What are Programming Paradigms?

- The concepts and abstractions used to represent the elements of a program (e.g., objects, functions, variables, constraints, etc.) and the steps that compose a computation (assignment, evaluation, continuations, data flows, etc.).
- Or, a way to classify the style of programming.

Most Languages Are Hybrids!

- This makes it hard to teach to students, because most languages have facets of several paradigms:
  - Called “Multi-paradigm” languages
  - Scratch too!
  - It’s like giving someone a juice drink (with many fruit in it) and asking to taste just one fruit!

Functional Programming (review)

- Computation is the evaluation of functions
  - Plugging pipes together
  - Each pipe, or function, has exactly 1 output
  - Functions can be input!
- Features
  - No state
  - E.g., variable assignments
  - No mutation
  - E.g., changing variable values
  - No side effects
- Examples (the not pure)
  - Scheme, Scratch BYOB

Impersonal Programming

- “Sequential” Programming
- Computation a series of steps
  - Assignment allowed
  - Setting variables
  - Mutation allowed
  - Changing variables
- Like following a recipe. E.g.,
  - Procedure (fx)
    - ans = x
    - ans = (x+3) * ans
    - return ans
- Examples: (the not pure)
  - Pascal, C

Object-Oriented Programming (OOP)

- Objects as data structures
  - With methods you ask of them
  - These are the behaviors
  - With local state, to remember
  - These are the attributes
- Classes & Instances
  - Instance an example of class
  - E.g., Fluffy is instance of Dog
- Inheritance saves code
  - Hierarchical classes
  - E.g., pianist special case of musician, a special case of performer
- Examples (the not pure)
  - Java, C++
OOP Example: SketchPad

- Dr. Ivan Sutherland
  - "Father of Computer Graphics"
  - 1988 Turing Award
  - Nobel prize for CS
  - Wrote Sketchpad for his foundational 1963 thesis

- The most impressive software ever written
- First...
  - Object-oriented system
  - Graphical user interface
  - non-procedural language

Spent the past few years doing research @ Berkeley in EECs dept!

OOP in BYOB

Declarative Programming

- Express what computation desired without specifying how it carries it out
  - Often a series of assertions and queries
  - Feels like magic
- Sub-categories
  - Logic
  - Constraint
    - We saw in Sketchpad!

- Example: Prolog

Anders Heijlsberg
- "The Future of C# @ PDC2008 channel9.msdn.com/pdc2008/716/

Declarative Programming Example

- Five schoolgirls sat for an examination. Their parents – so they thought – showed an undue degree of interest in the result. They therefore agreed that, in writing home about the examination, each girl should make one true statement and one untrue one. The following are the relevant passages from their letters:

  Kitty: I came out first
  Joan: I was 2nd
  Ethel: I was 3rd
  Mary: I was 4th
  Betty: I was 1st

  Kitty was last
  Joan was on top
  Ethel was not
  Betty was only 4th

Turing Completeness

- A Turing Machine has an infinite tape of 1s and 0s and instructions that say whether to move the tape left, right, read, or write it
  - Can simulate any computer algorithm
- A Universal Turing Machine is one that can simulate a Turing machine on any input
- A language is considered Turing Complete if it can simulate a Universal Turing Machine

- A way to decide that one programming language or paradigm is just as powerful as another

Ways to Remember the Paradigms

- Functional
  - Evaluate an expression and use the resulting value for something
- Object-oriented
  - Send messages between objects to simulate the temporal evolution of a set of real world phenomena

- Imperative
  - First do this and next do that
- Declarative
  - Answer a question via search for a solution