

Fluency Labs

This series of labs was developed to support and complement Lawrence Snyder's *Fluency with Information Technology: Skills, Concepts & Capabilities*. The labs guide students through small-scale activities, each focused on a small number of learning goals. They support the textbook in the sense that they offer contexts for practicing skills and applying concepts discussed in the textbook chapters. For example, labs are devoted to practicing algorithmic thinking, relational database tables, and applying fundamental programming concepts, including conditionals, functions, and iteration. The labs also serve a complementary role in teaching skills that are *not* covered extensively in the textbook. Specific examples include creating forms in a database—a skill we believe is better suited for learning from active, hands-on experience, rather than textbook reading.

The labs are designed in a sequence that parallels the textbook chapters. With the exception of Part II, whose content is more heavily concept-oriented, each part of the book has at least four accompanying labs, most of which correspond directly to specific chapters. Prerequisite readings and labs are specified for each lab.

Each lab includes introductory materials for the instructor and the student. For the instructor, basic technical requirements for conducting the lab are specified. We have tried to make the labs as generic as possible, so they should require minimal customization for local use. However, a few labs (e.g., Lab 1.2 on Internet Basics) do require the instructor to determine technical details specific to their local campus (e.g., campus network characteristics, lab computer configuration), and those labs include pre-lab setup instructions.

Of equal interest to the instructor and students, most labs also begin with a list of key learning goals (categorized according to the skills/concepts/capabilities content classification used in the textbook), list of key vocabulary, and post-lab questions. The post-lab questions are written to be suitable for written responses, but instructors might also consider using them as starting points for in-class discussion, whether in small groups or with the class as a whole. As implied by instructions asking students to read the post-lab questions *before* beginning the lab procedure, these questions are also intended as concrete (if not always comprehensive) focal points for directing learning through the course of the lab. As students proceed through the lab, keeping these post-lab questions in mind should help them focus their attention on the key content and/or encourage deeper reflection on the relevant topics than minimally required to complete the lab. In this sense, we hope the post-lab questions implicitly supplement the more abstract learning goals listed with each lab.

Example solutions are given for most of the post-lab questions. However, instructors should recognize that some of the questions are intentionally broad or open-ended. For such questions, we suggest that assessment focus on quality and sophistication of discussion and argument, rather than on the expectation of fixed, “correct” answers.

Separate from the post-lab questions, some lab procedure and discussion include short-answer questions that students are expected to answer as they work through the lab. Many of these in-lab questions are for recording observations, e.g., what happens when an HTML file or database table is changed in a particular way. The questions often ask students to interpret or explain these observations. Other in-lab questions are designed to engage students in making predictions prior to experimenting with the computer.

Although the in-lab and post-lab questions were primarily designed to encourage students to focus on and reflect on the lab process, instructors may elect to collect written responses to both the in-lab and post-lab questions for assessment or evaluation purposes. In the process of completing some of the labs, students will produce artifacts such as databases and Web pages with JavaScript programs, and instructors can also collect these electronically. Finally, instructors can ask students to submit (in printed or electronic form) screen snapshots to document important steps in the lab procedures. The labs are provided without specific instructions for submitting student work, leaving these decisions to the instructor's discretion.

The Excel labs corresponding to chapter 13 were written by Mark Frydenberg of Bentley College specifically for *Fluency with Information Technology: Skills, Concepts & Capabilities*, and were originally included only on the first edition's Instructor Resource Disk. The labs have been included on the Companion Website for user ease, and instructors can download the solutions from the Addison-Wesley Instructor Resource Center at <http://www.aw-bc.com/irc>.

Recommended Procedure for Screen Snapshots under Windows. All recent versions of Windows allow the user to take a snapshot of the currently active window or the whole desktop by using the **PrtScn** (Print Screen) key. To take a snapshot of the current window, press **Alt-PrtScn**. To take a snapshot of the whole desktop (to capture all visible and open windows), press **Ctrl-PrtScn**. There will be no visible or audible indication that anything has happened at this point, but an image of the window or desktop contents has been copied. Open Wordpad and select **Edit \ Paste** or press **Ctrl-v**, which should paste the window/desktop snapshot into the document. This document can then be saved and printed for submission. Larger snapshots might get cut off when printing, so use Wordpad's Print Preview (via the **File** menu) to check this before printing. If the snapshot is too large, try switching the document layout to landscape orientation (i.e., "printing sideways"), under **File \ Page Setup...** If the snapshot is still too large to fit on the page, resize the image in Wordpad by dragging the small square handles at the corners and sides of the image.