Instructor Notes:

Introduce the outline of the course.

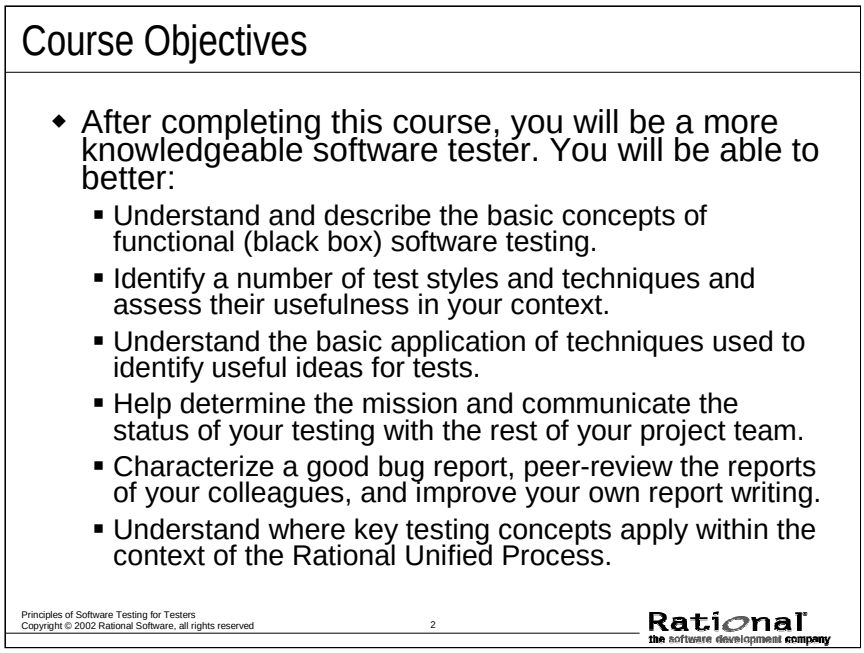
Explain to students that the following module (Module 0) is included in all Rational University methodology courses to give an overview of software development.

To complete this course in the time available, it is important that you cover this module briefly; do not get bogged down in the details. The extra information is there to provide context and further information to be read later. You can offer to discuss further during lunch or after class.

Principles of Software Testing for Testers

Module 0: About This Course

Instructor Notes:



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.

Instructor Notes:

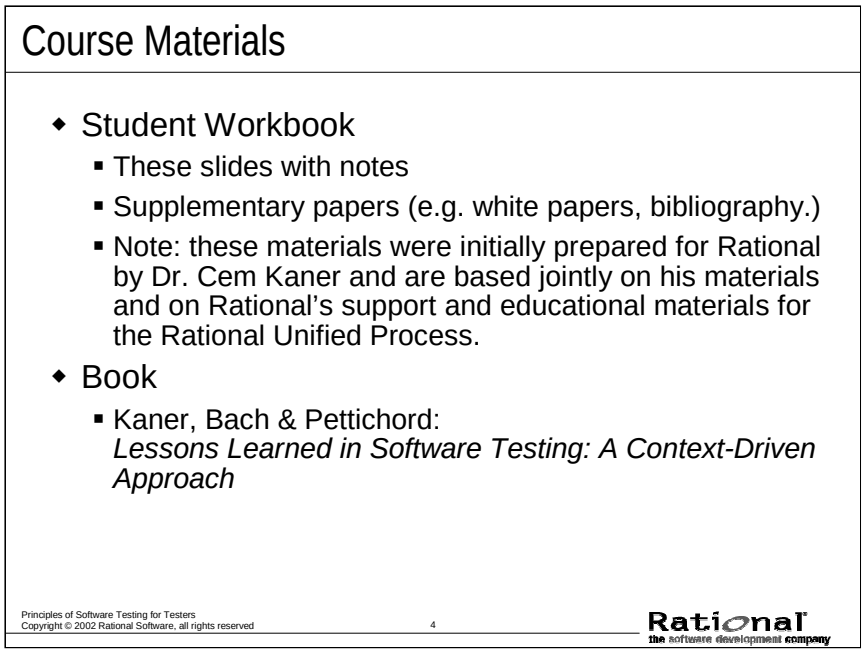
Intended Audience and Prerequisites

- 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.

Principles of Software Testing for Testers
Copyright © 2002 Rational Software, all rights reserved

Rational
the software development company

Instructor Notes:



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

Instructor Notes:

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
Copyright © 2002 Rational Software, all rights reserved

Rational
the software development company

Instructor Notes:

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
Copyright © 2002 Rational Software, all rights reserved

Rational
the software development company

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

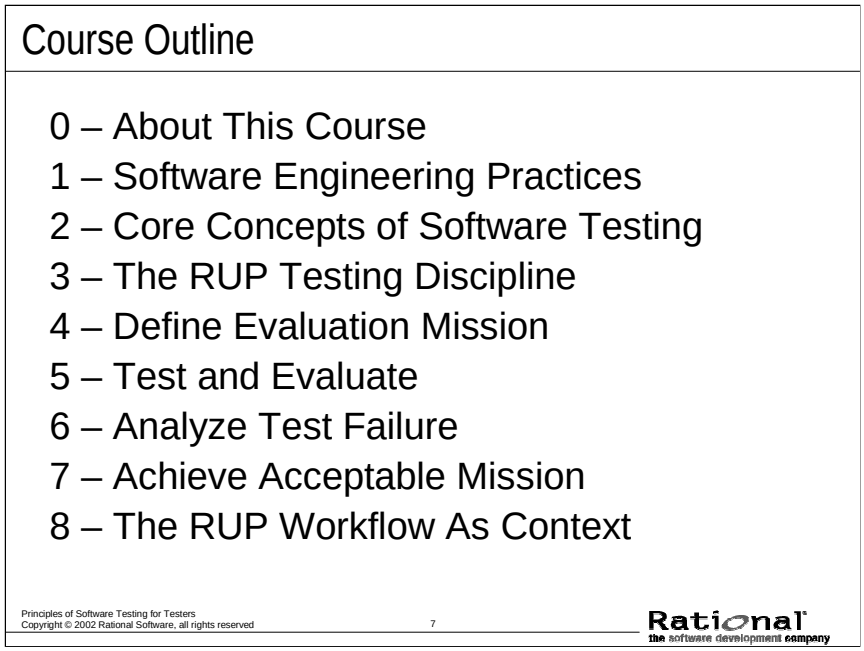
Instructor Notes:

Introduce the outline of the course.

Note the comments in the student notes. These comments are designed to ease the student into the somewhat "non-linear" style of delivery the course uses.

Explain to students that the following module (Module 1) is included in a similar format in all Rational University methodology courses to give an overview of software development. The idea is to level-set the classes understanding of these practices, and discuss them as they apply to the specific focus of this course.

To complete this course in the time available, it is important that you cover the first modules briefly. This may be a challenge!



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.

Instructor Notes:

Logistics

Instructor Notes:

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
Copyright © 2002 Rational Software, all rights reserved

Rational
the software development company

Instructor Notes:

Use this activity to introduce the students to each other, to find out the students' needs.

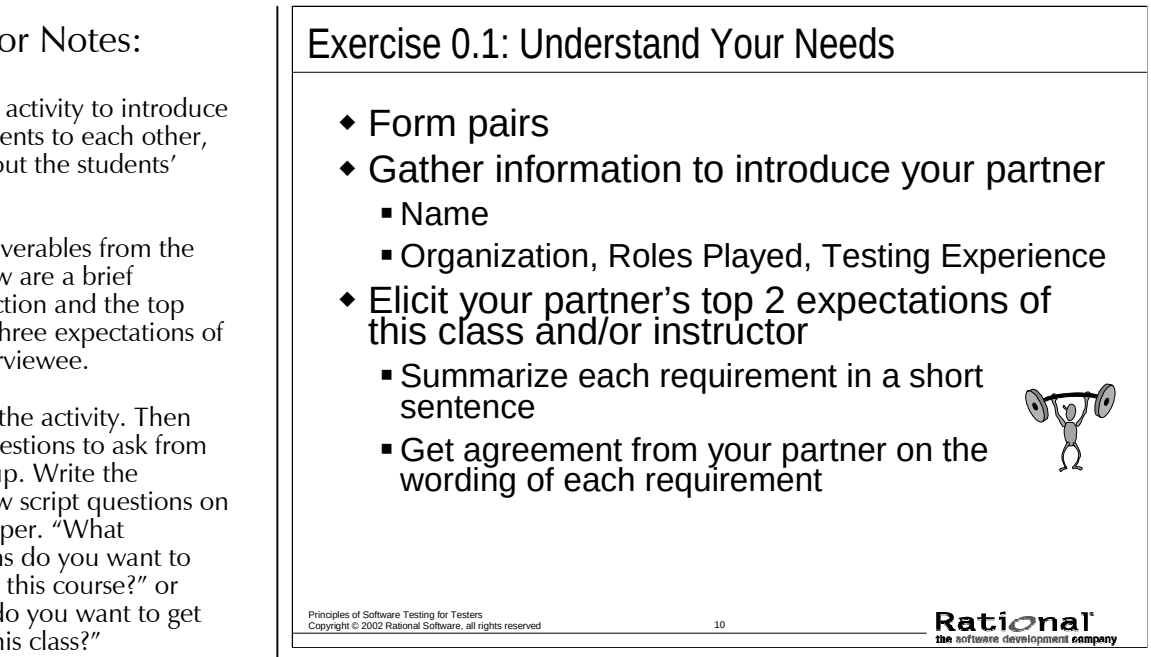
The deliverables from the interview are a brief introduction and the top two or three expectations of the interviewee.

Explain the activity. Then elicit questions to ask from the group. Write the interview script questions on easel paper. "What problems do you want to cover in this course?" or "What do you want to get out of this class?"

Break the class members into pairs of people who don't know each other. In offsite classes, pair up by different departments.

Allow 10-12 minutes with interviewer-interviewee and then reverse roles for another 10-12 minutes. Use a timer.

Gather the group and have each participant briefly introduce and report their customer's top priorities. Write the top priorities on easel paper; for repeats, mark another vote for an item.



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?