Lecture 7: Recursive vs. Iterative Code

Prof. John Wenskovitch
09/11/2015
Last Time

- Design of three classes: Human, CPU, and Game
- Implementation of the Human class
- Implementation of the CPU and Game classes
- Implementation of the Player class, which we set as a superclass of the Human and CPU classes
- Constructors, Polymorphism, Abstract Classes, and Dynamic Dispatch
Iterative vs. Recursive Code

- **Iterative** code – Code that repeats a process until a desired result is reached.
- **Recursive** code – A technique by which a method makes one or more calls to itself during execution.
  - **Base case** – Fixed values of the function.
  - **Recursive case** – Defines the function in terms of itself.
Example 1: Factorial Calculation

• **Question:** What is “12!”?
  – 12! = 12*11*10*9*8*7*6*5*4*3*2*1
  – 12! = 479,001,600

• **Iterative calculation:** Put it in a for-loop.

• **Recursive calculation:** Use fact(n-1) to calculate fact(n).
Example 1: Fibonacci Calculation

• **Question:** What are the Fibonacci numbers?
  – 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233
  – Each number is the sum of the two numbers preceding it.

• **Iterative calculation:** Put it in a for-loop.

• **Recursive calculation:** Use fib(n-1) and fib(n-2) to calculate fib(n).
Any Questions?