CSCI 1103: Simple Objects

Chris Kauffman

Last Updated:
Logistics

Reading from Eck
Ch 5 on Objects/Classes

Goals
- Finish methods, call stack
- Simple Objects

Lab07: Method Practice

Project 3
- Due Wednesday
- Questions?
Java Classes

- `.java` files define classes
- Classes serve 2 purposes
  1. Group related code together (binary/decimal conversions, Collatz methods, etc.)
  2. Define new reference types
- Mixture of purposes can cause confusion but is somewhat intentional
- Most of the time Methods and Data Types are tied together
Simple Objects

▶ Java classes define new reference types by specifying **fields**

// Represent data associated with an omelet.
public class Omelet{
    public int eggs; // How many eggs in the omelet
    public int cheese; // How many ounces of cheese
    public double cookedFor; // How long the omelet has cooked
    public String extras; // Extra ingredients added
}

▶ Fields are data data associated with the an **object** which as instance of the class. Every instance has its own fields

▶ **Instances** of objects are created with `new` and a constructor

    Omelet om = new Omelet();

▶ Field values can be accessed/changed using **dot notation** (if they are public)

    om.eggs = 5;
    int e = om.eggs;
Simple Objects Demo

// Represent data associated with an omelet.
public class Omelet{
    public int eggs; // How many eggs in the omelet
    public int cheese; // How many ounces of cheese
    public double cookedFor; // How long the omelet has cooked
    public String extras; // Extra ingredients added
}

public class OmeletMain{
    public static void main(String args[]){
        Omelet om = new Omelet();
        om.eggs = 3;
        om.cheese = 4;
        om.cookedFor = 4.5;
        om.extras = "ham";
        System.out.printf("Omelet has %d eggs\n",om.eggs);
        System.out.printf("Omelet has %d oz cheese\n",om.cheese);
        System.out.printf("Omelet cooked for %.1f minutes\n",om.cookedFor);
        System.out.printf("Omelet has these extras: %s\n",om.extras);
    }
}

> javac OmeletMain.java
> java OmeletMain
Omelet has 3 eggs
Omelet has 4 oz cheese
Omelet has these extras: ham
Omelet cooked for 4.5 minutes
Multiple Objects

- Each instance of class is its own object, has its own memory
- Can create as many omelets as you have memory for

```java
class TwoOmelets{
    public static void main(String args[]){
        Omelet om = new Omelet();
        om.eggs = 3;
        om.cheese = 4;
        om.cookedFor = 4.5;
        om.extras = "ham";

        Omelet coronary = new Omelet();
        coronary.eggs = 6;
        coronary.cheese = 10;
        coronary.cookedFor = 6.25;
        coronary.extras = "bacon";
    }
}
```
Exercise: Changes to memory

Draw changes to memory for commented lines 1-7

public class TwoOmeletsExercise{
    public static void main(String args[]){
        Omelet om = new Omelet();
        om.eggs = 3;
        om.cheese = 4;
        om.cookedFor = 4.5;
        om.extras = "ham";

        Omelet coronary = new Omelet();
        coronary.eggs = 6;
        coronary.cheese = 10;
        coronary.cookedFor = 6.25;
        coronary.extras = "bacon";

        om.cookedFor += 0.75;  // 1
        coronary.eggs++;
        om.extras = "turkey";
        coronary.extras = "ham";
        coronary.extras + " sausage";
        Omelet x = coronary;  // 5
        x.cookedFor += 1.0;   // 6
        om = new Omelet();    // 7
    }
}