Introduction to Programming Languages

CS152

Chris Pollett

Aug. 25, 2008.

Outline

- What is a Programming Language?

What is a Programming Language?

- If you look back on early computers like ENIAC, reprogramming meant rewiring the machine.
- So a programming language gives us a notation for us to tell a computer what to do without having to re-wire.
- For this to work, the programmer needs to understand what is being told to the computer and the the computer needs to be able to understand what it is supposed to do.
- Hence, a **programming language** is a notation system for describing *computation* in both a *machine readable* and *human readable* format.

What is a computation?

- This was a question that was studied in the 1930 and 1940s.
- Turing, Church and others defined different models of computation based on things such as physical implementability, operations from mathematics, linguistic manipulation, etc.
- Each of these definitions turned out to be equivalent to each other.
- So for our purposes a computation is what can be done a computer with as much memory as we need and with as time as we need.
- A language which can allow us to express any computation will be called **Turing complete**.

Machine Readable

- Since computers understand machine code instructions like:

      cmp eax, ebx
      jne do_something

  or the bits that correspond to them.

- We need to have some way to translate from a programming language to machine code.

- So that this always works we need a step by step procedure (an algorithm) which will work for all programs written in our language.

- For languages which are specified using certain kinds of context-free grammars such algorithms exist.

Human Readable

- This is a harder to define thing than machine readable.
- Basically, though it amounts to the programming language to provide **abstractions** which are close to how a human would think about solving a problem.
- Next day, we will talk about the basic kinds of abstractions people have considered.