What is in the miniJava language?

- classes including one special main class
  - fields, methods, inheritance
  - method overriding – sort of ...
- integer and boolean primitive types
- arrays of integers
- if, while, System.out.println, assignment
- array access, &&, new, method invocation
- Runtime checking
  - array bounds
  - nil object references – don’t really need to do this so long as “nil” is an invalid address
What is not in the miniJava language?

- interfaces
- char, long, float, double
- The Java library
  - import, package
- Dynamic method dispatch
- Nested and anonymous classes
- Overloaded methods
From “Functions” to “Java” (well, miniJava)

- What will be necessary to go from the Functions compiler to a miniJava compiler?
  - Scanning
  - Parsing
  - Scopes
  - Type checking
  - Runtime memory organization (stack frames)
  - Translation to IR code
Scanning

- Additional keywords
  - class, extends, if, else, while, this, void, ...

- Additional punctuation
  - “[”, “]”, “.”, ...
Parsing and ASTs

• Additional statement constructs
  – Classes, if, while, System.out.println

• Additional expressions
  – this, a.b(), a[i], new A(), new int[n]
Scopes

• Additional scope complexity
  – Classes contain fields
  – Inheritance
    • A subclass inherits all of the fields and methods of the superclass
  – The “.” operator “opens up” scopes
    • `a.m()` means look up m in the scope associated with the type of a
  – Classes and fields can be used before they are defined
    • Two passes over the AST are essential
Types

- Additional types
  - Array types (element type is int, index type is int)
- Class types
  - class B extends A {
  - A a; a = new B();
- What is the type of “this”?
Activation records and methods

- Nothing much changes
  - Every method takes an implicit (hidden) first argument
    - “this”
    - address of the object on which the method is invoked
    - used to find fields of the object
Translation to IR

- if
- while
- Array accesses
  - accessing memory
  - bounds checking
- Field accesses
  - accessing memory
- Method invocation
  - passing “this”