

# Choosing a Program Language

## Topic Paper #18

Alex Laird  
CS-3210-01

4/13/09

Survey of Programming Languages

Spring 2009

Computer Science, (319) 360-8771

[alexdlaird@cedarville.edu](mailto:alexdlaird@cedarville.edu)

Grading Rubric		Max	Earn
	On Time/Format	1	
	Correct	5	
	Clear	2	
	Concise	2	
	Total	10 pts	

Peer Reviewer
------------------

### ABSTRACT

This paper discusses the procedure for choosing a programming paradigm and language for use in developing a computer program.

### Keywords

Choosing, Programming, Paradigm, Language

## 1. INTRODUCTION

In this day and age, there are hundreds upon hundreds of programming languages and derivations thereof to choose from when attempting to complete a software implementation task. Depending on the scale and type of problem being implemented, different languages are more appropriate for different difficulty levels. Where one language will implement a task in a fairly straight forward manner, another language will be excruciatingly difficult, or not even possible, to implement the same task with. It is important to take a few key factors of programming into consideration when choosing an appropriate language.

## 2. TYPING

There are two standard type systems in programming: static and dynamic.

A programming language is said to be statically typed if type-checking is performed at compile time. The advantage of static typing is that many errors can be caught early on in the programming process instead of having to debug them out later. However, there are occasions where static typing can be limiting and less versatile than dynamic typing. Some of the most popular statically typed languages are C, C++, C#, Java, Fortran, and Perl.

A programming language is said to be dynamically typed if all (or most) of the type-checking is done at run-time. The compiler still ensures syntactical correctness, but types are set when the variables are used in runtime, and types can be changed for different portions of the program, even when using the same variable. Dynamic languages are known for asserting buggy behavior on runtime due to typing errors if the programmer was

sloppy; these are far more difficult to catch. Some of the most popular dynamically typed languages are Smalltalk, Objective-C, Python, Ruby, Smalltalk, and Perl. Perl is both static and dynamic).

## 3. PARADIGM

There are four major programming paradigms to choose from when choosing a programming language. The advantages and disadvantages of each paradigm have been discussed more extensively in previous topic papers, but different paradigms make different tasks simpler or more complicated. A task that can be done quickly with classes in Java may be made very difficult with recursion and functionality in Scheme. Some languages incorporate numerous paradigms, and these are some of the most versatile languages to use.

## 4. READABILITY, DOCUMENTATION, AND COMPATABILITY

Many languages are far more accessible when it comes to maintenance and compatibility. When working with on a collaborative group, it's important to use code that is readable, well documented itself, and easily able to be documents. Java is a perfect example of documentation as it is heavily documented and Javadocs is a wonderful feature that allows multiple programmers to easily collaborate and understand what their various functions and classes do.

Compatibility, or integration, is important in a large project if multiple languages are going to be incorporated into the same project, communicating with each other. Different languages communicate with other languages better and more freely.

## 5. CONCLUSIONS

The advantages and disadvantages of a programming languages characteristics should be none and thought through before deciding to use it on a project. Even on seemingly small projects, much time can be saved if one makes a wise choice of the language to use before writing the program.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Alex Laird, Cedarville University, Cedarville, Ohio, 45314  
Copyright 2009 Alex Laird