Wk 1, Day 2, slides:

* Describe the naming convention for widgets
* Explain Large, Medium, and Small TextViews
* Explain different types of EditTexts (numeric, etc.)
* Break away from the slide show (add slides to say demo) to demo:
  + Adding a layout and widgets to a Blank app.
  + Resizing a widget (can’t be done by dragging?)
  + Using a string resource (select using ellipsis)
* At the end of the slide show, before the exercises show them the lab and discuss it.

Lab 1:

Modify part 2 so that it doesn’t require an event handler (no button, since Murach doesn’t cover listeners in ch. 1-4)

Lab 2: Don’t do the pig game yet. (Or at least cover ch. 5 sections on layouts and hiding widgets and anything else relevant in the first week).

* Find a new online pig game. The link in the instructions doesn’t have a game any more.
* Make the pig game v1 lab 3. This is because some of the key concepts like: layouts, enabling and disabling widgets (modify the lab to require this), and images are covered in chapter Murach ch. 5.
* Instead of the pig game, do something simpler maybe a word game or quiz.
  + Require shared preferences be used to store something
  + Require a launcher icon

Chapter 5: In the future, don’t cover every type of widget in the slide show – shorten it to the most important ones.

Chapter 6: Event handlers: we could skip this chapter and have the students just use the one means of implementing handlers

RPS with fragments – Just put the game object in the second fragment. It isn’t needed in the first one or in the activities. Or is it?

Lab 4: Pig with fragments. Consider simplifying this assignment. Maybe eliminate settings and saving state? or make those XC?

Week 3, Day 1: Asynch Task and XML parsing. I moved this to day two because we spent an extra day on Fragments and I extended the due date for the Fragments lab. With my new SimpleFragment example and better management of time in week 2, we shouldn’t have to do this.

Lab 5, (a) tide table v1, in the example and in the instructions, make sure they don’t hard-code the names of the xml files to parse. (b) there seems to be “double-jeopardy) for crashing – modify the grading rubric.

All labs:

* In code quality: I don’t want to see remnants of other apps (mine or the textbook examples). Students should be building apps from scratch, not modifying existing apps (except for the lab assignments with v1, v2, etc.)
* Require students to use their name in the package name and to put their name and date in comments in each source code file.