Where We Are

Credit
Most of the slides in this lecture come from www.nand2tetris.org
Current Focus

Book chapters and Course projects:
9, 10, 11, 12
7, 8
1, 3, 4, 5, 6

The VM Idea

Perspective:
Next few lectures describe the VM model used in the Hack-Jack platform, which implements a stack machine.

Other VM models (like Java's JVM/JRE and .NET's IL/CLR) are similar in spirit but differ in scope and details.

Several different ways to think about the notion of a virtual machine:

- **Abstract software engineering view**: the VM is an interesting abstraction that makes sense in its own right
- **Practical software engineering view**: the VM code layer enables "managed code" (e.g. enhanced security)
- **Pragmatic compiler writing view**: a VM architecture makes writing a compiler much easier (as we'll see later in the course)
- **Opportunistic empire builder view**: a VM architecture allows writing high-level code once and have it run on many target platforms with little or no modification.

Yet Another View

"Programmers are creators of universes for which they alone are responsible. Universes of virtually unlimited complexity can be created in the form of computer programs."

(Joseph Weizenbaum)

Our VM model + language are an example of one such universe
Also see Simulation Hypothesis.

The Big Picture

- JVM
  - CLR
  - VM
  - 7, 8
- Java
  - C#
  - Jack
  - 9
- Java compiler
  - C# compiler
  - Jack compiler
  - 10, 11
- JRE
  - .NET base class library
  - Mini OS
  - 12
(Basic chapters and Course projects)