Problem Set 0

Warren Kim

April 5, 2023

Contents

1 Overview

1.1 What is a Programming Language?

A programming language is a structured system of communication designed to express computations in an abstract manner.

1.2 Why Different Languages?

Different languages are built for different use cases. Below are popular languages that were built for their respective use cases:

Javascript is the most popular language for anything related to web development. There are many frameworks for vanilla Javascript (e.g. React) as well as derivative languages (e.g. Typescript).

C/++ is a popular language for programs that require high performance (e.g. Linux).

C# is most commonly used for programs that are in Microsoft's .NET ecosystem.

Python is a popular language used in the field of artificial intelligence.

ba/z/sh is a scripting language for UNIX-based operating systems.

R is a popular language among statiticians (not sure why).

Lisp is a functional language used in the field of artificial intelligence and was used to write Emacs.

SQL and its variants are a set of querying languages used to communicate with databases.

2 Language Paradigms

There are four main language paradigms:

Imperative

Object-Oriented

Functional

Logic

2.1 Imperative Paradigm

Imperative programming uses a set of statements (e.g. control structures, mutable variables) that directly change the state of the program. More specifically, these statements are commands that control how the program behaves. Common examples of imperative languages include FORTRAN and C.

2.2 Object-Oriented Paradigm

The object-oriented programming is a type of imperative programming, and contains support for structured objects and classes that "talk" to each other via methods (e.g. `d` is a `Dog` object with the class method `bark()`, where `d.bark()` will invoke the `bark` function for the object `d`). Common examples of object-oriented languages include Java and C++.

2.3 Functional Paradigm

Functional programming is a type of declarative programming. They use expressions, functions, constants, and recursion to change the state of the program. There is no iteration or mutable variables. Common examples of functional languages include Haskell and Lua.

2.4 Logic Paradigm

Logic programming the most abstract and is a type of declarative programming. A set of facts and rules are defined within the scope of the program. Common examples of logic languages are Prolog and ASP.

3 Language Choices

There are many things to consider when building a programming language. Some of these include:

Static/Dynamic type checking

Passing parameters by value/reference/pointer/object reference

Scoping semantics

Manual/Automatic memory management

Implicit/Explicit variable declaration

Manual/Automatic bounds checking

Generally, a programming language can be broken down into its syntax and semantics.