

**SAMPLE PROGRAMMING LANGUAGE DIMENSIONS**

	<b>Pascal</b>	<b>C</b>	<b>Java</b>	<b>Logo</b>	<b>Scheme</b>	<b>OCaml</b>	<b>Haskell</b>
<b>Paradigm</b>	Imperative	Imperative	Object-Oriented	Imperative/ Functional	Mostly functional	Mostly functional	Purely functional
<b>Grammar</b>	Context free	Context free	Context free	Context sensitive	Context free	Context free	Mostly context free, but whitespace matters
<b>Functional Values</b>	Can be passed as arguments	yes	no	no	yes	yes	yes
<b>Block Structured</b>	yes	no	No (but inner classes help)	no	yes	yes	yes
<b>Scoping</b>	Static	Static	Static	Dynamic	Static	Static	Static
<b>Parameter Passing</b>	Call-by-value and Call-by-reference	Call-by-value (but addresses are values)	Call-by-value	Call-by-value	Call-by-value	Call-by-value	Call-by-need
<b>Typing</b>	Strong static (but has holes)	Weak static	Strong static	Strong dynamic	Strong dynamic	Strong static	Strong static
<b>Parametric Polymorphism</b>	no	For datatypes of same size	Class-based	Tagged data	Tagged data	Hindley/Milner	Hindley/Milner
<b>Properly tail recursive</b>	no	no	no	no	yes	some implementations	no
<b>Dynamic Array Bounds Checking</b>	no	no	yes	N/A	yes	yes	yes
<b>Explicit pointers</b>	yes	yes	no	no	no	no	no
<b>Heap Storage Deallocation</b>	Manual	Manual	Automatic (Garbage collected)	Automatic (Garbage collected)	Automatic (Garbage collected)	Automatic (Garbage collected)	Automatic (Garbage collected)

