,katrina).  parent(salman,aamir).  parent(diya,malaika).    %  % Rules  %    father(X,Y) :- parent(X,Y), male(X).  mother(X,Y) :- parent(X,Y), female(X).  grandparent(X,Y) :- parent(X,Z), parent(Z,Y).  paternalgrandfather(X,Y) :- father(X,Z), father(Z,Y).  sibling(X,Y) :- parent(Z,X), parent(Z,Y).  brothers(X,Y) :- sibling(X,Y),male(X),male(Y), \+ (X=Y).  %  % Queries  %  ?- mother(diya,malaika)  % Outupt: yes  ?- parent(salman,aamir)  % Outupt: yes  ?- parent(salman,malaika)  % Outupt: no |

It was really quick to come up with a solution in Prolog since it is logical programming language. Being Java developers, we tried to implement the same logic in Java and this is the stats that we came up with:



Please note that these stats are from the perspective of a Java professional. We would suggest you to focus a little more on the programming concepts an then how these are implemented in a language. If you have similar experience do share them with us by commenting below.