Lecture 03: Getting Started, Part II

Sierra College
CSCI-12
Spring 2015
Mon 02/02/15

Announcements

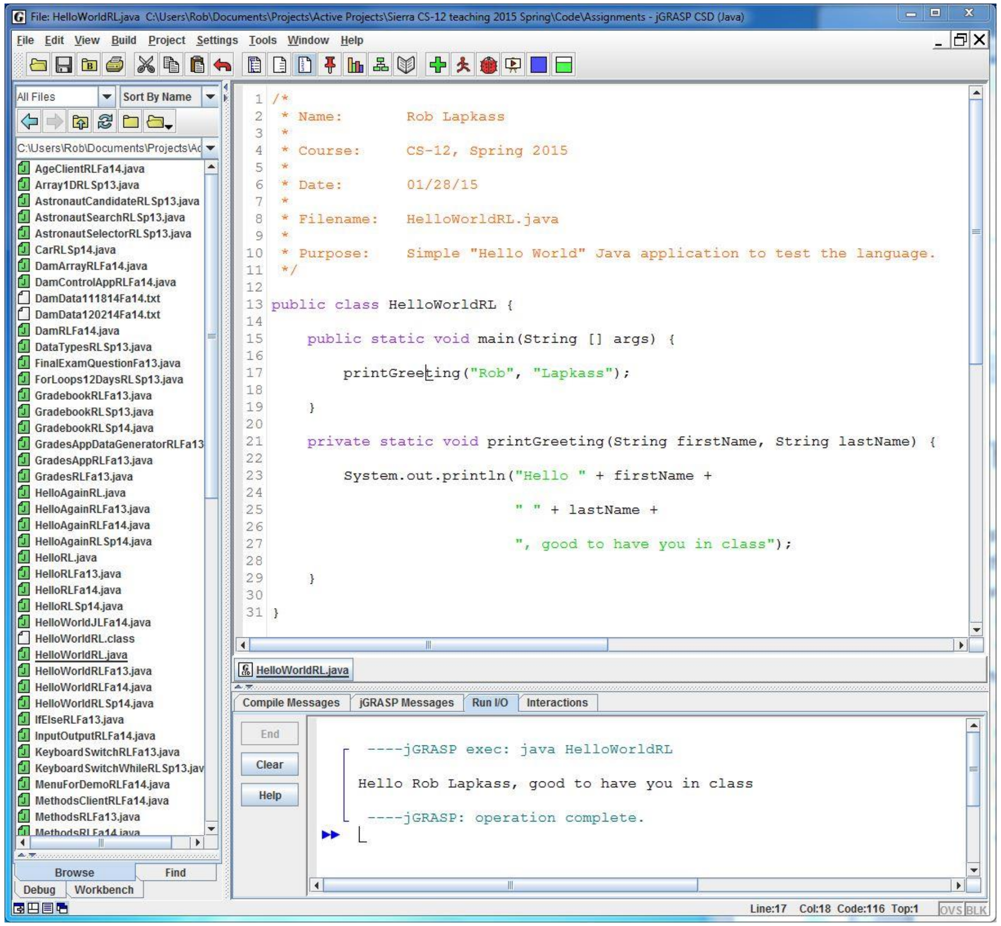
- **General**
  - Office hours starting this week
    - Mondays 8:30-9:30am, 12:30-1:30pm (before/after lectures) in V-105/lab
    - This has been updated in the online syllabus
- **Schedule**
  - Spring Add/Drop/Refund deadline is THIS Sunday 2/8
    - If no assignments are submitted by this deadline, I will consider this as a no-continue decision on your part, and will instructor-drop you from the course
- **Current assignments**
  - HW02: Canvas Intro, due Tues 2/3 @ 11pm (do ALL 3 parts!)
  - LAB02: Hello World, due Tues 2/3 @ 11pm
- **New assignments**
  - HW03: Why Code, due Fri 2/6 @ 11pm
    - Watch a short video (< 6 min), then post responses on discussion board

Lecture Topics

- **Last time**:
  - Hello World (simple getting started program)
- **Today**:
  - Some lessons learned from Hello World
  - An object-oriented revision of Hello World

The “Hello World” Program in jGRASP

- First usage of language with development environment
  - Initial “stick time” with Java + IDE
- Simple, trivial program
  - Make changes, compile, run (repeat)
  - Get program to display output
  - Starting point for experimentation



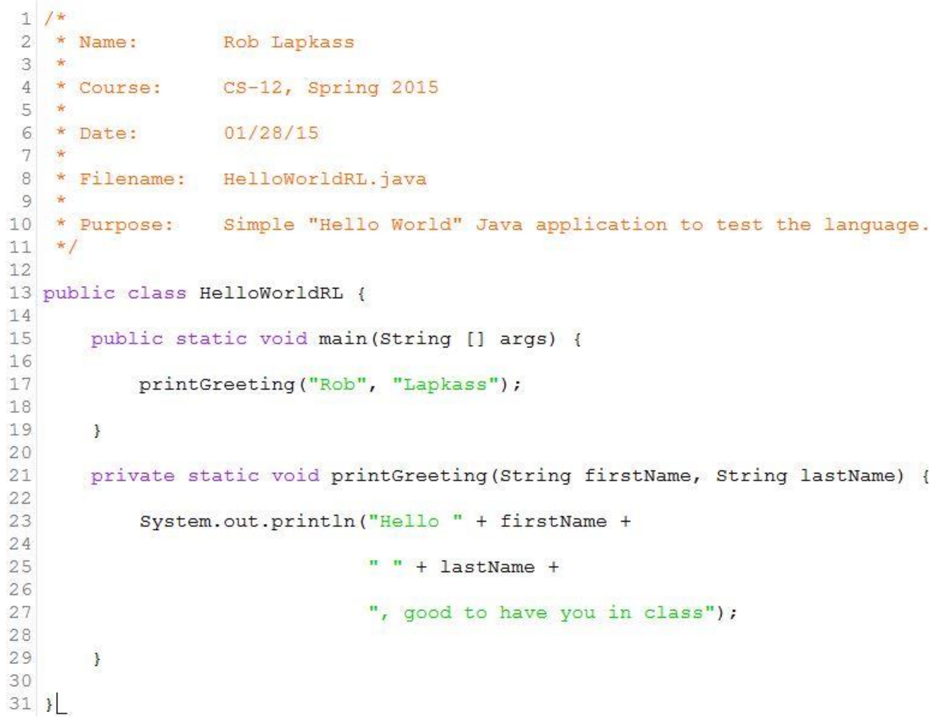
File: HelloWorldRL.java C:\Users\Rob\Documents\Projects\Active Projects\Sierra CS-12 teaching 2015 Spring\Code\Assignments - jGRASP CSD (Java)

```java
/*
 * Name:       Rob Lapkass
 *
 * Course:     CS-12, Spring 2015
 *
 * Date:       01/28/15
 *
 * Filename:   HelloWorldRL.java
 *
 * Purpose:    Simple "Hello World" Java application to test the language.
 */

public class HelloWorldRL {

    public static void main(String [] args) {

        printGreeting("Rob", "Lapkass");

    }

    private static void printGreeting(String firstName, String lastName) {

        System.out.println("Hello " + firstName +

                           " " + lastName +

                           ", good to have you in class");

    }

}
```

```
 ----jGRASP exec: java HelloWorldRL

Hello Rob Lapkass, good to have you in class

 ----jGRASP: operation complete.
```

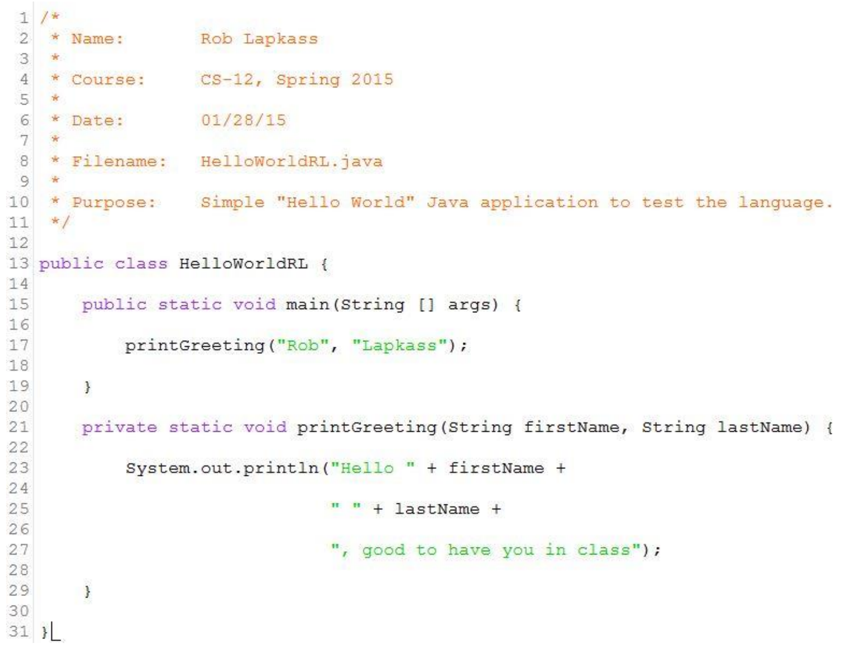
Hello World Takeaway Lessons

- Class name and filename must match exactly
- Spelling matters
- Case sensitivity matters
- All executable code must be within an open/close pair of curly braces
- Semicolon must terminate each statement
- Short, tight, iterative cycle allows for forward progress
  - Make changes, compile, run
- Java compiler flags the smallest errors ("spell check")
- IDE's context-aware colorations helps us to "see" the structure of our code

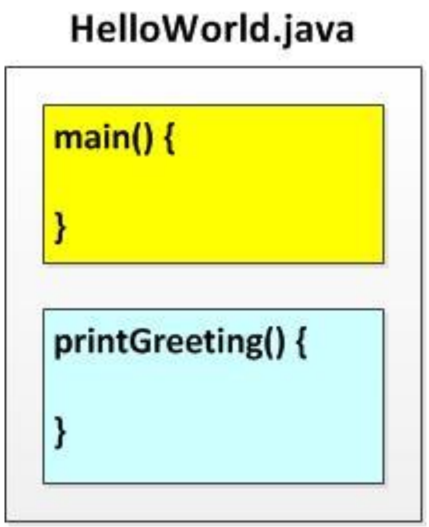


```java
/*
 * Name:        Rob Lapkass
 *
 * Course:      CS-12, Spring 2015
 *
 * Date:        01/28/15
 *
 * Filename:    HelloWorldRL.java
 *
 * Purpose:     Simple "Hello World" Java application to test the language.
 */

public class HelloWorldRL {

    public static void main(String [] args) {

        printGreeting("Rob", "Lapkass");

    }

    private static void printGreeting(String firstName, String lastName) {

        System.out.println("Hello " + firstName +

                           " " + lastName +

                           ", good to have you in class");

    }

}
```

General Structure of Hello World



```
/*
 * Name:        Rob Lapkass
 *
 * Course:      CS-12, Spring 2015
 *
 * Date:        01/28/15
 *
 * Filename:    HelloWorldRL.java
 *
 * Purpose:     Simple "Hello World" Java application to test the language.
 */

public class HelloWorldRL {

    public static void main(String [] args) {

        printGreeting("Rob", "Lapkass");

    }

    private static void printGreeting(String firstName, String lastName) {

        System.out.println("Hello " + firstName +

                           " " + lastName +

                           ", good to have you in class");

    }

}
```



- Header block
  - Java comments, for human users only, not even seen by Java
- A **method** is a callable-by-name “container” for executable Java statements
- 2 methods
  - **main()**: starting point for any Java application, one of these is REQUIRED
  - **printGreeting()**: given first/last names, constructs and prints a message
- The Java **class** HelloWorld is itself the **container** for the two internal methods

Programming Flows of Control

- Our Hello World program demonstrates 2 of the 4 programming flows of control
- **Flows of control** are the different orderings of instructions that a program's logic can take
  - **Sequential execution**
    - Execution of instructions in line-by-line order
    - We see this inside both main() and printGreeting()
  - **Method call**
    - Program control "jumps" to some other named method, then returns
    - We see this when main() "calls" (invokes) printGreeting()
  - Selection (we'll see this in Ch.5)
    - Decision-making, "forks in the road"
  - Looping (we'll see this in Ch.6)
    - Performing same instructions over and over again

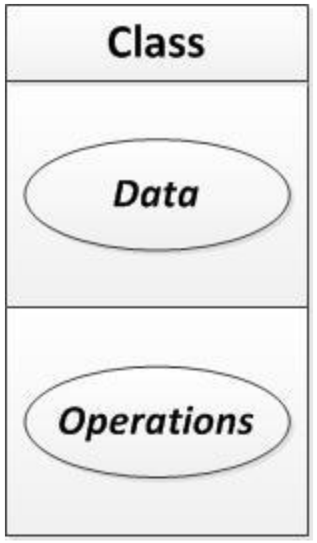
Desired Revision of Hello World

- In a next revision of our Hello World program, we'd like to see the following things:
  - Make the main client program shorter and leaner
  - Carve out the underlying print details, and hide them away elsewhere
  - Reuse the print greeting message for different names
  - Turn "Hello World" into a more traditional object-oriented structure:
    - One short client program class
    - A second, longer utility class

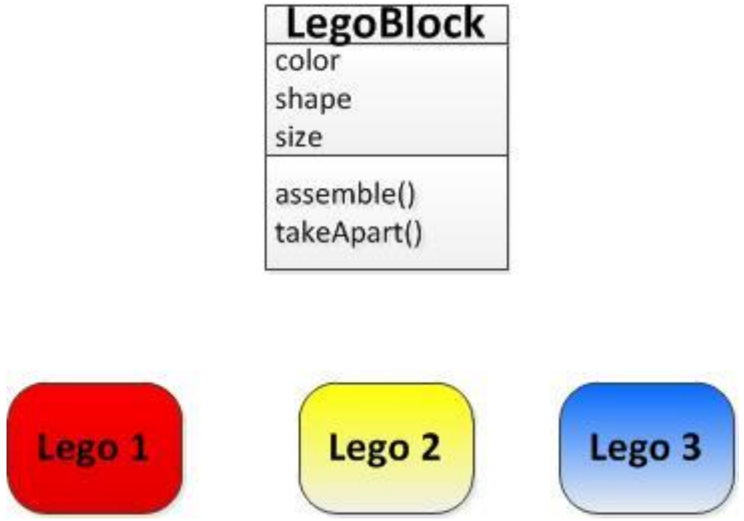
- As review: **a class is the "template" or "blueprint" for a reusable software component, describing some real-world "thing"**

Reprise From Lecture 01

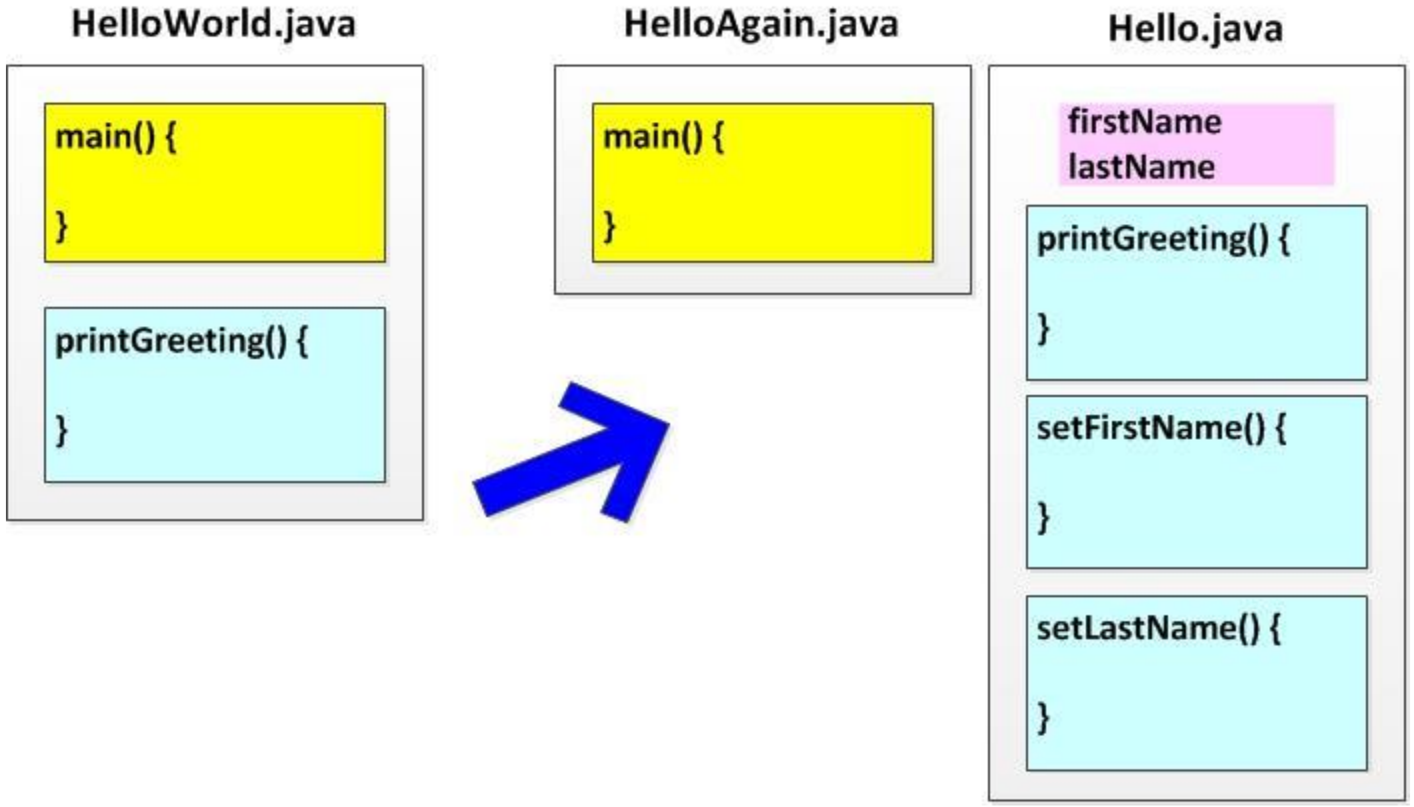
Classes: blueprints for “things” (software components)



Objects: individual “things” created from the blueprint



Revised Structure for Hello Again



Hello Again

- The second lab assignment is a revision of last week's "Hello World" program
  - Write a simple program to display some user-specified text
  - Full details in the assignment handout to be posted in Canvas
  - No hardcopy this time, softcopy only from here on out
    - ***But you may want to print your own hardcopy and bring it to lab***
- We will talk thru it in class
- You will implement it yourself during your lab period

- Review the Hello Again assignment example…

For Next Time

- Lecture Prep
  - Text readings and lecture notes
- Assignments
  - LAB02, due Tuesday
  - HW02, due Tuesday (all 3 parts)
  - Watch the video for HW03 (it’s less than 6 mins)