|  |
| --- |
| University of Advancing Technology |
| Programming Paradigms |
| Emerging Programming Languages |
|  |
| **Caleb J Thompson** |
| **9/18/2010** |

|  |
| --- |
|  |

Table of Contents

[Procedure Oriented 3](#_Toc272684461)

[Essential Elements. 3](#_Toc272684462)

[Differences from other paradigms. 3](#_Toc272684463)

[Representative languages. 3](#_Toc272684464)

[Resources. 3](#_Toc272684465)

[Object Oriented 4](#_Toc272684466)

[Essential Elements. 4](#_Toc272684467)

[Differences from other paradigms. 4](#_Toc272684468)

[Representative languages. 4](#_Toc272684469)

[Resources. 4](#_Toc272684470)

[Functional 4](#_Toc272684471)

[Essential Elements. 5](#_Toc272684472)

[Differences from other paradigms. 5](#_Toc272684473)

[Representative languages. 5](#_Toc272684474)

[Resources. 5](#_Toc272684475)

[Logic Programming 5](#_Toc272684476)

[Essential Elements. 5](#_Toc272684477)

[Differences from other paradigms. 6](#_Toc272684478)

[Representative languages. 6](#_Toc272684479)

[Resources. 6](#_Toc272684480)

## Procedure Oriented

The most common high-level languages today are procedure-oriented languages. In these languages, one or more related blocks of statements that perform some complete function are grouped together into a program module, or procedure, and given a name such as “procedure A” (Microsoft Encarta, 2009).

### Essential Elements.

The procedural programming paradigm derives from structured programming, which is based upon the concept of the function call. Functions are simply a series of computational steps to be carried out, and a given function can be called from anywhere in the program’s code, including inside other functions or itself. This gives us the first element, Modularity. Modularity allows reuse of code and specifies inputs and outputs for specific functions. Scoping, the second element, is a technique which acts similarly to folders on an operating system, allowing further modularity as well as reuse of function names for greater readability (i.e. List.Sort(), Dictionary.Sort()).

### Differences from other paradigms.

The procedural paradigm is different from the functional paradigm in that it is slightly more verbose and doesn’t focus so much on recursion. Procedural languages support assignment to variables while functional focus on static members. Procedural languages are different from object-oriented models in that they do not attempt to model objects in the problem as classes in favor of creating an ordered list of instructions.

### Representative languages.

* C
* C++
* Fortran
* Pascal
* Basic

### Resources.

Procedural programming. (2010, September 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 22:01, September 19, 2010, from [http://en.wikipedia.org/w/index.php?title=Procedural\_programming](http://en.wikipedia.org/w/index.php?title=Procedural_programming&oldid=382285897)

"Programming Language," Microsoft® Encarta® Online Encyclopedia 2009

Functional vs. Procedural Programming Language. (2010, September 18). Retrieved 15:49, September 19, 2010, from <http://amath.colorado.edu/computing/mmm/funcproc.html>

## Object Oriented

Object-oriented programming (OOP) is a programming paradigm that uses "objects" – data structures consisting of data fields and methods together with their interactions – to design applications and computer programs. Programming techniques may include features such as data abstraction, encapsulation, modularity, polymorphism, and inheritance (Wikipedia).

### Essential Elements.

As indicated by the name, the object oriented paradigm is characterized by the use of objects as a data structure. Objects are used to model what in a simple sentence’s description of a problem would be a noun. Objects consist of data fields and methods.

### Differences from other paradigms.

The object-oriented approach is knows to introduce a significant performance penalty compared to classical procedural programming, according the Chatzigeorgiou. Other paradigms do not include OOP’s fundamental idea, which is the object.

### Representative languages.

* C#
* Objective C
* Java

### Resources.

Chatzigeorgiou, Alexander (2003). ["Performance and power evaluation of C++ object-oriented programming in embedded processors"](http://www.sciencedirect.com/science/article/B6V0B-47MJ59W-1/2/accc87febd5a5db8a73f924ea23ada47). *Information and Software Technology* **45** (4): 195–201.

Functional vs. Procedural Programming Language. (2010, September 18). Retrieved 15:49, September 19, 2010, from <http://amath.colorado.edu/computing/mmm/funcproc.html>

Object-oriented programming. (2010, September 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 01:04, September 20, 2010, from [http://en.wikipedia.org/w/index.php?title=Object-oriented\_programming](http://en.wikipedia.org/w/index.php?title=Object-oriented_programming&oldid=385199389)

## Functional

Functional programming languages emphasize rules and pattern-matching. While they appear non-intuitive to those who have only experienced procedural languages, they provide succinct and natural programming structures for those who gain some experience. Functional programming is particularly useful for mathematical work, where the notion of ``function'' is already a well established concept (Colorado University).

### Essential Elements.

Functional programming focuses on immutable data members and avoids state. It treats computation as the evaluation of mathematical functions. Functions are also treated as values the same way variables are, the affect of which I still do not understand. Functional programming is declarative.

### Differences from other paradigms.

Functional programming often appears backward relative to other paradigms because it involves doing either the beginning or end of a task then recursing on the list of arguments to complete. Data is immutable and unlike imperative paradigms there is no state.

### Representative languages.

* Lisp (Scheme)
* F#
* Mathematica

### Resources.

Functional programming. (2010, September 10). In *Wikipedia, The Free Encyclopedia*. Retrieved 01:15, September 20, 2010, from [http://en.wikipedia.org/w/index.php?title=Functional\_programming](http://en.wikipedia.org/w/index.php?title=Functional_programming&oldid=383995245)

Functional Programming For The Rest of Us. (2006, June 19). In *defmacro*. Retrieved 18:18, September 19, 2010 from <http://www.defmacro.org/ramblings/fp.html>

## Logic Programming

### Essential Elements.

Logic programming is inherently declarative. This means that instead of explaining in code *how* something is to be accomplished the programmer explains *what* needs to be done. Logic programming uses mathematical logic for computer programming. It is used to apply declarative statements in the form of implications.

For example, it treats the implication:

If you press the alarm signal button,

then you alert the driver of the train of a possible emergency

as the procedure:

To alert the driver of the train of a possible emergency,

press the alarm signal button.

### Differences from other paradigms.

The programmer of a logic paradigm language is responsible for the truth of programs. The programmer is also relied upon to ensure that inferences are generated efficiently.

### Representative languages.

* Prolog
* R++
* Datalog

### Resources.

Chitta Baral and Michael Gelfond. [Logic programming and knowledge representation](http://www.cs.ttu.edu/~mgelfond/papers/survey.pdf). Journal of Logic Programming. 1994, Vol. 19, 73-148.

Logic programming. (2010, June 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 23:37, September 19, 2010, from [http://en.wikipedia.org/w/index.php?title=Logic\_programming](http://en.wikipedia.org/w/index.php?title=Logic_programming&oldid=371003841)

Robert Kowalski. [The Early Years of Logic Programming](http://www.doc.ic.ac.uk/%7Erak/papers/the%20early%20years.pdf) CACM. January 1988.