Programming Languages

- What is a programming language?
Programming Languages

• What is a programming language?
  – Abstraction of virtual machine

```java
int sum(int[] x) {
    int sum = 0;
    n = 0;
    while (n < x.length) {
        sum += x[n];
    }
    return sum;
}
```

00101010101010
10101011111010
11101010101110
00101010101010
...

Programming Languages

• What is a programming language?
  – Donald Knuth:
    » Programming is the art of telling another human being what one
      wants the computer to do

```java
int sum(int[] x) {
    int sum = 0;
    n = 0;
    while (n < x.length) {
        sum += x[n];
    }
    return sum;
}
```

00101010101010
10101011111010
11101010101110
00101010101010
...

The Number of Programming Languages

- How many programming languages do you know?
  - This is a sample list…
    » http://dmoz.org/Computers/Programming/Languages/

- Why is the number of programming languages so large?
  - Evolution
  - Special Purpose
  - Personal Preference
Evolution: Genealogy

The Number of Programming Languages

- How many programming languages do you know?
  - This is a sample list...
    - http://dmoz.org/Computers/Programming/Languages/

- Why is the number of programming languages so large?
  - Evolution
  - Special Purpose
  - Personal Preference
The Number of Programming Languages

• How many programming languages do you know?
  – This is a sample list…
    » http://dmoz.org/Computers/Programming/Languages/

• Why is the number of programming languages so large?
  – Evolution
  – Special Purpose
  – Personal Preference

• A programming language is a way of thinking
  – Different people think in a different way
Quicksort in C

```c
qsort( a, lo, hi ) int a[], hi, lo;
{
    int h, l, p, t;
    if (lo < hi) {
        l = lo;
        h = hi;
        p = a[hi];
        do {
            while ((l < h) && (a[l] <= p))
                l = l+1;
            while ((h > l) && (a[h] >= p))
                h = h-1;
            if (l < h) {
                t = a[l];
                a[l] = a[h];
                a[h] = t;
            }
        } while (l < h);
        t = a[l];
        a[l] = a[hi];
        a[hi] = t;
        qsort( a, lo, l-1);
        qsort( a, l+1, hi);
    }
}
```
Quicksort in Haskell

```haskell
qsort [] = []
qsort (x:xs) = qsort lt_x ++ [x] ++ qsort ge_x
  where
    lt_x = [y | y <- xs, y < x]
    ge_x = [y | y <- xs, y >= x]
```

Successful Programming Languages

- Are all languages equally successful?
  - No!
- What makes a language successful?
  - Expressive power
  - Ease of use for the novice
  - Ease of implementation
  - Excellent compilers
  - Economics, patronage, and inertia
Why study programming languages?

• Use the most *appropriate* programming language for your task
  – E.g. Java is great for writing applications
  – E.g. C is great for systems programming

• Make it easier to learn new languages
  – Evolution => Similarities

• Make good better use of language features
  – Obscure features
  – Cost of features
  – Simulate useful features

Classification of Programming Languages

• **Imperative languages**
  – What the computer is to do

• **Von Neumann languages**
  » E.g. Fortran, Basic, C

• **Object-oriented languages**
  » E.g. C++, Java
Classification of Programming Languages

- **Declarative languages**
  - How the computer should do it
- **Functional languages**
  - E.g. Lisp, ML, and Haskell
- **Dataflow languages**
  - E.g. Id and Val
- **Logic or constraint-based languages**
  - E.g. Prolog

Summary

- **Programming languages:**
  - Set of abstractions => virtual machine
  - A way of thinking
- **COMP 144:**
  - Examine the fundamental principles of contemporary programming languages
    » Design
    » Implementation
  - Program in four completely different programming languages
    » Practical experience