Contents

1 Lecture 1: 8/31/21

A language is Turing Complete if it can compute any function computable by the 1936 Turing Machine. Theoretically, all general purpose languages are equivalent in the programs that they can compute. In practice however, some are better than others for specific tasks. There are also non-Turing Complete languages, like G-Code, JSON, HTML/CSS, which do not have variables, ifs, and loops, etc.

There are two main paradigms, imperative and declarative. Imperative languages (Python, C, Java, C Sharp, etc.) use commands, while declarative languages (Prolog, etc.) use a set of rules, and use statements rather than commands. Early languages were imperative, such as FORTRAN, Pascal, and C. These map closely to machine behavior (Von Neumann architecture). They are statements and procedures that modify variable data.

Functional programs started with LISP, and programs are collections of functions, typically recursive. In pure functional languages, there are no variable updates, everything is immutable. Functions can be higher order, and can be passed in as arguments to other functions.

Object oriented programs started with Smalltalk, and objects store the state and functions, and computation is done via passing messages to objects. Programs are collections of objects.

Syntax is the grammar for a language. LISP is a prefix language, operations are placed before arguments. Java on the other hand has infix notation. The differences can often be pretty trivial.

Semantics is the meaning for the language, and Pragmatics is the conventions for the language, such as good habits and patterns, as well as the natural approach to using the langauge as it was designed.

Ther are also compiled languages and interpreted languages. Interpreted languages run line by line, while compiled languages are converted to machine code completely, and then linked to the input to produce the output.

2 Lecture 2: 9/2/21

Ruby is made as a scripting language, made for doing a lot of things quickly, but not things that require a lot of power. Scripting languages are a broad genre, and they have had increasing use due to higher layer abstractions.

Ruby is imperative and OO, and has many of the same features. Something that is new is the lack of primitives, everything is a primitive. It also has lightweight syntax, less strict than Java. It also has dynamic implicit typing, along with new data types. It also has regular expressions and code blocks.

Ruby was designed to make programmers happy, and was created in 1993.

Ruby has implicit typing, where the type is inferred and not given by the programmer, and also dynamic typing, where a variable can change types in a program. This is the opposite of Java, which has explicit and static typing.