Our Toolkit

• Graphics  
  – lines, shapes, images, text, color, ...

• Data of Various Types  
  – Numbers (with and without decimal places)  
  – Booleans (true, false)  
  – Color (two color models)  
  – Characters and Strings

• Variables  
  – Hold/name any type of data values

• Operators  
  – Mathematical (+, *, ++, %, ...)  
  – Relational (<, >, >=, ==, ...)  
  – Logical (&&, ||, !)

Our Toolkit (Continued)

• Functions  
  – Mathematical, Graphical, Utility, ...
  – Of our own design

• Expressions  
  – Combination of data, variables, operators, functions

• Conditionals  
  – if-statements

• Iterations  
  – while-loop  
  – for-loop

• Data Structures  
  – Arrays  
  – Functions that manipulate arrays

• Objects

Top-Down Design

• At first blush, solving a hard problem can seem daunting  
  – Create a clone of Adobe Photoshop  
  – Create a new web browser

• A common technique for solving complex problems is called Top-Down Design  
  – a.k.a. “Step-wise Refinement”  
  1. Define a sequence of steps to solve a given problem at the highest, most abstract level.  
  2. Recursively, list a sequence of sub-steps to solve each higher-level step  
  3. Repeat until the sub-problem is “easy enough” to solve directly

Top-Down Design - Advantages

• Promotes Organization  
  – Your code is naturally organized, and easy to understand  
  – Avoids the “spaghetti code” syndrome

• Simplifies the Problem  
  – The larger complex problem reduces to several smaller, more simple problems

• Promotes Reuse  
  – Several sub-problem solutions may be reusable by multiple parts of your program  
  – Some sub-problems have existing solutions implemented

• Enables Shared Development  
  – Multiple people can work on different parts of the problem at the same time

Top-Down Design - Example

Have Dinner
1. Cook Food
2. Set Table
3. Serve Food
4. Eat Food
5. Clean Up

Top-Down Design - Example

Have Dinner
1. Cook Food
   1. Boil Noodles
   2. Stir-fry Veggies
   3. Mix together
2. Set Table
3. Serve Food
4. Eat Food
5. Clean Up
Top-Down Design - Example

Have Dinner
1. Cook Food
   1. Boil Noodles
      1. Boil water
      2. Pour in dry noodles
      3. Let cook
      4. Strain noodles
   2. Stir-fry Veggies
   3. Mix
2. Set Table
3. Serve Food
4. Eat Food
5. Clean Up

Pop
• A game that measures your balloon-popping skill.
• How it should work...
  – As game runs, randomly placed balloons inflate
  – When the player pops (clicks on) a balloon, 1 point is earned
  – Points are added up throughout the game duration
  – If one click is over top multiple balloons, all balloons pop and
    multiple points are earned
  – The game runs for 30 seconds, and then ends