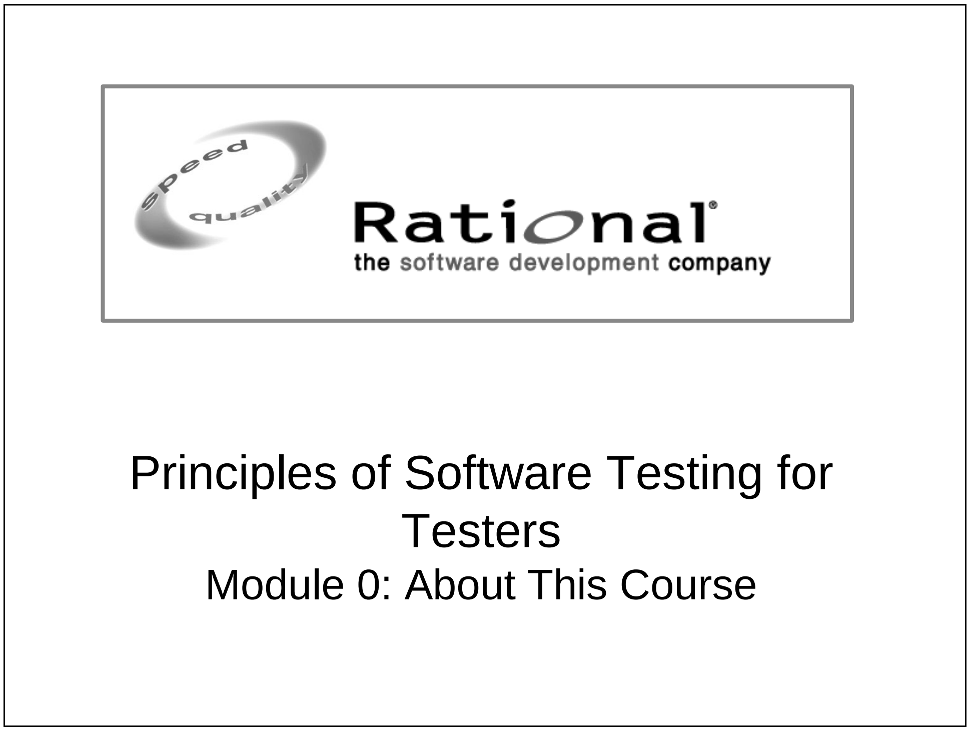
ü ü ü

Module 0
About This Course

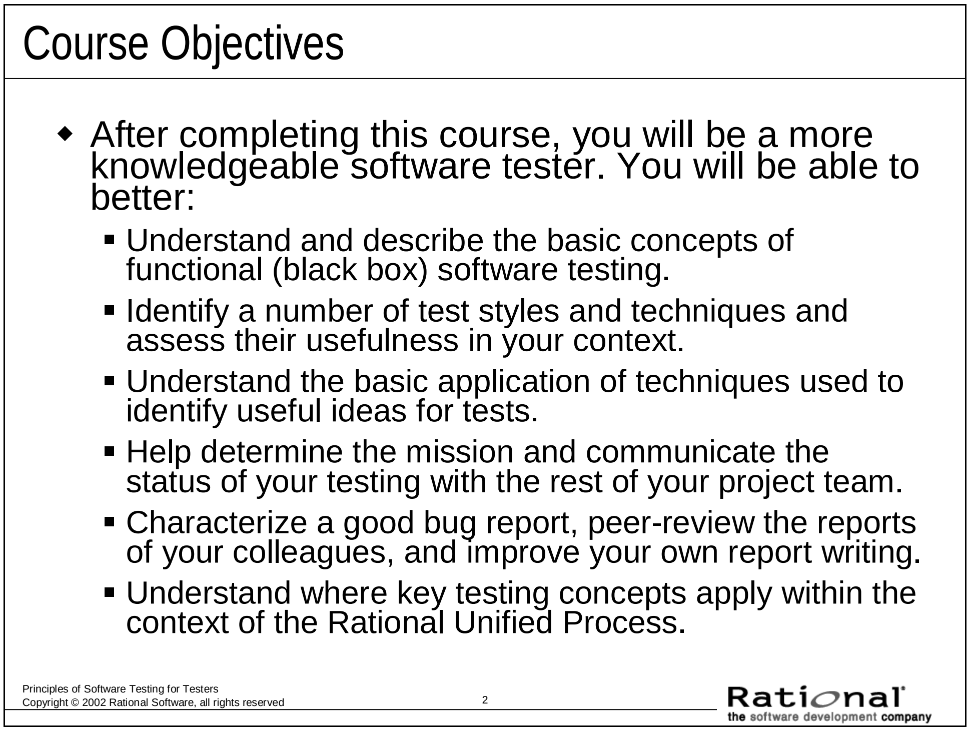

speed
quality
Rational
the software development company

Principles of Software Testing for Testers

Module 0: About This Course

Topics

Course Objectives



Course Objectives

- After completing this course, you will be a more knowledgeable software tester. You will be able to better:
  - Understand and describe the basic concepts of functional (black box) software testing.
  - Identify a number of test styles and techniques and assess their usefulness in your context.
  - Understand the basic application of techniques used to identify useful ideas for tests.
  - Help determine the mission and communicate the status of your testing with the rest of your project team.
  - Characterize a good bug report, peer-review the reports of your colleagues, and improve your own report writing.
  - Understand where key testing concepts apply within the context of the Rational Unified Process.

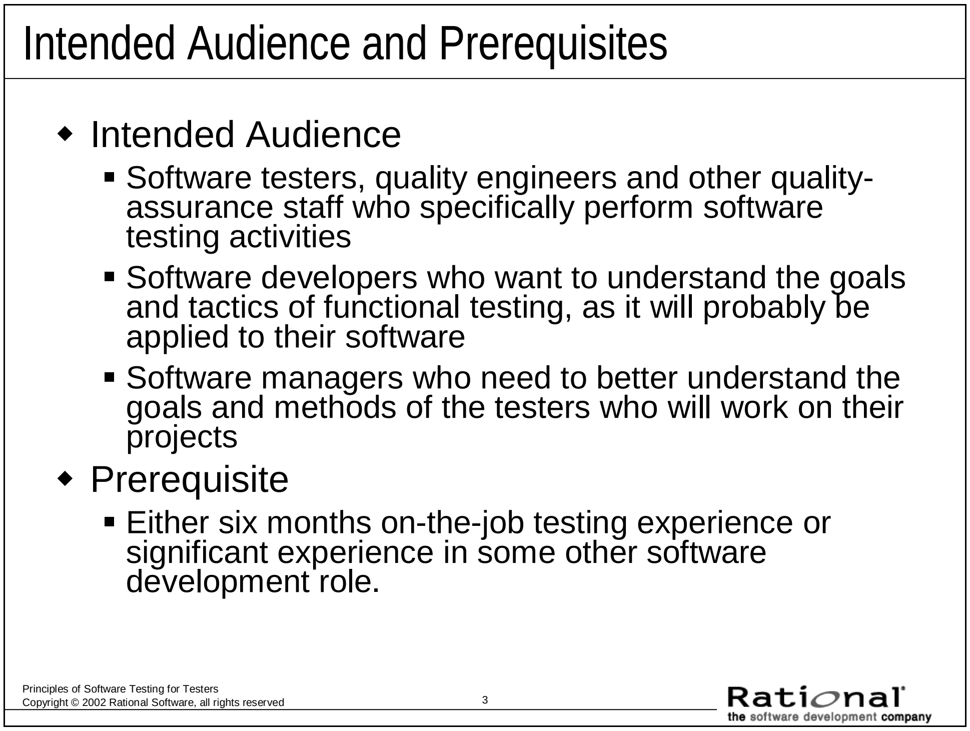
Principles of Software Testing for Testers

2

Rational
the software development company

Note that there are other courses in the Rational University curriculum that focus on using Rational tools, and that this course does not. Rather, this course focuses on the concepts, process, and practices that will help you be a better software tester.

Intended Audience and Prerequisites

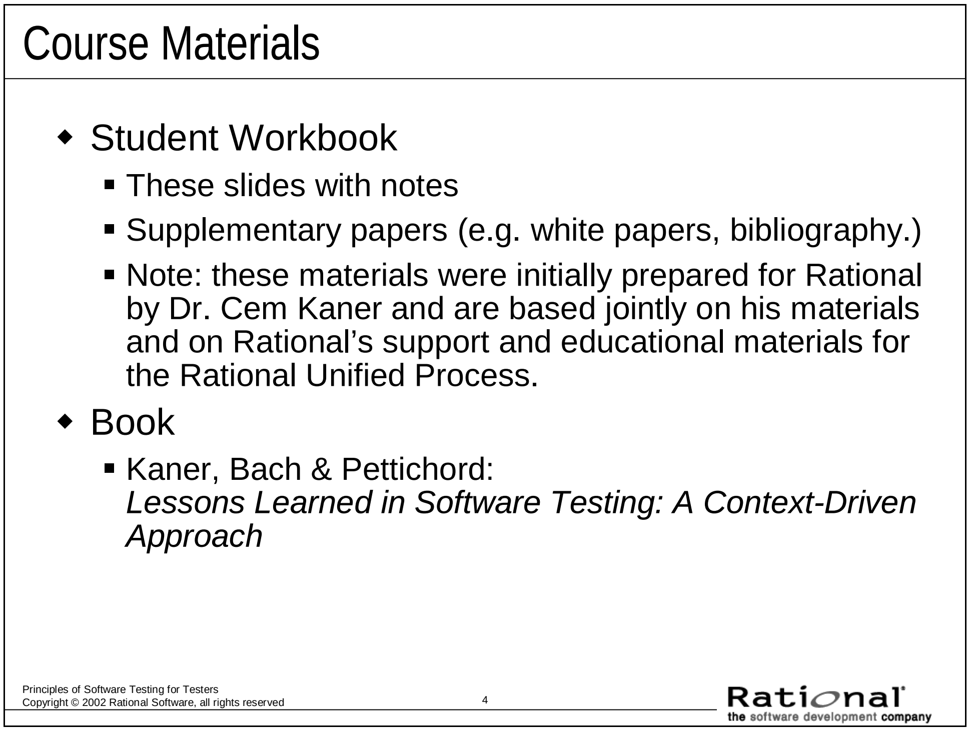


Intended Audience and Prerequisites

- Intended Audience
  - Software testers, quality engineers and other quality-assurance staff who specifically perform software testing activities
  - Software developers who want to understand the goals and tactics of functional testing, as it will probably be applied to their software
  - Software managers who need to better understand the goals and methods of the testers who will work on their projects
- Prerequisite
  - Either six months on-the-job testing experience or significant experience in some other software development role.

Rational
the software development company

Course Materials



Course Materials

- Student Workbook
  - These slides with notes
  - Supplementary papers (e.g. white papers, bibliography.)
  - Note: these materials were initially prepared for Rational by Dr. Cem Kaner and are based jointly on his materials and on Rational’s support and educational materials for the Rational Unified Process.
- Book
  - Kaner, Bach & Pettichord:
    Lessons Learned in Software Testing: A Context-Driven Approach

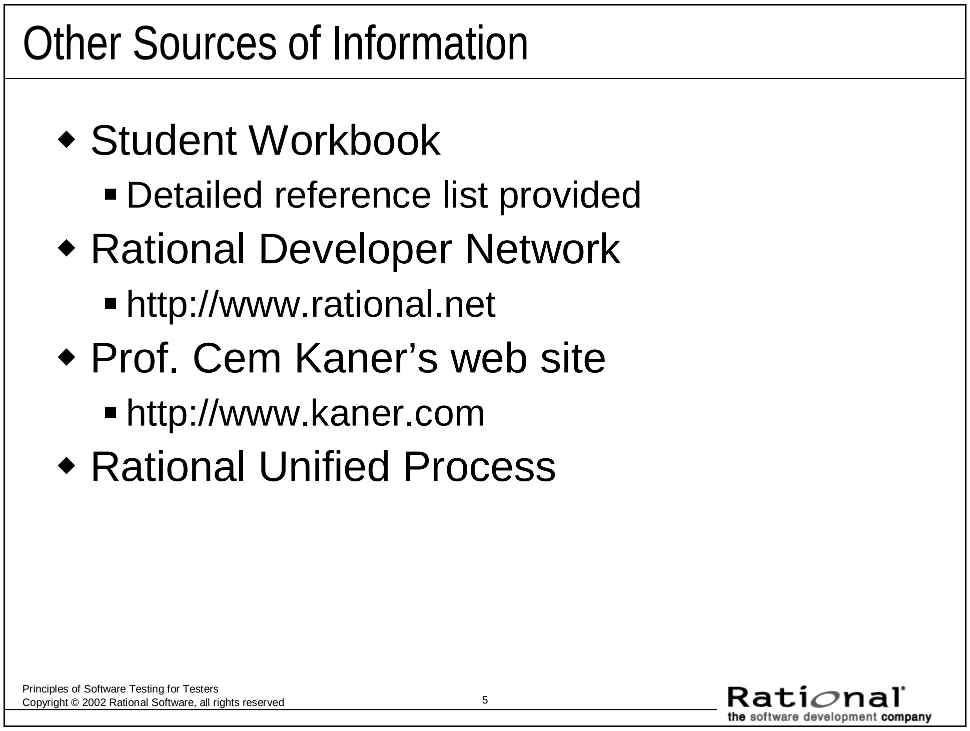
Principles of Software Testing for Testers

4

Rational
the software development company

For a complete listing of sources referenced in the creation of this course, please see the bibliography included in the supplementary papers.

Other Sources of Information



Other Sources of Information

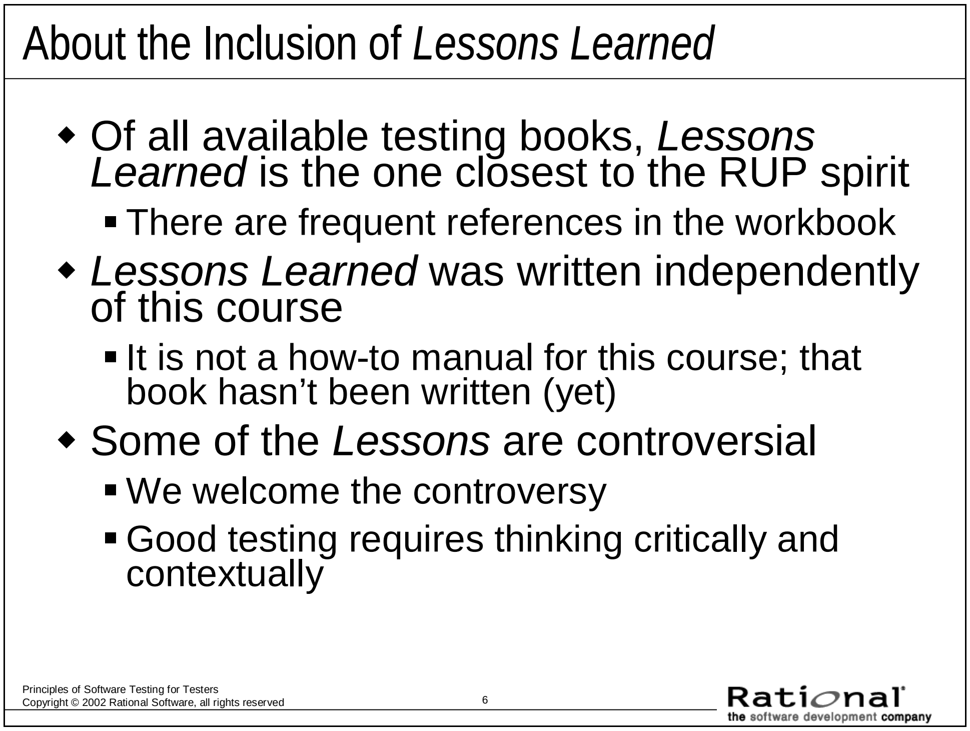
- Student Workbook
  - Detailed reference list provided
- Rational Developer Network
  - http://www.rational.net
- Prof. Cem Kaner’s web site
  - http://www.kaner.com
- Rational Unified Process

Principles of Software Testing for Testers

5

Rational
the software development company

About the Inclusion of Lessons Learned



About the Inclusion of *Lessons Learned*

- Of all available testing books, *Lessons Learned* is the one closest to the RUP spirit
  - There are frequent references in the workbook
- *Lessons Learned* was written independently of this course
  - It is not a how-to manual for this course; that book hasn't been written (yet)
- Some of the *Lessons* are controversial
  - We welcome the controversy
  - Good testing requires thinking critically and contextually

Principles of Software Testing for Testers

6

From the Preface to *Lessons Learned*, pp xx-xxi:

In our world, software development teams often work under ambitious deadlines, discovering what needs to be done at the same time as they're discovering how to do it. Sometimes their approaches are more formal, sometimes less. It depends on a wide range of circumstances.

We follow the *context-driven approach* in software testing. We expect that a method that works wonderfully under some circumstances will not work under others. Rather than talk about best practices, we talk about practices that are well suited to the context at hand. We discuss the context-driven approach at the end of the book, but in essence, context-driven testing looks at the "what" of testing (the techniques, tools, strategies, and so on) in terms of "who", "when", "where", "why", and "what if."

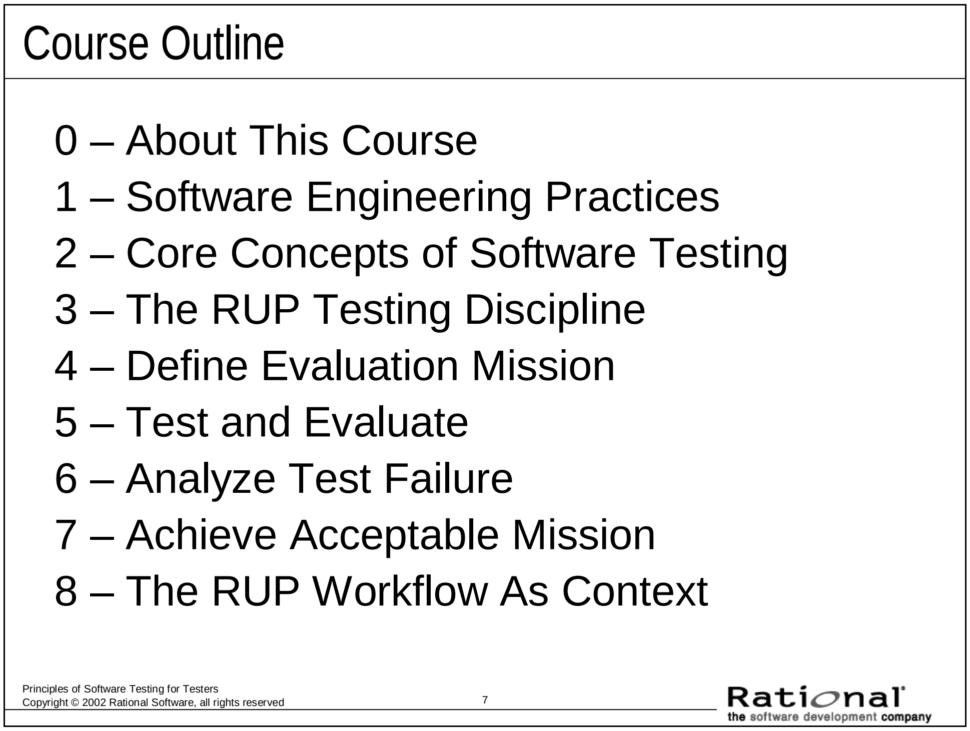
Our goal is to match our selection of practices to the circumstances at hand in a way that achieves excellent testing. We don't expect to achieve great testing by taking over the project, nor by stamping our feet and telling the project manager (or executives) how Real Professionals would run the project. We don't expect to achieve great testing by intimidating programmers, nor by being obsequious to them. We don't expect to do great testing by filling out thousands of little pieces of paper (or comparable electronic records), nor by wasting everyone else's time on unnecessary processes.

We don't have a political or a bureaucratic or a formal-methods recipe for great testing

That recipe doesn't exist!

We think great testing involves skillful technical work (searching for defects) and accurate, persuasive communication.

Course Outline



Course Outline

0 – About This Course

1 – Software Engineering Practices

2 – Core Concepts of Software Testing

3 – The RUP Testing Discipline

4 – Define Evaluation Mission

5 – Test and Evaluate

6 – Analyze Test Failure

7 – Achieve Acceptable Mission

8 – The RUP Workflow As Context

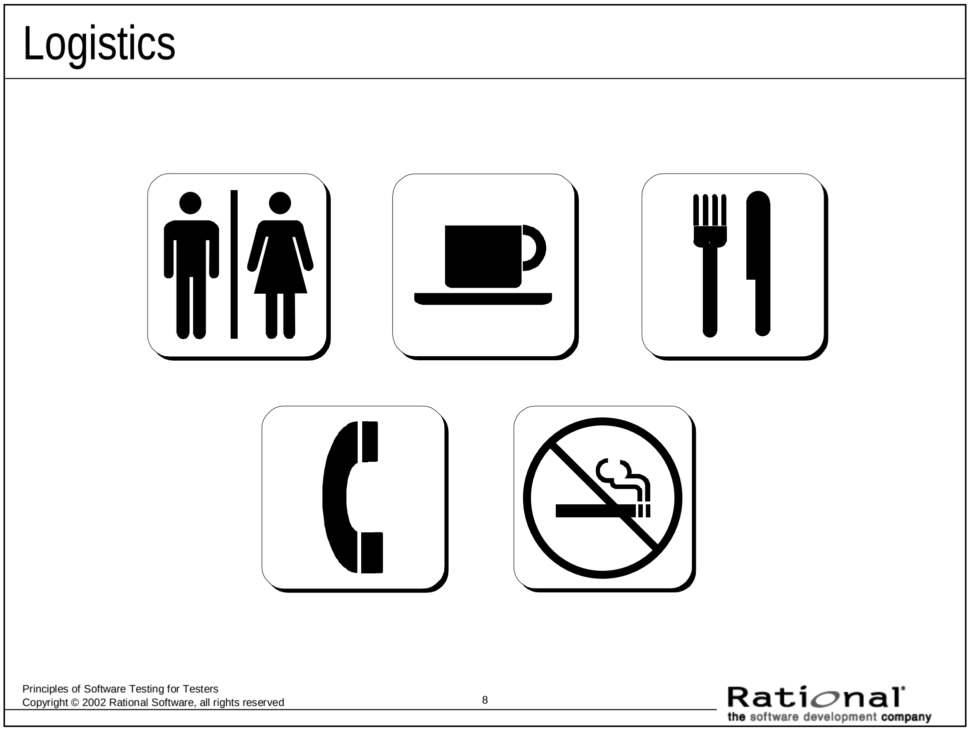
Principles of Software Testing for Testers

7

Rational
the software development company

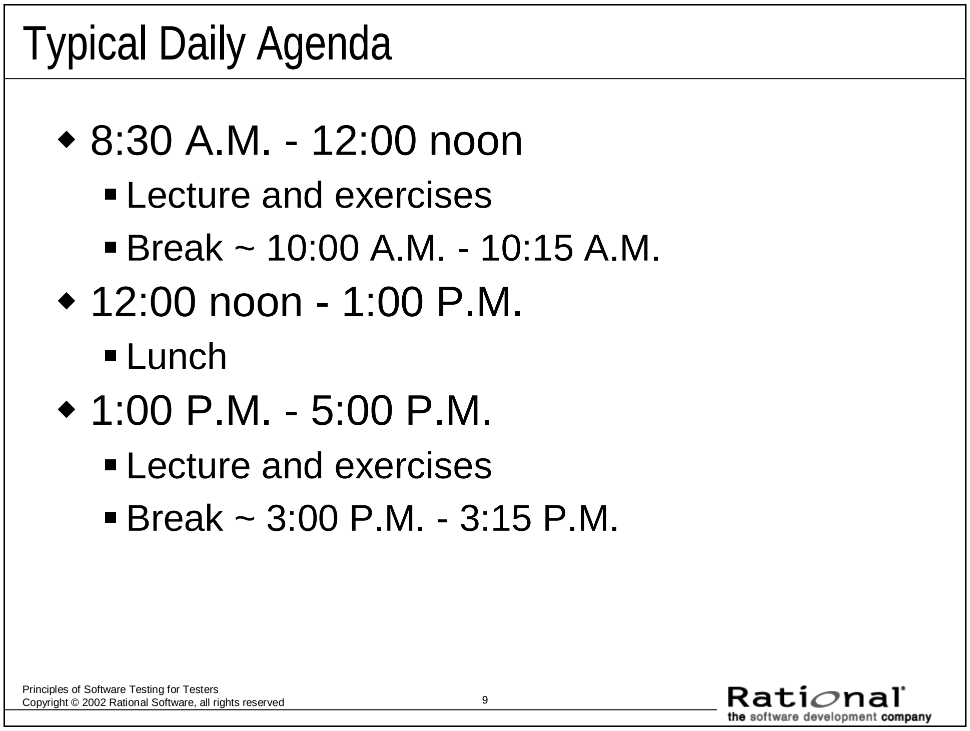
This course will introduce various key concepts incrementally, in a “fundamentals first” delivery. Initially, each concept will be discussed somewhat in isolation, and progressively “woven” into a collective framework of concepts as the course progresses.

Logistics



Logistics

Typical Daily Agenda



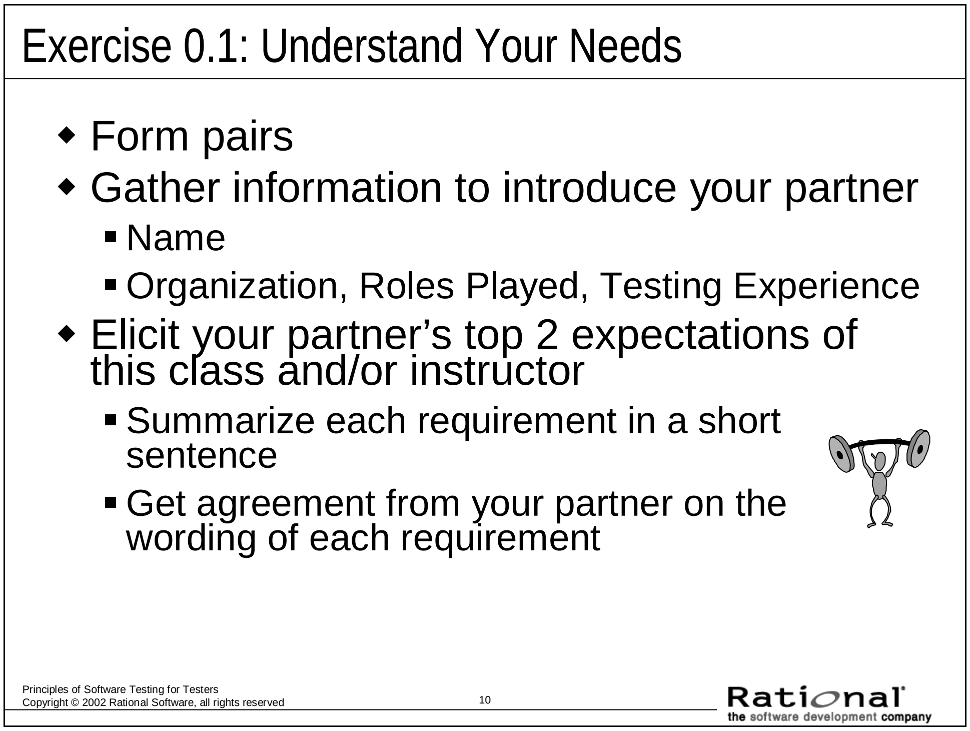
Typical Daily Agenda

- 8:30 A.M. - 12:00 noon
  - Lecture and exercises
  - Break ~ 10:00 A.M. - 10:15 A.M.
- 12:00 noon - 1:00 P.M.
  - Lunch
- 1:00 P.M. - 5:00 P.M.
  - Lecture and exercises
  - Break ~ 3:00 P.M. - 3:15 P.M.

Principles of Software Testing for Testers

9

Exercise 0.1: Understand Your Needs



Exercise 0.1: Understand Your Needs

- Form pairs
- Gather information to introduce your partner
  - Name
  - Organization, Roles Played, Testing Experience
- Elicit your partner's top 2 expectations of this class and/or instructor
  - Summarize each requirement in a short sentence
  - Get agreement from your partner on the wording of each requirement

Principles of Software Testing for Testers

10

Rational the software development company

The purpose of this activity is to:

1. Help your instructor understand your expectations for this course.
2. Introduce your fellow students to the instructor and to the class.
3. Baseline the class's definition of quality under which they toil.

Some sample questions you may want to ask are:

- What difficulties or obstacles do you hope to reduce by attending this class?
- What are your expectations of the course?
- Who will be the main recipient of what you learn?