# BBST:
# Evolving a Course in
# Black Box Software Testing

Cem Kaner, J.D., Ph.D.

Florida Institute of Technology

January 2008

Advisory Board Meeting, Testing Course Project

These notes are partially based on research that was supported by NSF Grants EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers" and CCLI-0717613 "Adaptation & Implementation of an Activity-Based Online or Hybrid Course in Software Testing." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

# Overview

- The primary objective of the NSF project is adaptation and implementation of the BBST course, in several markets:
  - Academic (traditional)
  - Academic (online)
  - In-house training by in-house staff
  - Commercial training (face-to-face or online)
- To support / sustain these goals, we are working on several tasks, such as:
  - Improve the materials
  - Create supporting materials, e.g. collections of activities
  - Create a self-sustaining BBST instructor community

# Instructional formats of BBST

- I've done these:
  - Non-credit (or certificate-credit) face-to-face professional development (public courses and within-company)
  - Academic credit (undergrad/grad) face-to-face traditional lecture *(Florida Tech)*
  - Academic credit hybrid (video lecture, live coaching) *(Florida Tech)*
  - Professional development, purely online, with extensive assessment *(AST series)*
- I haven't done these, but maybe someone else has
  - Professional development, face-to-face, with extensive assessment *(Ajay will talk about this)*
  - Academic credit, purely online

Quick skim of the section header pages

# OVERVIEW

# What makes THIS BBST worth the effort?

- BBST combines several ideas about how to teach well
  - I don't think any of the individual ideas are original
  - I think the combination is pretty good
  - I think the following have been success factors for the course:

# Success factors

1. Strong content
2. Story-based teaching
3. Detailed examples
4. Video lectures
5. In-class activities that tie to the lecture
6. Application to a real product under test
7. Orientation exercises
8. Open book quizzes
9. Study-guide based exam
10. Challenging but focused assignments
11. Task scaffolding
12. Peer review
13. Explicit discussions of learning issues in the course design.
14. Open discussion of (employment) value of the material and the work
15. Organic evolution of the class (rather than process-constrained)
16. Enthusiasm and ongoing renewal (Hawthorne effect)
17. Instruction on test-taking skills
18. SALG feedback

# Should be success factors

I know these should make the course better, but I haven't succeeded in figuring out how:

1. Drill / problem sets, to help students
   - Experience worked examples
   - Develop skills through practice
   - Experience an underlying common core when there is a lot of more superficial variation
2. Paired testing
3. Testing competitions
4. Student presentations
5. Employer / famous-person visitors

# Current challenges

1. Disappointing essay exams
2. Getting students to watch videos in advance
3. Getting students to do preparatory exercises
4. Coping with an unstandardizable vocabulary
5. Classroom time management (discussion versus lab time)
6. Videos feature one white man
7. No active discussion in the videos
8. On-the-record videos make some storytelling difficult
9. Synchrony is important when students rely on each other
10. Grading time is substantial
11. Activities are hard to design
12. Multiple choice pool is small
13. We need better assigned readings
14. Student prerequisites

# Additional activity

- BBST instructors course
- AST adaptation of BBST
- Tester certification

# Project assessment

- Project task tracking
- SALG characterizations
- Blind comparisons of final exams across courses (how do the answers rank, across courses)
- Open comparisons of final exams across courses (how are the answers similar or different across courses)
- Employer reactions X months later
- Student reactions X months later
- Instructor reactions
- Adoption statistics?
- Peer review / external evaluation
- ??? What else ???

# Where we're going

- Opportunities for:
  - Broad collaboration across industry / academic and across academic institutional boundaries
  - Broad data pool
  - Grants to support collaboration and assessment
  - Commercial profit
  - A broader pool of people achieving technician-level or higher-level entry into the skilled workforce
- More general benefits
  - Activities pool more broadly useful in SE education
  - Instructional methods more broadly applicable

# What the project needs to work on

- Additional venues (including access to assessment data)
- Figuring out what assessment data we should collect
- Actually doing the analyses
- New models for video content (and doing the videos)
- Publishable pools of activities
- The Oxford English analogy for documenting testing vocabulary
- Funding to support additional venues
- Funding to support administration of the project
- Active collaborators on the instructors' course
- Writing up what we're learning

# CONTEXT OF THE COURSE

# Context of the problem

1. Testing is evolving slowly because there is so little educational support for it.
2. University support will continue to be inadequate for the foreseeable future. Companies will therefore have to develop their own training strategies.
3. Commercial short courses are often ineffective because they
   - try to cover too much,
   - at too shallow a level,
   - without application to the learner's specific situation,
   - with too little opportunity for practice,
   - and less opportunity for assessment and feedback.

My idea has been to develop courses in an academic environment (where I can learn more about what works and why), with the goal of providing an alternative model for commercial (in-house) training and professional self-study

| Commercial | Academic |
|---|---|
| Drive-by teaching: 2-5 days, rapid-fire ideas, visiting instructor | Local teaching: Several months, a few hours per week, students get to know instructor |
| Broad, shallow coverage | Deeper coverage |
| Time constraints limit activities | Activities expected to develop skills |
| No time for homework | Extensive homework |
| No exams | Assessment expected |
| Coached, repeated practice seen as time-wasting | Coached, repeated practice is highly appreciated |
| Familiarity | Capability |
| Work experience helps to bring home concepts | Students have no work experience, need context |
| Richer grounding in real practice | Harder to connect to real practice |
| Some (occasional) student groups share a genuine, current need | Students don't naturally come to a course as a group with a shared problem |
| Objective: one applicable new idea per day | Expect mastery of several concepts and skills |

# The instructional challenge, as I see it

*Software testing*

*is cognitively complex,*

*requires critical thinking,*

*effective communication, and*

*rapid self-directed learning.*

# Software testing:

Empirical, technical investigation of the product under test conducted to provide stakeholders with quality-related information.

**Empirical**

- We run experiments (tests). Code inspections are valuable, but they are not tests.

**technical**

- We use technical means, including experimentation, logic, mathematics, models, tools (testing-support programs), and tools (measuring instruments, event generators, etc.)

**investigation**

- an organized and thorough search for information
- this is an active process of inquiry. We ask hard questions (aka run hard test cases) and look carefully at the results

**provide quality-related information**

- see next slide (information objectives)

# Information Objectives

- Find important bugs, to get them fixed
- Assess the quality of the product
- Help managers make release decisions
- Block premature product releases
- Help predict and control costs of product support
- Check interoperability with other products
- Find safe scenarios for use of the product
- Assess conformance to specifications
- Certify the product meets a particular standard
- Ensure the testing process meets accountability standards
- Minimize the risk of safety-related lawsuits
- Help clients improve product quality & testability
- Help clients improve their processes
- Evaluate the product for a third party

Different objectives require different test techniques and strategies.
They will yield different tests, different test documentation and different test results.

# Test techniques: a few examples

- Function testing
- Specification-based testing
- Domain testing
- Risk-based testing
- Scenario testing
- Regression testing
- Stress testing
- User testing
- State-model based testing
- High-volume automated testing

**To different degrees, good tests have these attributes:**

- **Power**. When a problem exists, the test will reveal it.
- **Valid**. When the test reveals a problem, it is a genuine problem.
- **Value**. It reveals things your clients want to know about the product or project.
- **Credible**. Your client will believe that people will do the things that are done in this test.
- **Representative** of events most likely to be encountered by the user. (xref. Musa's Software Reliability Engineering).
- **Motivating**. Your client will want to fix the problem exposed by this test.
- **Performable**. It can be performed as designed.
- **Maintainable**. Easy to revise in the face of product changes.
- **Repeatable**. It is easy and inexpensive to reuse the test.
- **Pop**. (short for Karl Popper) It reveal things about our basic or critical assumptions.
- **Coverage**. It exercises the product in a way that isn't already taken care of by other tests.
- **Easy to evaluate.**
- **Supports troubleshooting.** Provides useful information for the debugging programmer.
- **Appropriately complex.** As the program gets more stable, you can hit it with more complex tests and more closely simulate use by experienced users.
- **Accountable**. You can explain, justify, and prove you ran it.
- **Cost**. This includes time and effort, as well as direct costs.
- **Opportunity Cost.** Developing and performing this test may prevent you from doing other tests (or other work)

# Contexts Vary Across Projects

Testers must learn, for each new product:

- What are the goals and quality criteria for the project
- What skills and resources are available to the project
- What is in the product
- How it could fail
- What the consequences of potential failures could be
- Who might care about which consequence of what failure
- How to trigger a fault that generates the failure we're seeking
- How to recognize failure
- How to decide what result variables to pay attention to
- How to decide what other result variables to pay attention to in the event of intermittent failure
- How to troubleshoot and simplify a failure, so as to better
  - (a) motivate a stakeholder who might advocate for a fix
  - (b) enable a fixer to identify and stomp the bug more quickly
- How to expose, and who to expose to, undelivered benefits, unsatisfied implications, traps, and missed opportunities.

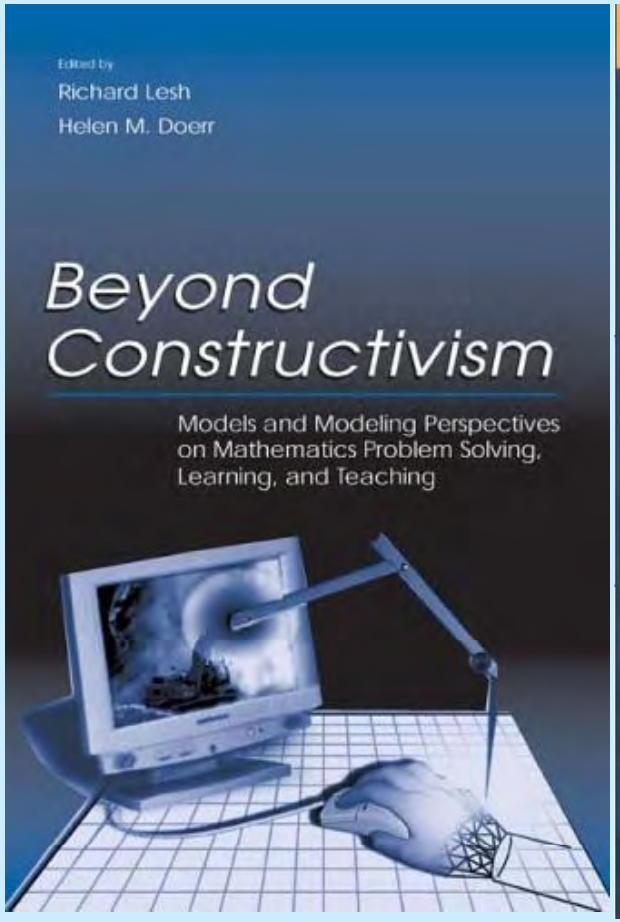
# Testing is not manufacturing QC

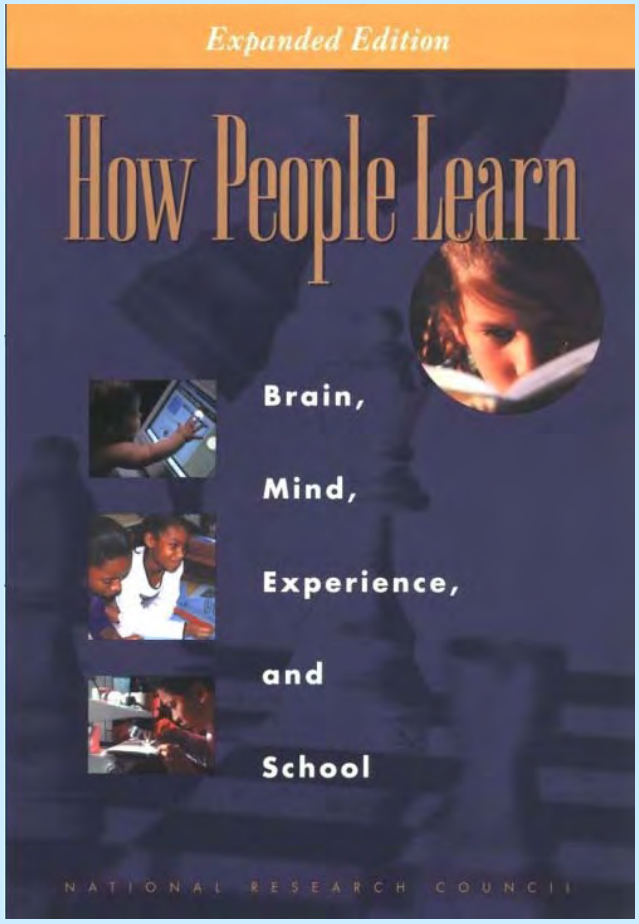
**Software testing is more like design evaluation than manufacturing quality control.**

- A **manufacturing defect** appears in an individual instance of a product (like badly wired brakes in a car). It makes sense to look at every instance in the same ways (regression tests) because any one might fail in a given way, even if the one before and the one after did not.
- A **design defect** appears in every instance of the product. The challenge of design QC is to understand the full range of implications of the design, not to look for the same problem over and over.

By the way, Six Sigma is a manufacturing quality management methodology. The "six sigmas" are six standard deviations surrounding the mean of a probability distribution. I have never heard a rationale for applying this to software. (I've seen the enthusiasm, but not the mathematics.)

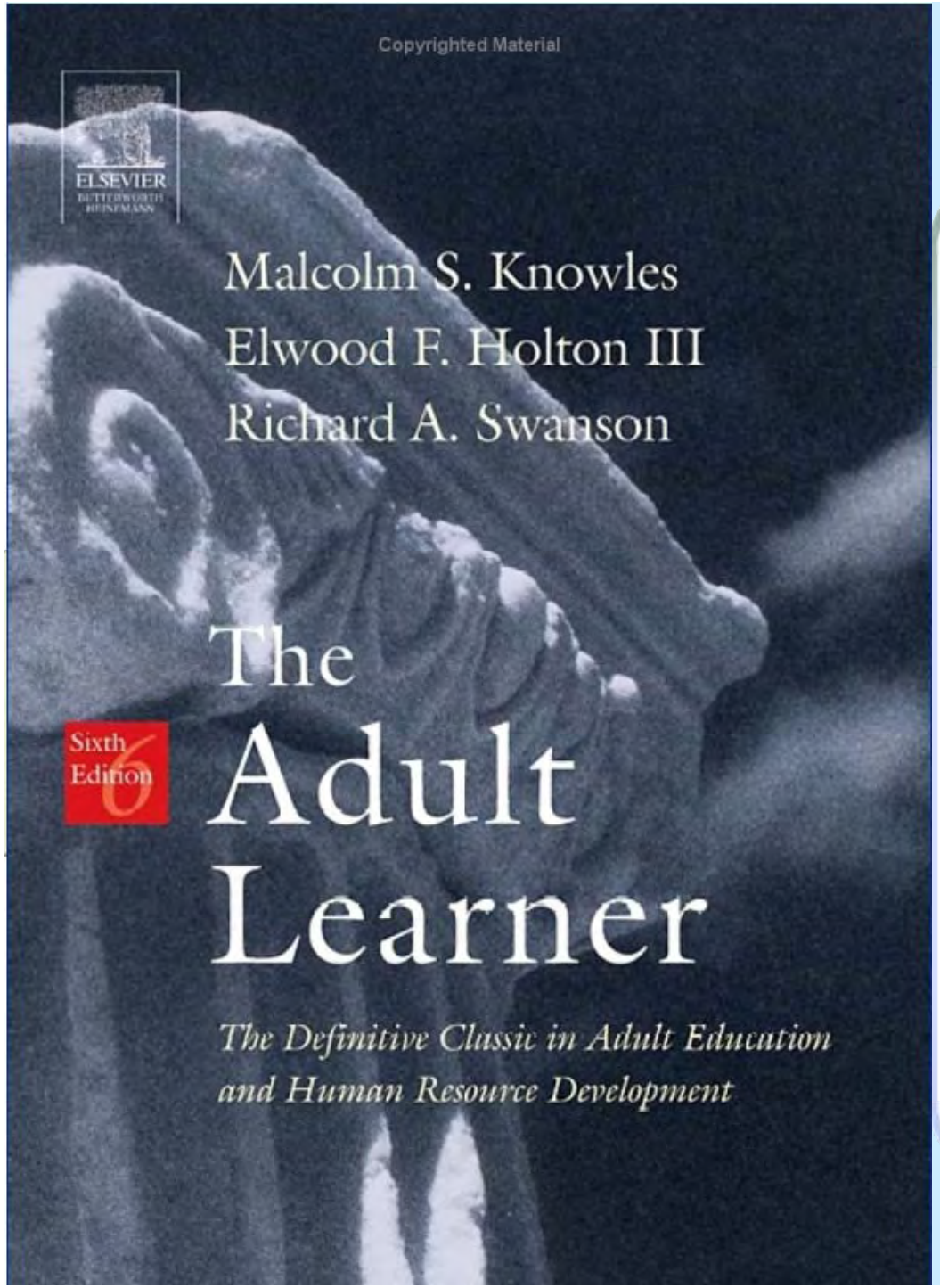
# Dealing with complexity




In science / math education, the transfer problem is driving fundamental change in the classroom

Students learn (and transfer) better when they discover concepts, rather than by being told them




# Andragogy

Pedagogy: study of teaching / learning of children

Andragogy: study of teaching / learning of adults

University undergrads are in a middle ground between the teacher-directed child and the fully-self-directed adult

Both groups, but especially adults, benefit from activity-based and discovery-based styles

# The industrial need for testing courses

- Up to ½ of the software engineering effort involves testing,
- Many companies have 1:5 to 1:1 ratios of testers to programmers
- Few universities teach testing courses
- Many of the newer courses are broad and very shallow (new IEEE/ACM curriculum guide lists 100 pounds of test-content potatoes for a less-than-3-credit sack.)
- Very few universities offer a second / third course in testing

# The Test-Related Labor Market

Lots of advice that testers should work as programmers

- Unit and API test (independent or pair with programmers)
- Write GUI regression test suites
- Write performance tests
- Write test tools
- Write test code to drive devices or other systems
- Write non-regression tests that use technology to reach beyond what humans can do manually,
  - high volume (long sequence) testing
  - high precision testing
  - high diversity (directed search) testing

To what extent should we expect this generation of testers to be programmers?

And for those who are programmers, what should we ask them to do?

# Our Labour Pool – data from 2004

- Nationally, CS enrolment is down 70% since 2001
- **90,000 new software development positions per year** (plus 29,000 support & hw positions).
- **60,000 computing B.Sc. grads**
  - (including computer engineers)
- 20,208 M.Sc. (many have B.Sc. already)
- 40,000 Associate degree (many go on to B.Sc.)
- Many of these are not from the top-ranked universities (2004 data):

| | | |
|---|---|---|
| – DeVry Institute of Tech | 3894 | BSCS graduates |
| – University of Phoenix | 2552 | |
| – American Intercontinental | 1060 | |
| – Strayer University | 993 | |

# Labour Pool

**U.S. tech job growth continues**

U.S. IT employment continues on a growth path, rising 6% from a year ago to reach 3.68 million employed, according to the most-recent Bureau of Labor Statistics employment survey. IT unemployment was 2%, according to an average of the past four quarters of BLS data, including its most recent third-quarter results. That unemployment rate is down from 2.2% in 2006 and as high as 5.6% in the third quarter of 2003. The total IT workforce, employed and unemployed, also grew about 6% from a year ago. The unemployment rate in management and professional jobs overall was also 2.0%. The biggest job growth categories continue to be software engineers, computer scientists and systems analysts, and IS managers. Software engineers, the largest category, grew 8% from a year ago and make up a quarter of all IT jobs. *(InformationWeek 10/17/07)*

# Our Labour Pool #2

- My understanding is since 2004:
  - open jobs have increased, while
  - CS enrolment has continued to significantly decrease.
- We appear to have touched bottom and might grow back significantly, but even if enrolment doubles in academic 2008-2009, those folks won't graduate until 2012.

# Our Labour Pool #3

A CS degree is no guarantee of programming capability. I've visited schools around the country over the past two years.

- Several schools emphasize theory over programming skill (a senior professor at one school told me, "Few of our students can write a working100-line program when they graduate"). This is also widely perceived as a problem common to many CS graduates from India.
- Few CS or Software Engineering programmings emphasize (or even expose students) to soft skills (interviewing, context assessment, usability-oriented design, role playing, persuasive speaking and writing).
- Many courses in design and requirements analysis are essentially tutorials in patterns, UML, and creation of massive template-driven documentation.
- Many courses in software testing are broad and superficial.
- Another block of entrants into the field come from business schools, but many graduates with degrees in "Information Systems" have minimal education in software development or assessment.

# Our Labour Pool #4

What I think this means…

- Of technically proficient graduates interested in testing, most seem to go to big publishers (Microsoft, Google) who aggressively recruit them.
- The IT community is unlikely to meet its needs for new testers with university graduate computer science majors who can write adequate code.

**Over the next 5 years, few companies' new-hire testers will be appropriate for test-first programming, glass-box testing or serious test automation.**

# Labor Pool #5

- Will continue to include large portion of manual testers who have weak backgrounds in computing
  - 40,000 recent certifications by ISTQB
- The question will be how to hire and train the best people for a combination of:
  - Manual testing positions
  - GUI automation positions
  - Non-GUI (e.g. toolbuilder or HVAT) automation positions
  - Glass-box testing and test-first programming positions
- The proportions might shift over time, but the four roles (and in some companies, several other test-group roles) will continue.

# Advice I give to employers on who to hire

For much of the past 30 years, many leaders in the testing community have urged us to dumb our work down, make it more routine and then cost-reduce it.

In my view, this often leads to serious inefficiency and weak testing.

Rather than

bringing testing down to a level that weak testers can do it (albeit it, weakly),

I think we should

Hire people with strong potential, and train them to do strong work.

# Test groups should offer diverse, collaborating specialists

Test groups need people who understand

- the application under test,
- the technical environment in which it will run (and the associated risks),
- the market (and their expectations, demands, and support needs),
- the architecture and mechanics of tools to support the testing effort,
- and the underlying implementation of the code.

You cannot find all this in any one person. You can build a group of strikingly different people, encourage them to collaborate and cross-train, and assign them to project areas that need what they know.

http://www.kaner.com/pdfs/JobsRev6.pdf

**These people need test-related education / training**

# DETAILS ABOUT THE COURSE

# A course tour, on Moodle

www.moodle.org

Free

Course management system

Useful for:

- Live short courses (requires web access)
- Live academic courses (long term, homework)
- Hybrid of remote / live
- Remote courses
  - Synchronous (live web conference tools are better)
  - Asynchronous

Moodle - A Free, Open Source Course Management System for Online Learning - Minefield

File Edit View History Bookmarks Tools Help

http://moodle.org/

moodle

Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox

moodle

You are not logged in. (Login)

English (en) Go

## Main Menu

- **Free Support**
- Documentation
- Issue Tracker
- Donations
- Download Moodle
- Modules and plugins
- Themes
- Moodle Buzz
- Moodle Sites
- Moodle Statistics
- Moodle News

## Latest News

*17 Sep, 09:08*
Martin Dougiamas
Moodle 1.6.2 is available for download more...

*20 Jun, 03:39*
Martin Dougiamas
Moodle 1.6 goes gold!

## Welcome to Moodle!

Moodle is a course management system (CMS) - a free, Open Source software package designed using sound pedagogical principles, to help educators create effective online learning communities. You can download and use it on any computer you have handy (including webhosts), yet it can scale from a single-teacher site to a 50,000-student University. This site itself is created using Moodle, so check out the Moodle Demonstration Courses or read the latest Moodle Buzz.

## Moodle Community

Moodle has a large and diverse user community with over 130,000 registered users on this site alone, speaking over 75 languages in over 160 countries (we have more statistics here). The best place to start is Using Moodle, which is where the main international discussions are held in English, but we have a variety of groups discussing other topics and in other languages.

## Search

Search this site

## Login

Username:

Password:

Login

Create new account
Lost password?

Official Moodle Partner:

moodle rooms

hosting, integrations, consulting, support, installations, training

Scripts Currently Forbidden [<script>: 12] [J+F+P: 0]

Options...

Waiting for www.w3.org...

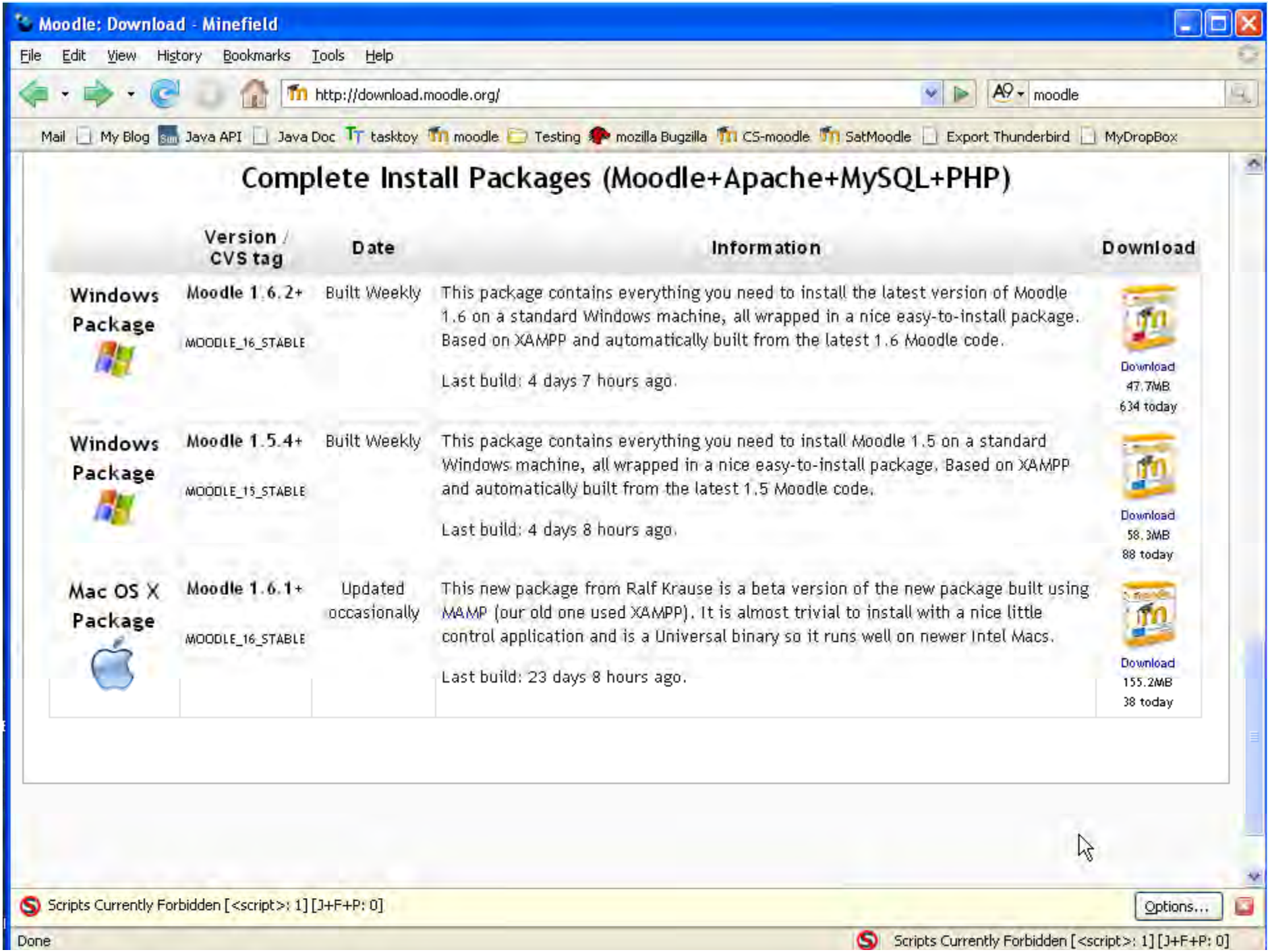
Scripts Currently Forbidden [<script>: 12] [J+F+P: 0]

# Moodle platforms

Windows

Mac

Linux


Moodle: Download - Minefield
File Edit View History Bookmarks Tools Help
http://download.moodle.org/
moodle
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Complete Install Packages (Moodle+Apache+MySQL+PHP)
Version / CVS tag
Date
Information
Download
Windows Package
Moodle 1.6.2+
MOODLE_16_STABLE
Built Weekly
This package contains everything you need to install the latest version of Moodle 1.6 on a standard Windows machine, all wrapped in a nice easy-to-install package. Based on XAMPP and automatically built from the latest 1.6 Moodle code.
Last build: 4 days 7 hours ago.
Download
47.7MB
634 today
Windows Package
Moodle 1.5.4+
MOODLE_15_STABLE
Built Weekly
This package contains everything you need to install Moodle 1.5 on a standard Windows machine, all wrapped in a nice easy-to-install package. Based on XAMPP and automatically built from the latest 1.5 Moodle code.
Last build: 4 days 8 hours ago.
Download
58.3MB
88 today
Mac OS X Package
Moodle 1.6.1+
MOODLE_16_STABLE
Updated occasionally
This new package from Ralf Krause is a beta version of the new package built using MAMP (our old one used XAMPP). It is almost trivial to install with a nice little control application and is a Universal binary so it runs well on newer Intel Macs.
Last build: 23 days 8 hours ago.
Download
155.2MB
38 today
Scripts Currently Forbidden [<script>: 1] [J+F+P: 0]
Options...
Done
Scripts Currently Forbidden [<script>: 1] [J+F+P: 0]

# Overview of Moodle

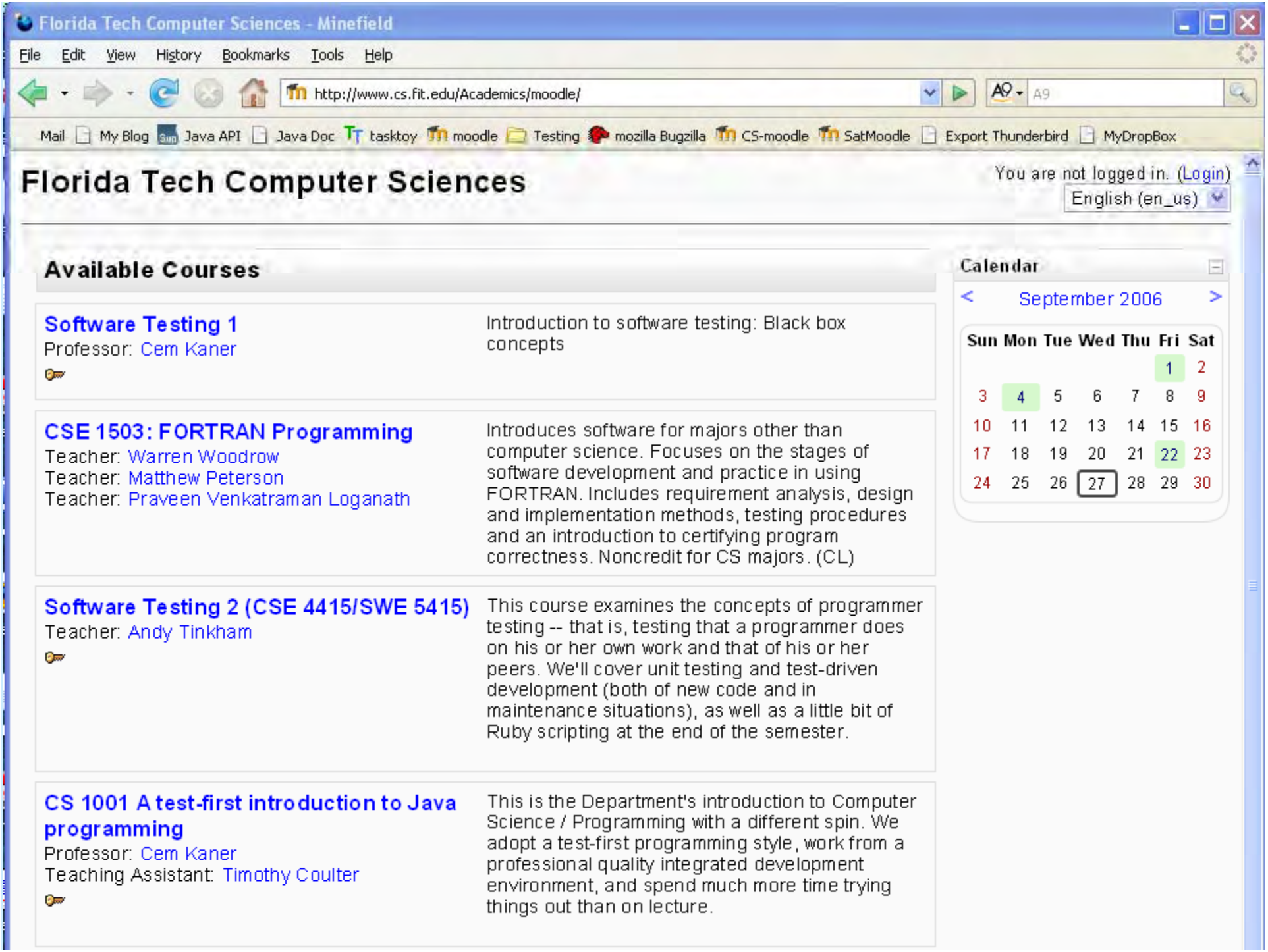
The following slides illustrate what we can do with course management systems, using Moodle as the specific example.

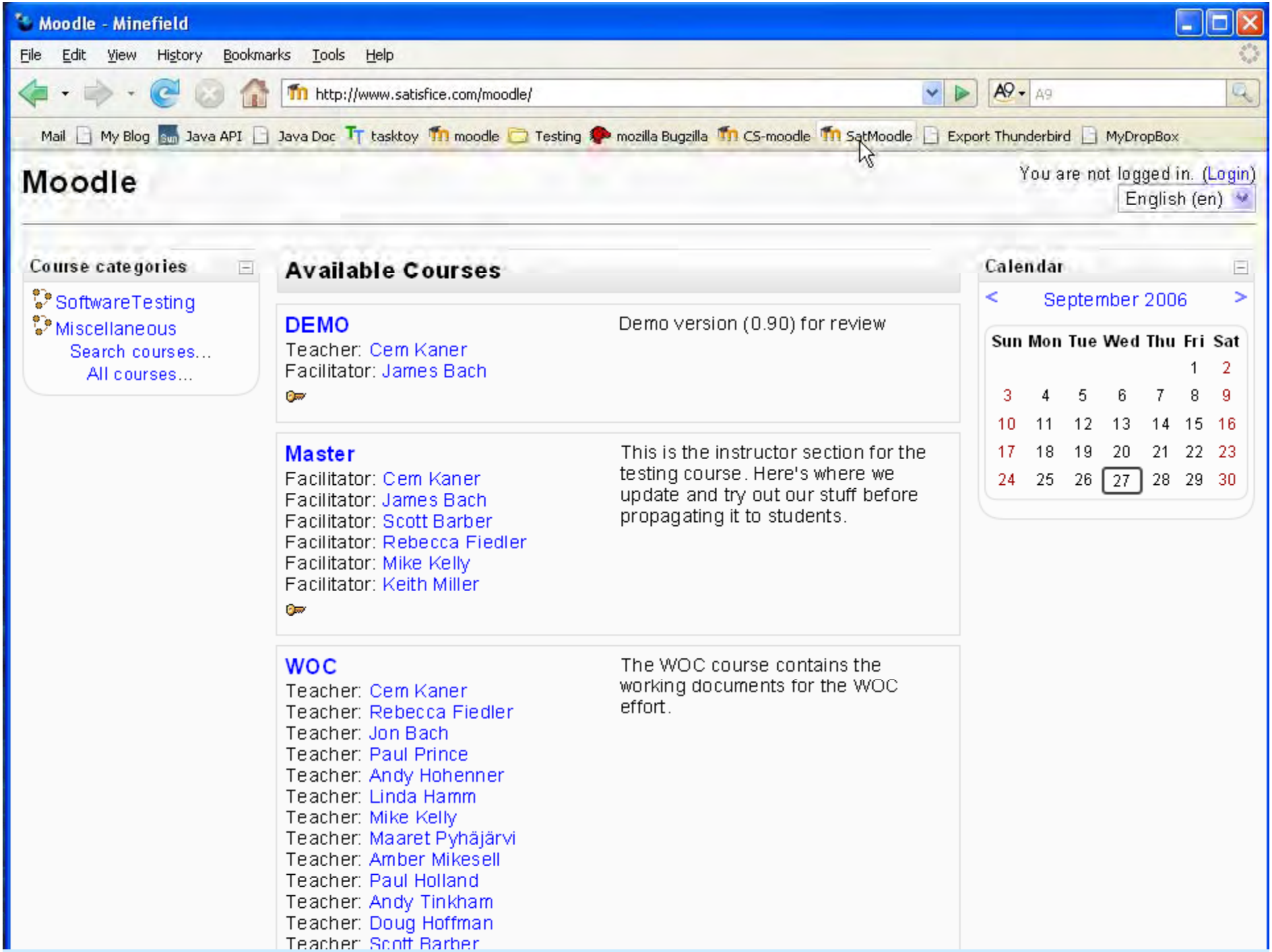
If most / all of you are familiar with course management systems, I'll skip to the next section.

==========

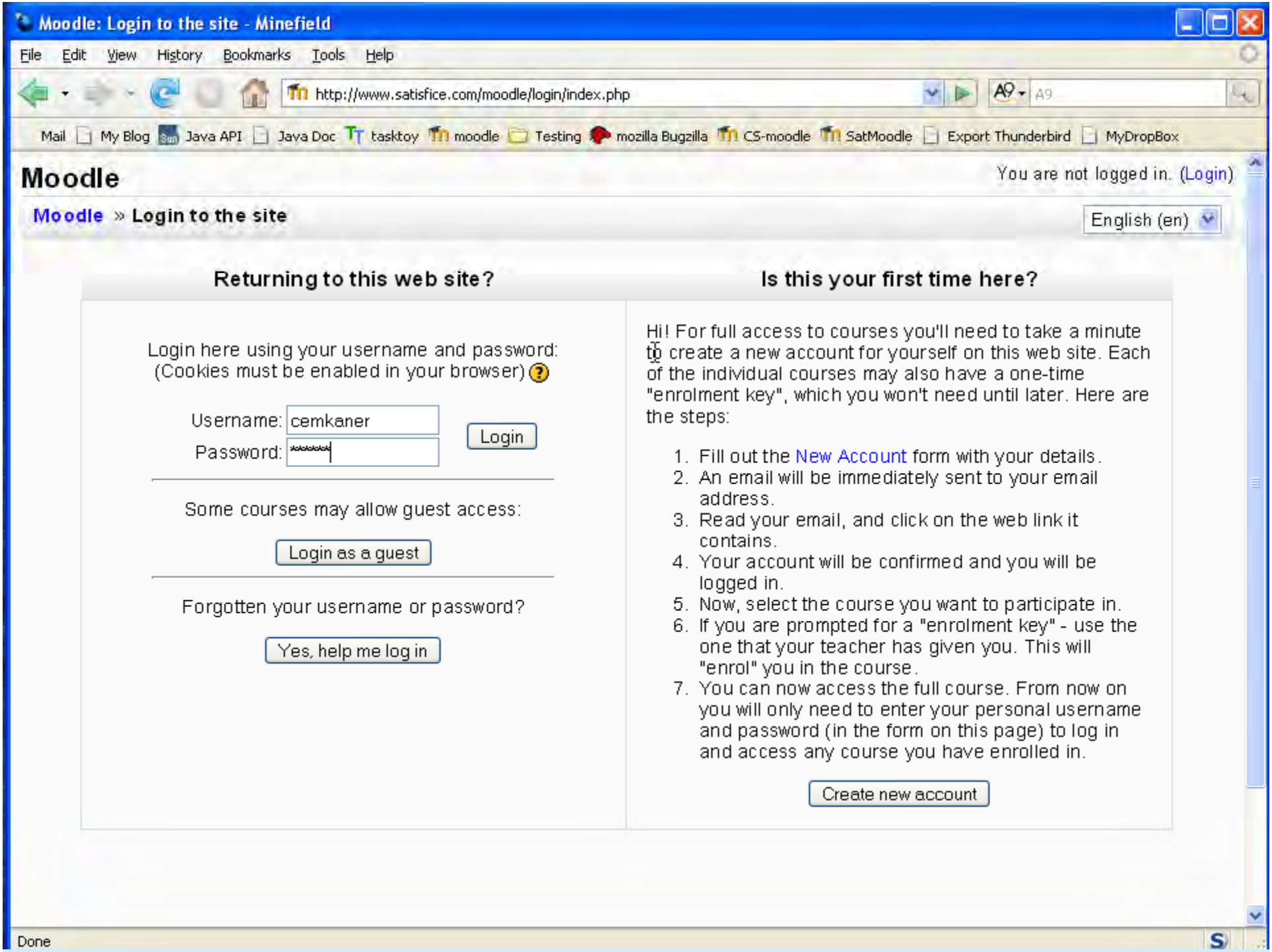
These screen shots are samples from some courses / activities that I host on Moodle

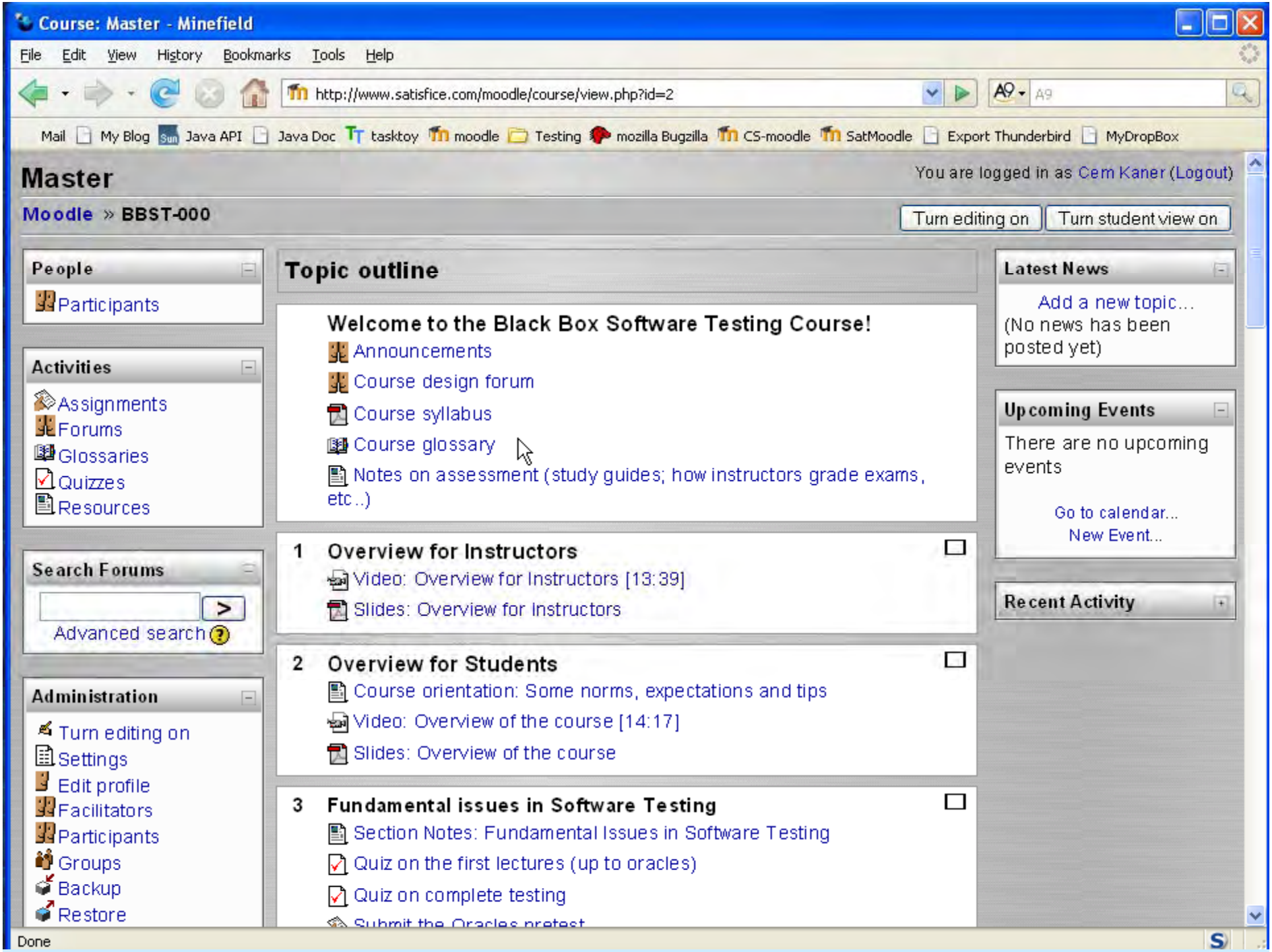
Some data / demonstrations are unavailable (e.g. layout of quiz results) because of student confidentiality rules

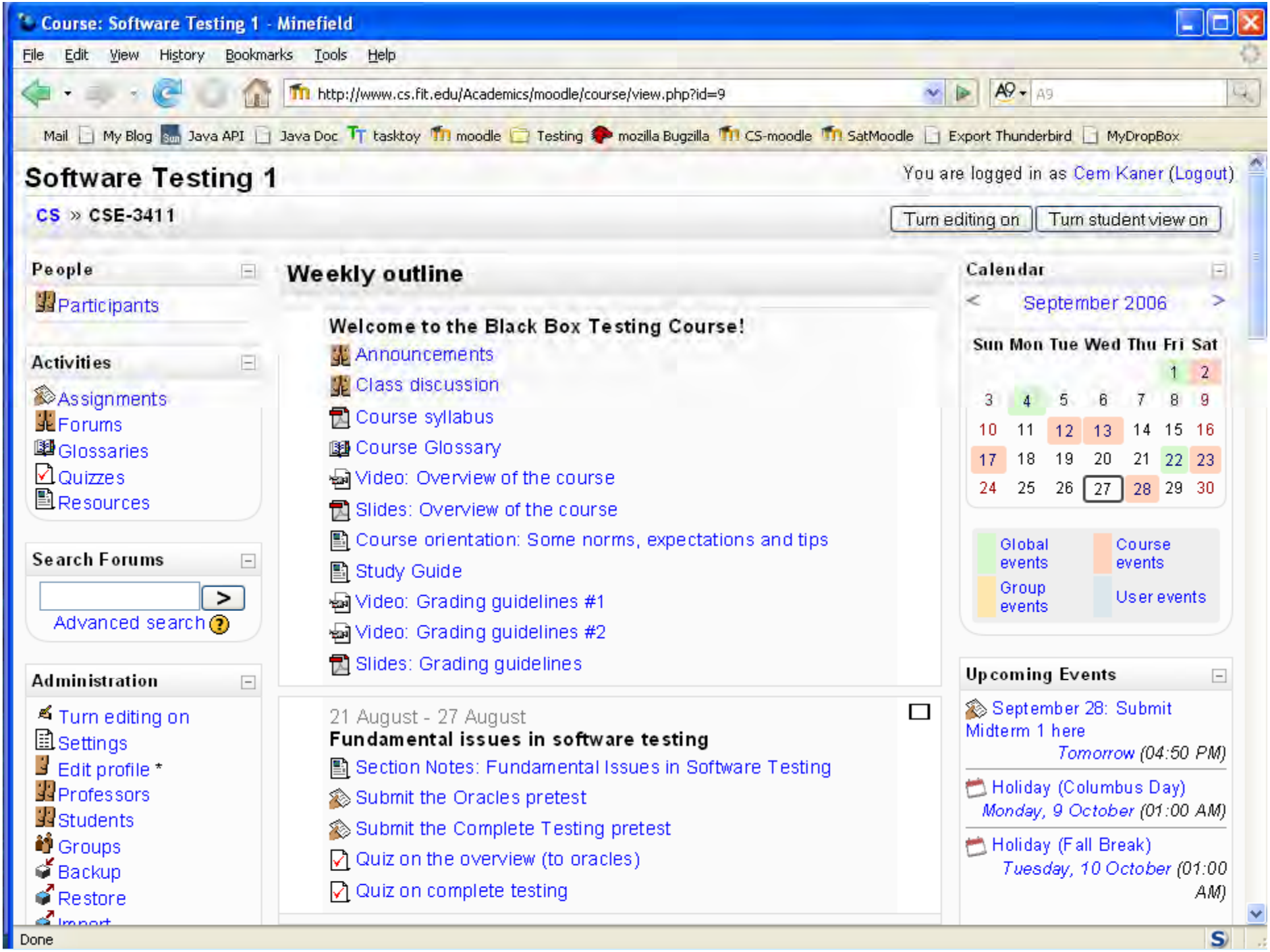

Florida Tech Computer Sciences - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Florida Tech Computer Sciences
You are not logged in. (Login)
English (en_us)
Available Courses
Software Testing 1
Professor: Cem Kaner
Introduction to software testing: Black box concepts
CSE 1503: FORTRAN Programming
Teacher: Warren Woodrow
Teacher: Matthew Peterson
Teacher: Praveen Venkatraman Loganath
Introduces software for majors other than computer science. Focuses on the stages of software development and practice in using FORTRAN. Includes requirement analysis, design and implementation methods, testing procedures and an introduction to certifying program correctness. Noncredit for CS majors. (CL)
Software Testing 2 (CSE 4415/SWE 5415)
Teacher: Andy Tinkham
This course examines the concepts of programmer testing -- that is, testing that a programmer does on his or her own work and that of his or her peers. We'll cover unit testing and test-driven development (both of new code and in maintenance situations), as well as a little bit of Ruby scripting at the end of the semester.
CS 1001 A test-first introduction to Java programming
Professor: Cem Kaner
Teaching Assistant: Timothy Coulter
This is the Department's introduction to Computer Science / Programming with a different spin. We adopt a test-first programming style, work from a professional quality integrated development environment, and spend much more time trying things out than on lecture.
Calendar
< September 2006 >
Sun Mon Tue Wed Thu Fri Sat
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

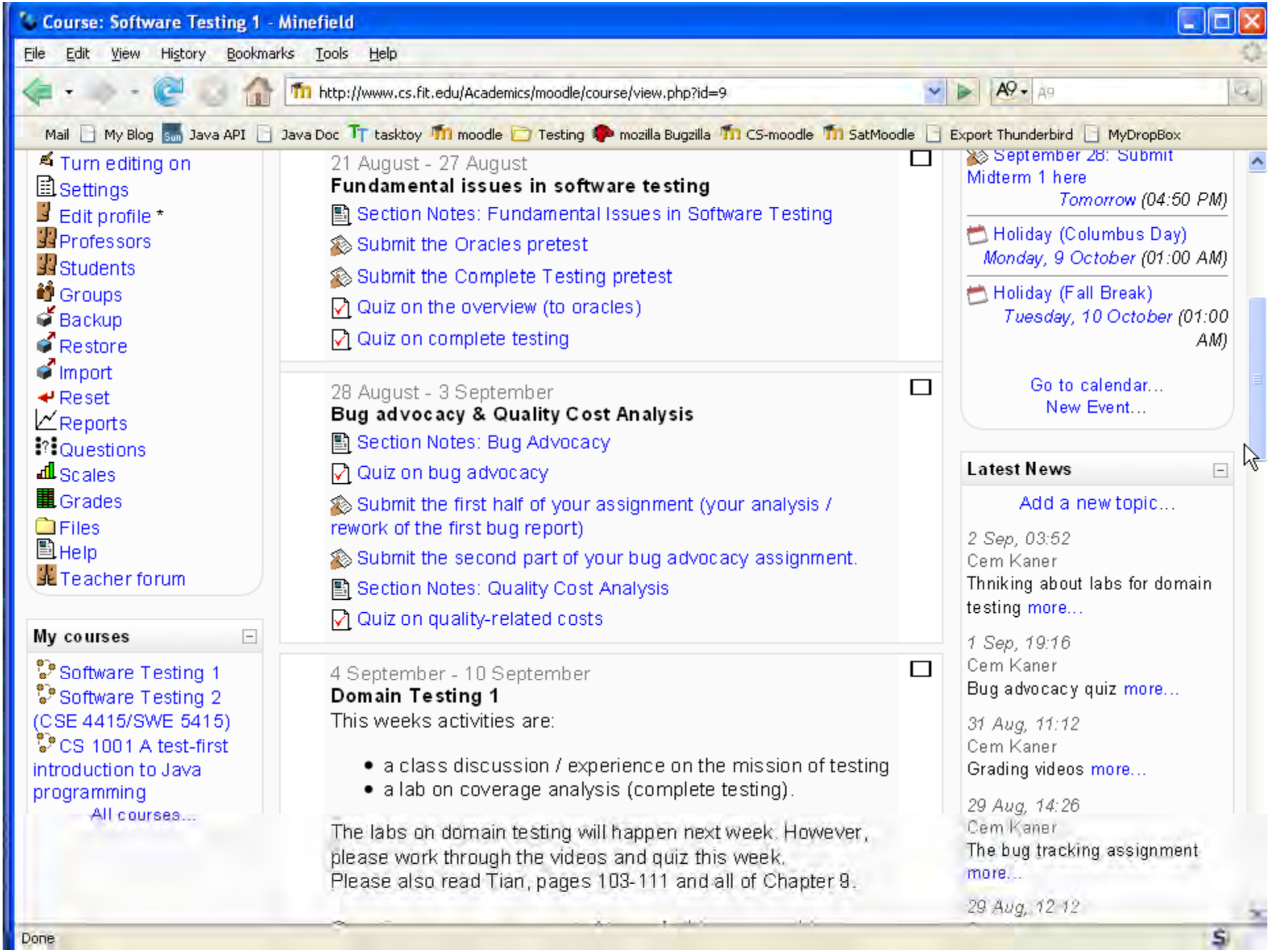

Moodle - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Moodle
You are not logged in. (Login)
English (en)
Course categories
SoftwareTesting
Miscellaneous
Search courses...
All courses...
Available Courses
DEMO
Teacher: Cem Kaner
Facilitator: James Bach
Demo version (0.90) for review
Master
Facilitator: Cem Kaner
Facilitator: James Bach
Facilitator: Scott Barber
Facilitator: Rebecca Fiedler
Facilitator: Mike Kelly
Facilitator: Keith Miller
This is the instructor section for the testing course. Here's where we update and try out our stuff before propagating it to students.
WOC
Teacher: Cem Kaner
Teacher: Rebecca Fiedler
Teacher: Jon Bach
Teacher: Paul Prince
Teacher: Andy Hohenner
Teacher: Linda Hamm
Teacher: Mike Kelly
Teacher: Maaret Pyhäjärvi
Teacher: Amber Mikesell
Teacher: Paul Holland
Teacher: Andy Tinkham
Teacher: Doug Hoffman
Teacher: Scott Barber
The WOC course contains the working documents for the WOC effort.
Calendar
< September 2006 >
Sun Mon Tue Wed Thu Fri Sat
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

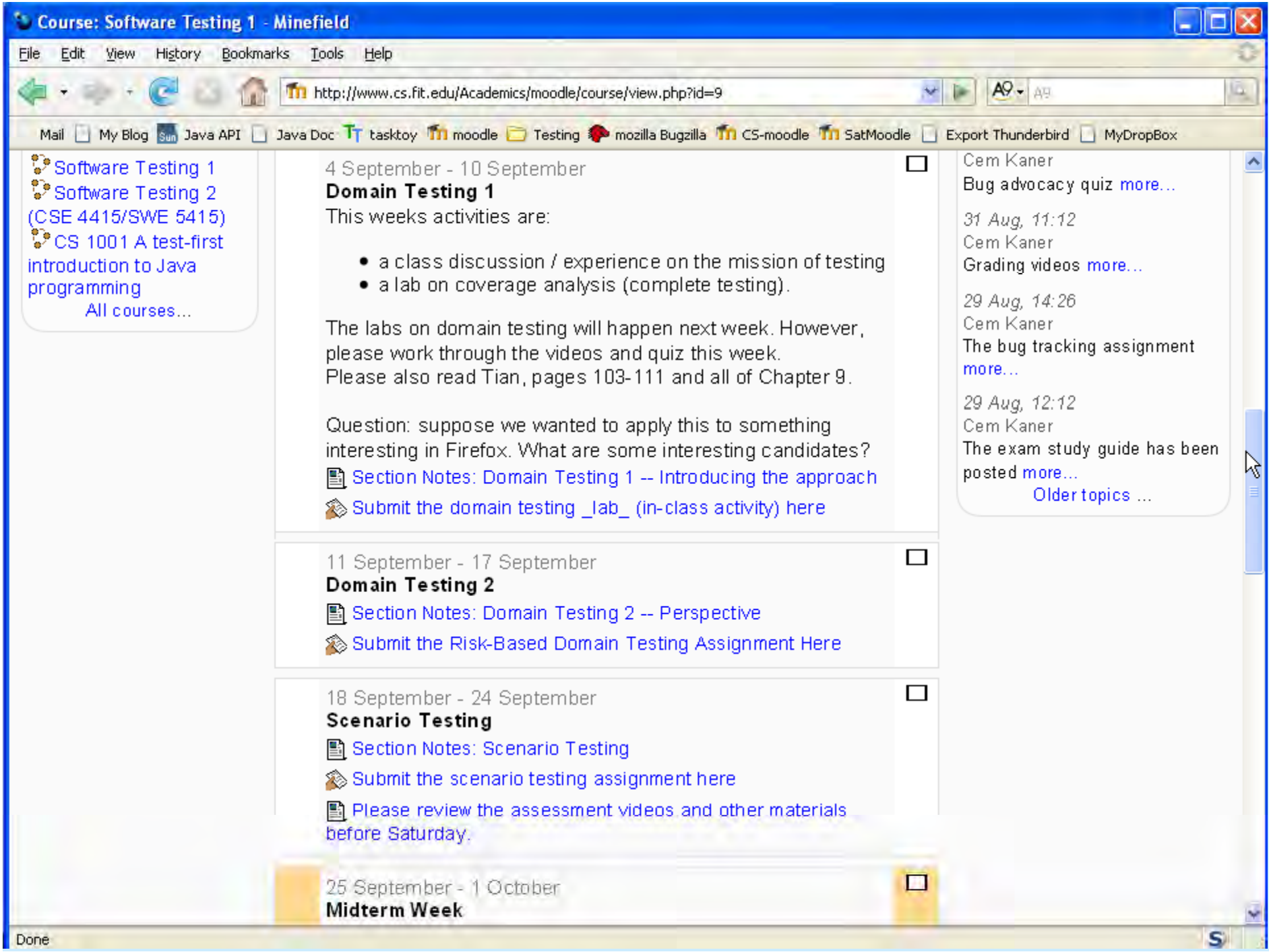

A Course in Black Box Testing - Minefield
File Edit View History Bookmarks Tools Help
http://www.testingeducation.org/BBST/
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Center for Software Testing Education & Research
Home
Support the Lab
Call for Donations
Submit Materials
Content
Course Notes
Lab Articles
Our Workshops
Other Courses
About the Lab
Members
Active Students
Acknowledgements
Lab Policies
Find
www CSTER
Black Box Software Testing: By Cem Kaner & James Bach
We're setting up a mailing list for a low-traffic, moderated discussion of how to teach or self-study this course. If you're interested, sign up here. We will NOT share addresses with third parties or send commercial advertising.
These are our course materials, in our suggested sequence. Each section includes a video lecture, slides, and additional types of learning aids. Click on the topic name to see its set of materials.
Course Syllabus -- Fall 2005
Fall 2005 -- Exam review question set [PDF]
Overview of the course [for students] Slides [PPT]
Overview of the course [for teachers]
1. Introduction: The strategy problem and the oracle problem
2. Introduction 2: The impossibility of complete testing and the measurement problem
3. Bug advocacy: How to win friends, influence programmers, and stomp bugs
4. Quality cost analysis
5. More on bug advocacy, your credibility, and the mission of the tracking system.
6. Testing techniques: Domain testing
7. Testing techniques: Scenario testing

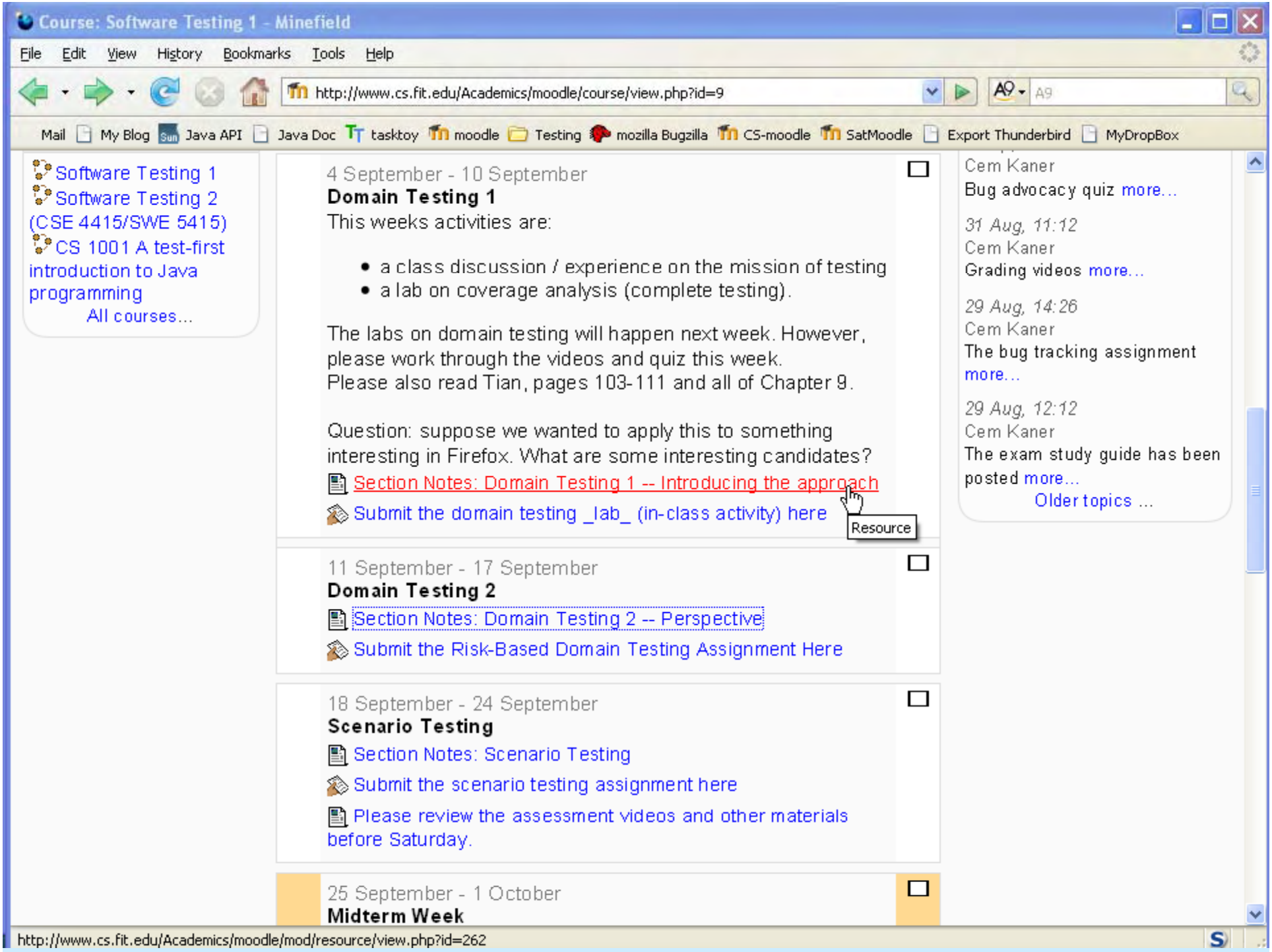

Moodle: Login to the site - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/login/index.php
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Moodle
You are not logged in. (Login)
Moodle » Login to the site
English (en)
Returning to this web site?
Login here using your username and password:
(Cookies must be enabled in your browser)
Username: cemkaner
Password: *******
Login
Some courses may allow guest access:
Login as a guest
Forgotten your username or password?
Yes, help me log in
Is this your first time here?
Hi! For full access to courses you'll need to take a minute to create a new account for yourself on this web site. Each of the individual courses may also have a one-time "enrolment key", which you won't need until later. Here are the steps:
1. Fill out the New Account form with your details.
2. An email will be immediately sent to your email address.
3. Read your email, and click on the web link it contains.
4. Your account will be confirmed and you will be logged in.
5. Now, select the course you want to participate in.
6. If you are prompted for a "enrolment key" - use the one that your teacher has given you. This will "enrol" you in the course.
7. You can now access the full course. From now on you will only need to enter your personal username and password (in the form on this page) to log in and access any course you have enrolled in.
Create new account
Done

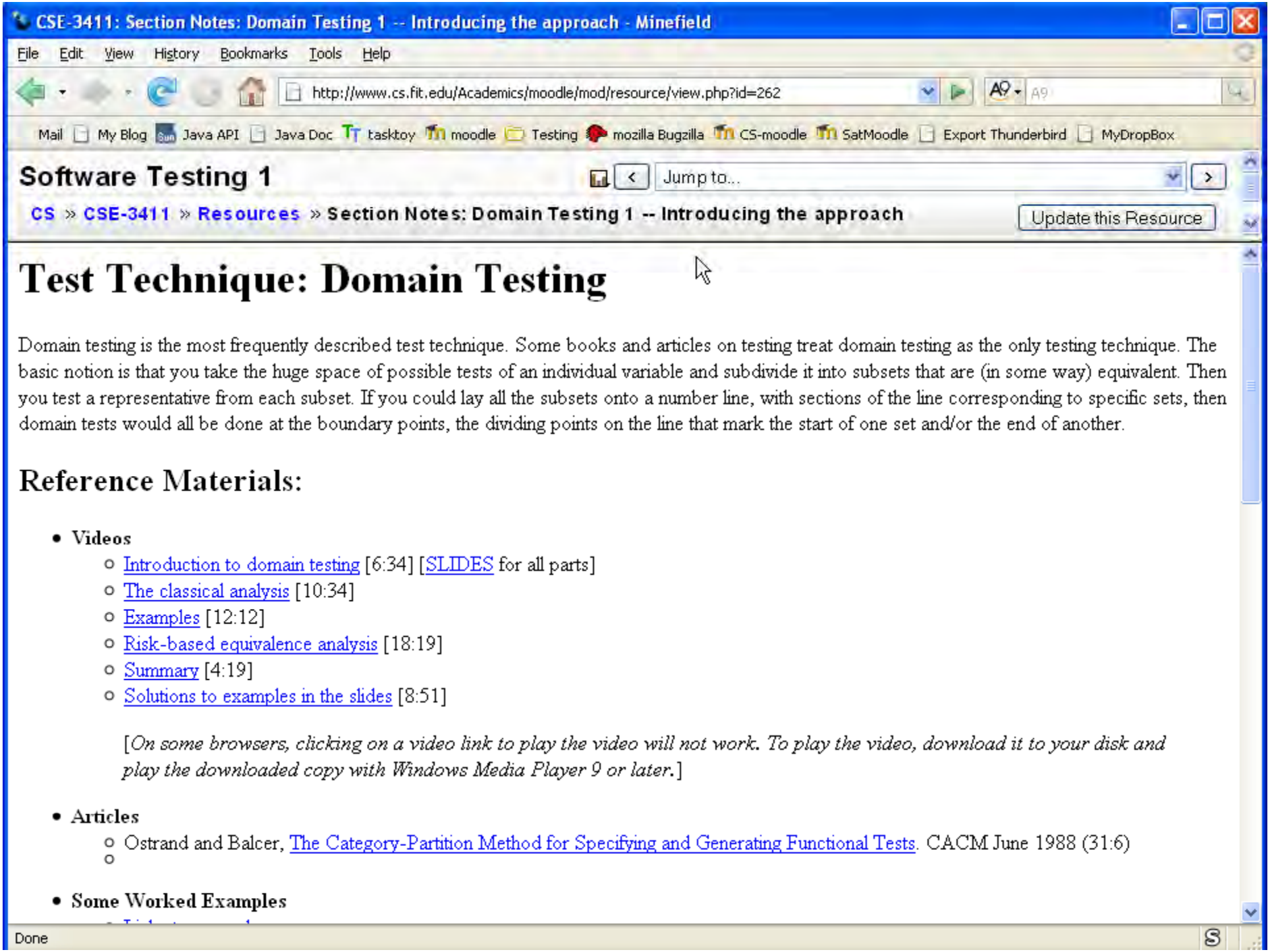

Course: Master - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=2
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000
Turn editing on
Turn student view on
People
Participants
Activities
Assignments
Forums
Glossaries
Quizzes
Resources
Search Forums
Advanced search
Administration
Turn editing on
Settings
Edit profile
Facilitators
Participants
Groups
Backup
Restore
Topic outline
Welcome to the Black Box Software Testing Course!
Announcements
Course design forum
Course syllabus
Course glossary
Notes on assessment (study guides; how instructors grade exams, etc..)
1 Overview for Instructors
Video: Overview for Instructors [13:39]
Slides: Overview for Instructors
2 Overview for Students
Course orientation: Some norms, expectations and tips
Video: Overview of the course [14:17]
Slides: Overview of the course
3 Fundamental issues in Software Testing
Section Notes: Fundamental Issues in Software Testing
Quiz on the first lectures (up to oracles)
Quiz on complete testing
Latest News
Add a new topic...
(No news has been posted yet)
Upcoming Events
There are no upcoming events
Go to calendar...
New Event...
Recent Activity
Done

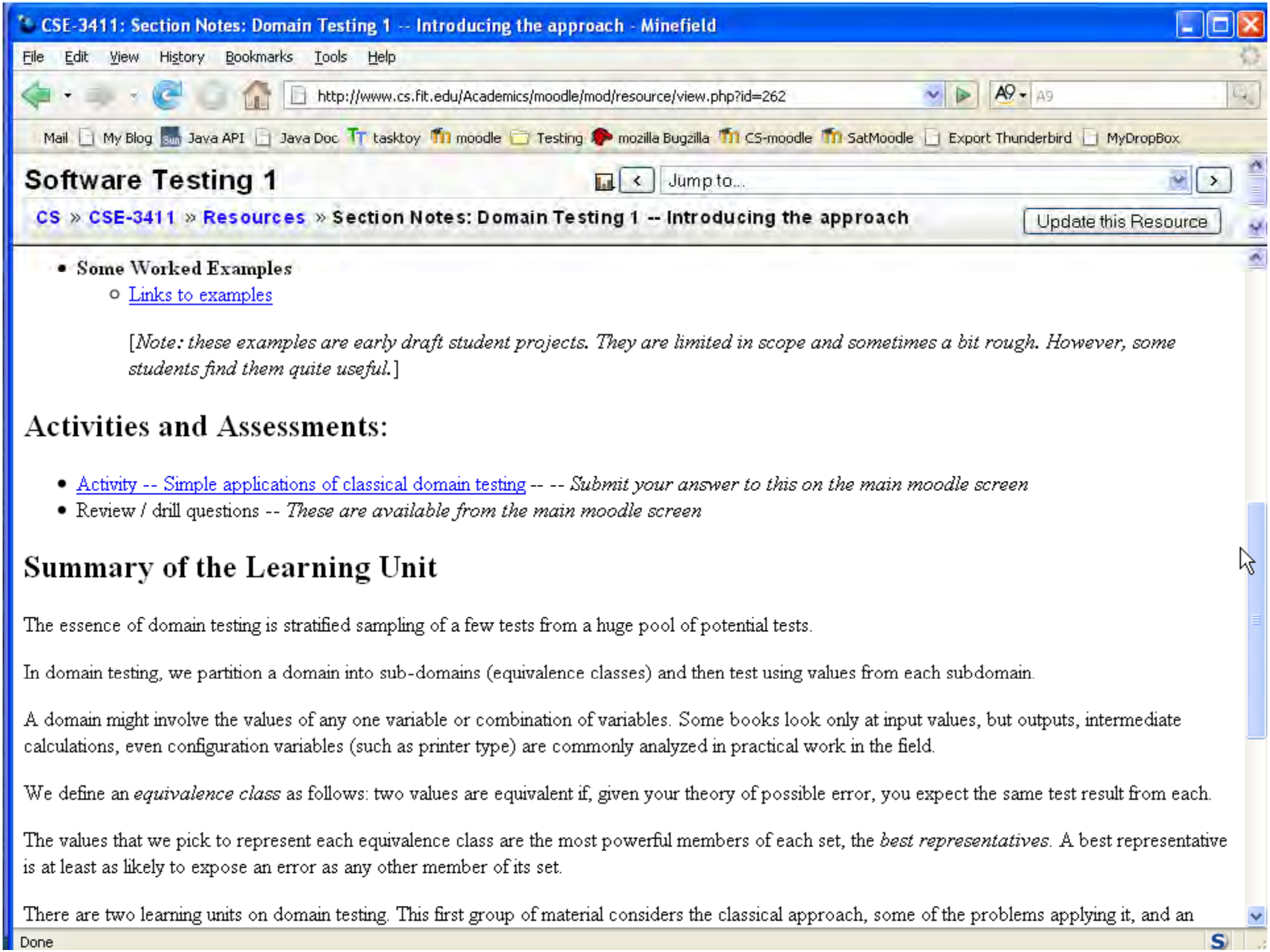

Course: Software Testing 1 - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/course/view.php?id=9
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Software Testing 1
You are logged in as Cem Kaner (Logout)
CS » CSE-3411
Turn editing on
Turn student view on
People
Participants
Activities
Assignments
Forums
Glossaries
Quizzes
Resources
Search Forums
Advanced search
Administration
Turn editing on
Settings
Edit profile *
Professors
Students
Groups
Backup
Restore
Weekly outline
Welcome to the Black Box Testing Course!
Announcements
Class discussion
Course syllabus
Course Glossary
Video: Overview of the course
Slides: Overview of the course
Course orientation: Some norms, expectations and tips
Study Guide
Video: Grading guidelines #1
Video: Grading guidelines #2
Slides: Grading guidelines
21 August - 27 August
Fundamental issues in software testing
Section Notes: Fundamental Issues in Software Testing
Submit the Oracles pretest
Submit the Complete Testing pretest
Quiz on the overview (to oracles)
Quiz on complete testing
Calendar
< September 2006 >
Sun Mon Tue Wed Thu Fri Sat
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
Global events
Course events
Group events
User events
Upcoming Events
September 28: Submit Midterm 1 here
Tomorrow (04:50 PM)
Holiday (Columbus Day)
Monday, 9 October (01:00 AM)
Holiday (Fall Break)
Tuesday, 10 October (01:00 AM)
Done

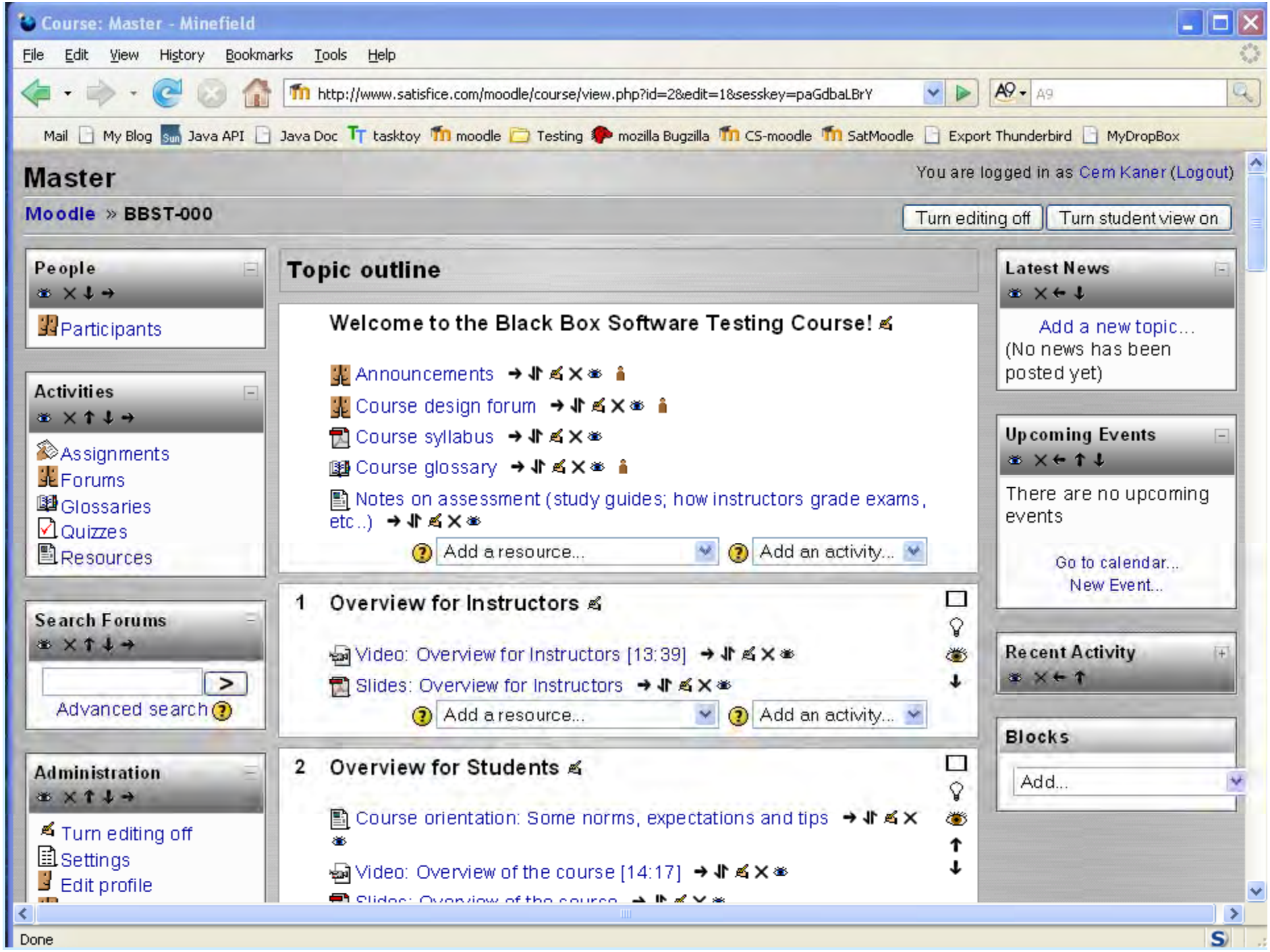

Course: Software Testing 1 - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/course/view.php?id=9
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Turn editing on
Settings
Edit profile *
Professors
Students
Groups
Backup
Restore
Import
Reset
Reports
Questions
Scales
Grades
Files
Help
Teacher forum
My courses
Software Testing 1
Software Testing 2 (CSE 4415/SWE 5415)
CS 1001 A test-first introduction to Java programming
All courses...
21 August - 27 August
Fundamental issues in software testing
Section Notes: Fundamental Issues in Software Testing
Submit the Oracles pretest
Submit the Complete Testing pretest
Quiz on the overview (to oracles)
Quiz on complete testing
28 August - 3 September
Bug advocacy & Quality Cost Analysis
Section Notes: Bug Advocacy
Quiz on bug advocacy
Submit the first half of your assignment (your analysis / rework of the first bug report)
Submit the second part of your bug advocacy assignment.
Section Notes: Quality Cost Analysis
Quiz on quality-related costs
4 September - 10 September
Domain Testing 1
This weeks activities are:
a class discussion / experience on the mission of testing
a lab on coverage analysis (complete testing).
The labs on domain testing will happen next week. However, please work through the videos and quiz this week.
Please also read Tian, pages 103-111 and all of Chapter 9.
September 28: Submit Midterm 1 here
Tomorrow (04:50 PM)
Holiday (Columbus Day)
Monday, 9 October (01:00 AM)
Holiday (Fall Break)
Tuesday, 10 October (01:00 AM)
Go to calendar...
New Event...
Latest News
Add a new topic...
2 Sep, 03:52
Cem Kaner
Thniking about labs for domain testing more...
1 Sep, 19:16
Cem Kaner
Bug advocacy quiz more...
31 Aug, 11:12
Cem Kaner
Grading videos more...
29 Aug, 14:26
Cem Kaner
The bug tracking assignment more...
29 Aug, 12:12
Done

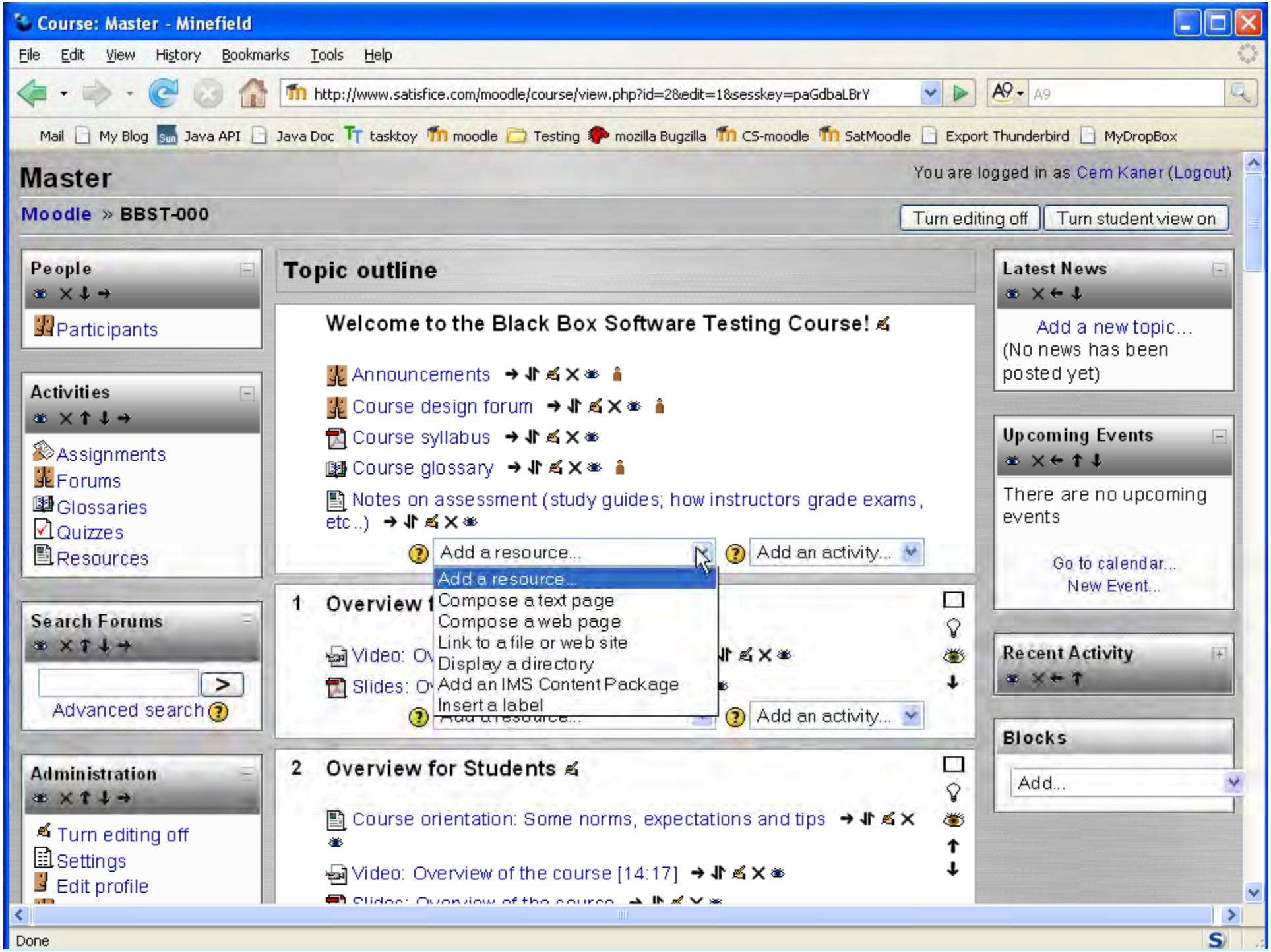

Course: Software Testing 1 - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/course/view.php?id=9
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Software Testing 1
Software Testing 2 (CSE 4415/SWE 5415)
CS 1001 A test-first introduction to Java programming
All courses...
4 September - 10 September
Domain Testing 1
This weeks activities are:
a class discussion / experience on the mission of testing
a lab on coverage analysis (complete testing).
The labs on domain testing will happen next week. However, please work through the videos and quiz this week.
Please also read Tian, pages 103-111 and all of Chapter 9.
Question: suppose we wanted to apply this to something interesting in Firefox. What are some interesting candidates?
Section Notes: Domain Testing 1 -- Introducing the approach
Submit the domain testing _lab_ (in-class activity) here
11 September - 17 September
Domain Testing 2
Section Notes: Domain Testing 2 -- Perspective
Submit the Risk-Based Domain Testing Assignment Here
18 September - 24 September
Scenario Testing
Section Notes: Scenario Testing
Submit the scenario testing assignment here
Please review the assessment videos and other materials before Saturday.
25 September - 1 October
Midterm Week
Cem Kaner
Bug advocacy quiz more...
31 Aug, 11:12
Cem Kaner
Grading videos more...
29 Aug, 14:26
Cem Kaner
The bug tracking assignment more...
29 Aug, 12:12
Cem Kaner
The exam study guide has been posted more...
Older topics ...
Done

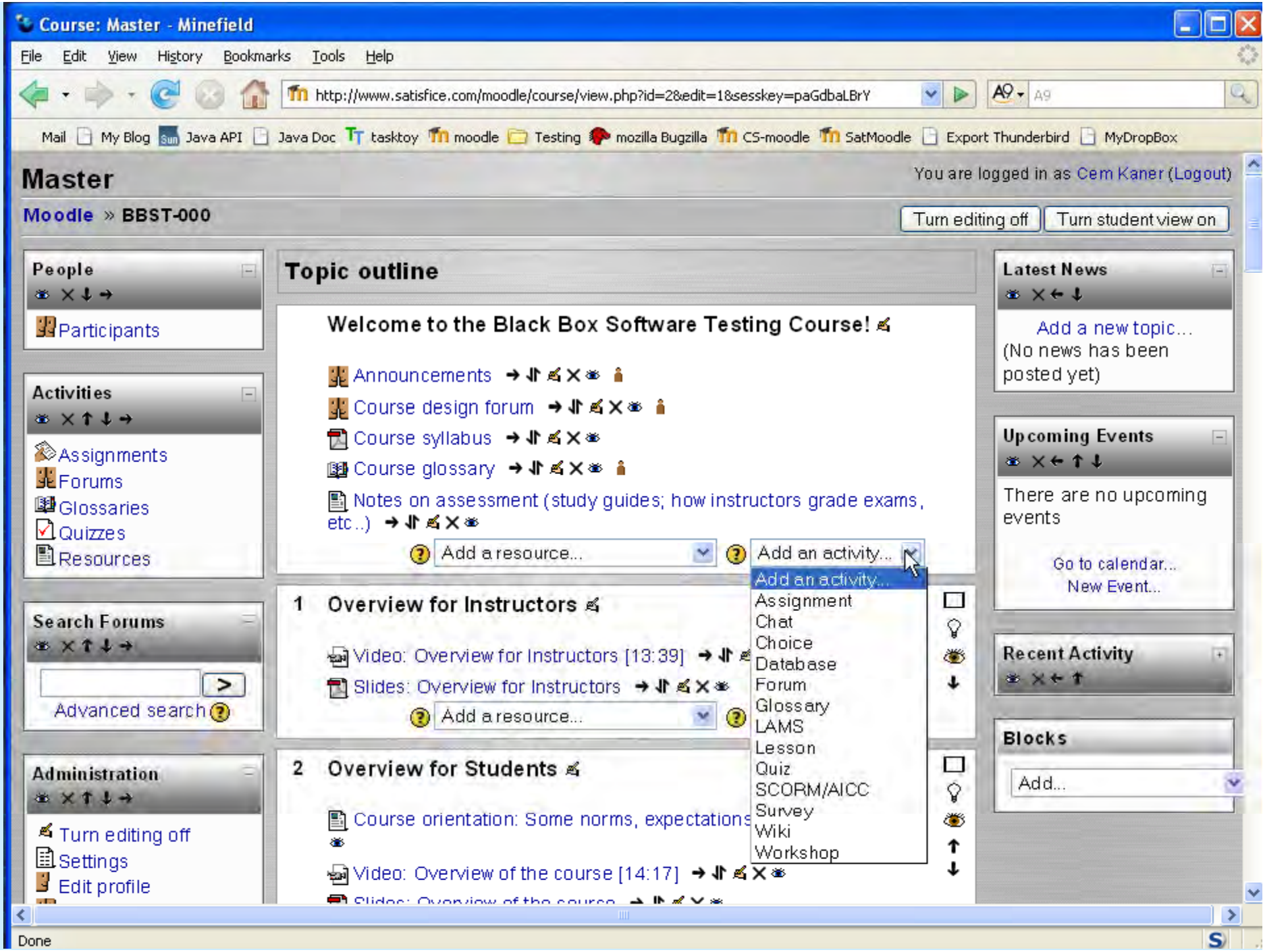

Course: Software Testing 1 - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/course/view.php?id=9
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Software Testing 1
Software Testing 2 (CSE 4415/SWE 5415)
CS 1001 A test-first introduction to Java programming
All courses...
4 September - 10 September
Domain Testing 1
This weeks activities are:
a class discussion / experience on the mission of testing
a lab on coverage analysis (complete testing).
The labs on domain testing will happen next week. However, please work through the videos and quiz this week.
Please also read Tian, pages 103-111 and all of Chapter 9.
Question: suppose we wanted to apply this to something interesting in Firefox. What are some interesting candidates?
Section Notes: Domain Testing 1 -- Introducing the approach
Submit the domain testing _lab_ (in-class activity) here
Resource
11 September - 17 September
Domain Testing 2
Section Notes: Domain Testing 2 -- Perspective
Submit the Risk-Based Domain Testing Assignment Here
18 September - 24 September
Scenario Testing
Section Notes: Scenario Testing
Submit the scenario testing assignment here
Please review the assessment videos and other materials before Saturday.
25 September - 1 October
Midterm Week
Cem Kaner
Bug advocacy quiz more...
31 Aug, 11:12
Cem Kaner
Grading videos more...
29 Aug, 14:26
Cem Kaner
The bug tracking assignment more...
29 Aug, 12:12
Cem Kaner
The exam study guide has been posted more...
Older topics ...
http://www.cs.fit.edu/Academics/moodle/mod/resource/view.php?id=262

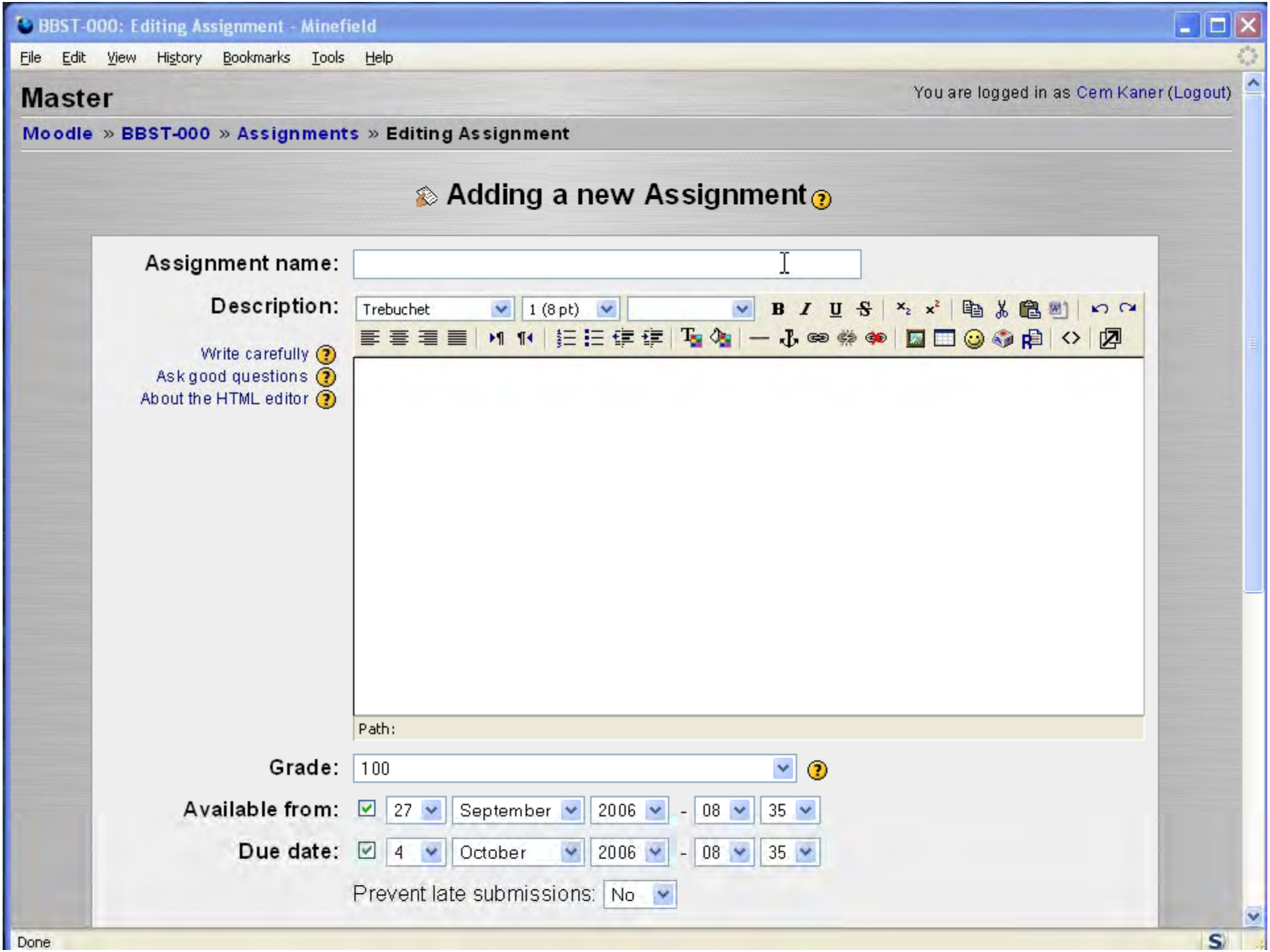

CSE-3411: Section Notes: Domain Testing 1 -- Introducing the approach - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/mod/resource/view.php?id=262
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Software Testing 1
Jump to...
CS » CSE-3411 » Resources » Section Notes: Domain Testing 1 -- Introducing the approach
Update this Resource
Test Technique: Domain Testing
Domain testing is the most frequently described test technique. Some books and articles on testing treat domain testing as the only testing technique. The basic notion is that you take the huge space of possible tests of an individual variable and subdivide it into subsets that are (in some way) equivalent. Then you test a representative from each subset. If you could lay all the subsets onto a number line, with sections of the line corresponding to specific sets, then domain tests would all be done at the boundary points, the dividing points on the line that mark the start of one set and/or the end of another.
Reference Materials:
Videos
Introduction to domain testing [6:34] [SLIDES for all parts]
The classical analysis [10:34]
Examples [12:12]
Risk-based equivalence analysis [18:19]
Summary [4:19]
Solutions to examples in the slides [8:51]
[On some browsers, clicking on a video link to play the video will not work. To play the video, download it to your disk and play the downloaded copy with Windows Media Player 9 or later.]
Articles
Ostrand and Balcer, The Category-Partition Method for Specifying and Generating Functional Tests. CACM June 1988 (31:6)
Some Worked Examples
Done

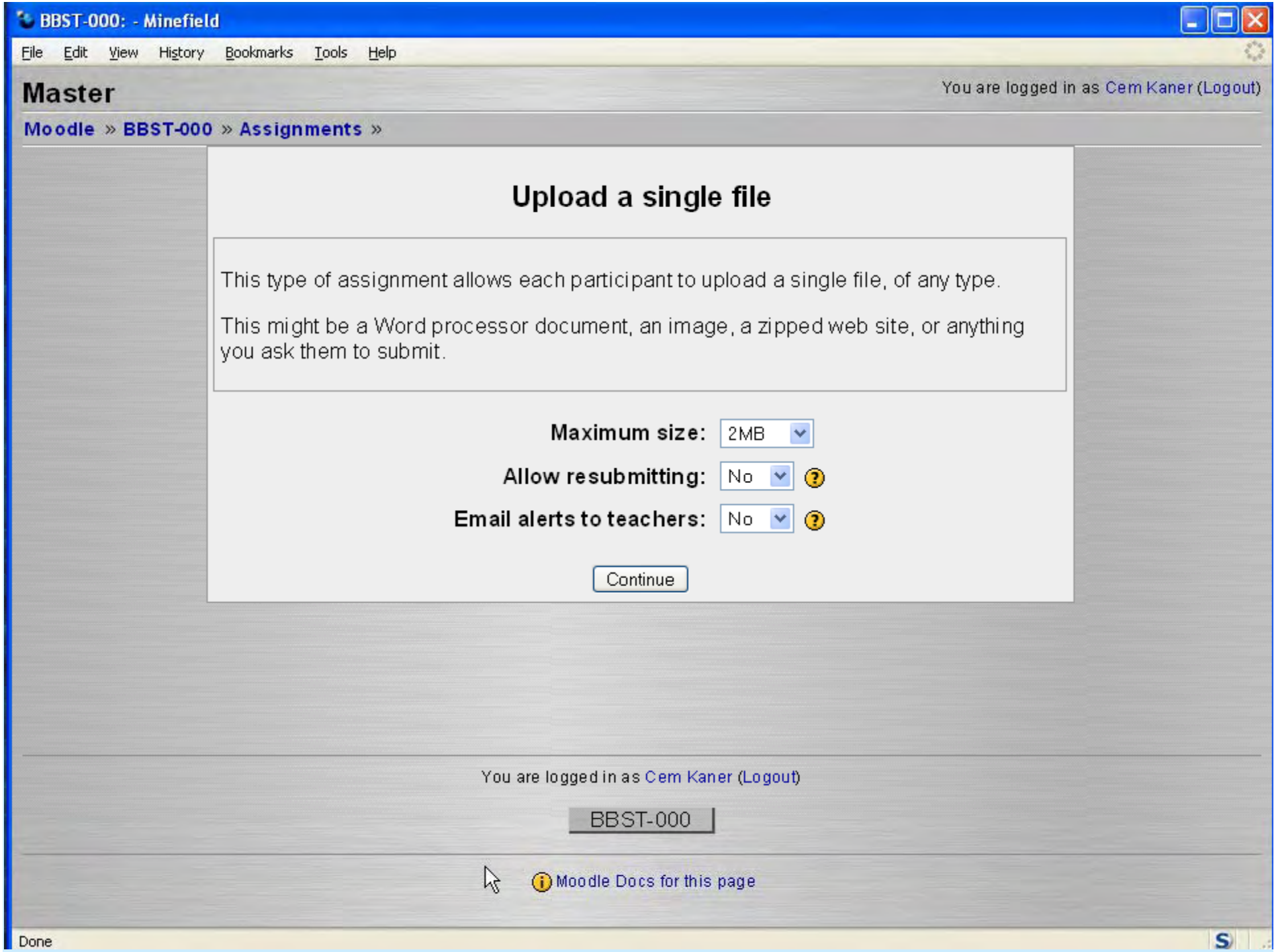

CSE-3411: Section Notes: Domain Testing 1 -- Introducing the approach - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/mod/resource/view.php?id=262
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Software Testing 1
Jump to...
CS » CSE-3411 » Resources » Section Notes: Domain Testing 1 -- Introducing the approach
Update this Resource
Some Worked Examples
Links to examples
[Note: these examples are early draft student projects. They are limited in scope and sometimes a bit rough. However, some students find them quite useful.]
Activities and Assessments:
Activity -- Simple applications of classical domain testing -- -- Submit your answer to this on the main moodle screen
Review / drill questions -- These are available from the main moodle screen
Summary of the Learning Unit
The essence of domain testing is stratified sampling of a few tests from a huge pool of potential tests.
In domain testing, we partition a domain into sub-domains (equivalence classes) and then test using values from each subdomain.
A domain might involve the values of any one variable or combination of variables. Some books look only at input values, but outputs, intermediate calculations, even configuration variables (such as printer type) are commonly analyzed in practical work in the field.
We define an equivalence class as follows: two values are equivalent if, given your theory of possible error, you expect the same test result from each.
The values that we pick to represent each equivalence class are the most powerful members of each set, the best representatives. A best representative is at least as likely to expose an error as any other member of its set.
There are two learning units on domain testing. This first group of material considers the classical approach, some of the problems applying it, and an
Done

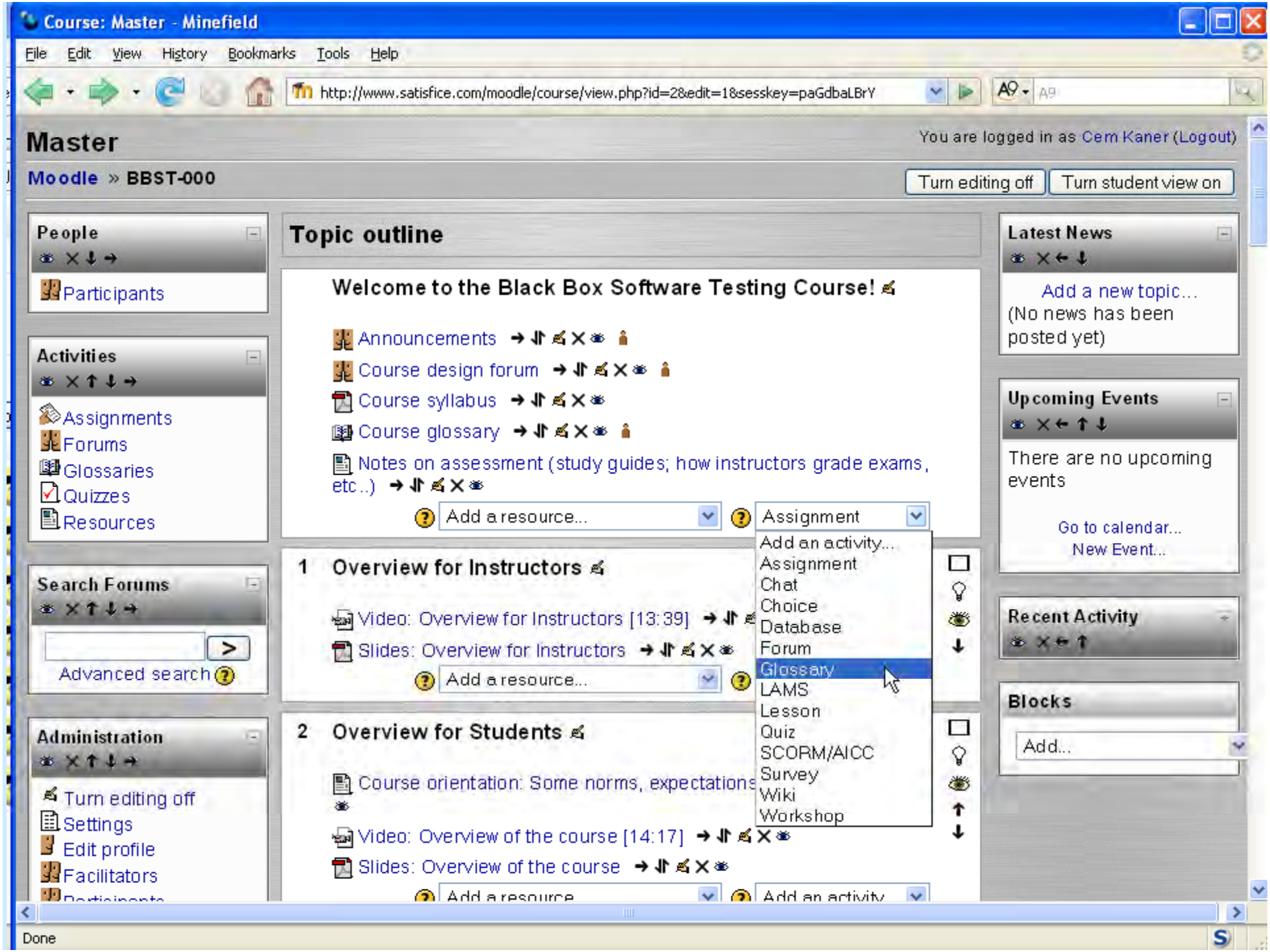

Course: Master - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=2&edit=1&sesskey=paGdbaLBrY
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000
Turn editing off
Turn student view on
People
Participants
Activities
Assignments
Forums
Glossaries
Quizzes
Resources
Search Forums
Advanced search
Administration
Turn editing off
Settings
Edit profile
Topic outline
Welcome to the Black Box Software Testing Course!
Announcements
Course design forum
Course syllabus
Course glossary
Notes on assessment (study guides; how instructors grade exams, etc..)
Add a resource...
Add an activity...
1 Overview for Instructors
Video: Overview for Instructors [13:39]
Slides: Overview for Instructors
Add a resource...
Add an activity...
2 Overview for Students
Course orientation: Some norms, expectations and tips
Video: Overview of the course [14:17]
Latest News
Add a new topic...
(No news has been posted yet)
Upcoming Events
There are no upcoming events
Go to calendar...
New Event...
Recent Activity
Blocks
Add...
Done


Course: Master - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=2&edit=1&sesskey=paGdbaLBrY
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000
Turn editing off
Turn student view on
People
Participants
Activities
Assignments
Forums
Glossaries
Quizzes
Resources
Search Forums
Advanced search
Administration
Turn editing off
Settings
Edit profile
Topic outline
Welcome to the Black Box Software Testing Course!
Announcements
Course design forum
Course syllabus
Course glossary
Notes on assessment (study guides; how instructors grade exams, etc..)
Add a resource...
Add an activity...
Add a resource...
Compose a text page
Compose a web page
Link to a file or web site
Display a directory
Add an IMS Content Package
Insert a label
1 Overview
Video: Ov
Slides: Ov
Add an activity...
2 Overview for Students
Course orientation: Some norms, expectations and tips
Video: Overview of the course [14:17]
Latest News
Add a new topic...
(No news has been posted yet)
Upcoming Events
There are no upcoming events
Go to calendar...
New Event...
Recent Activity
Blocks
Add...
Done

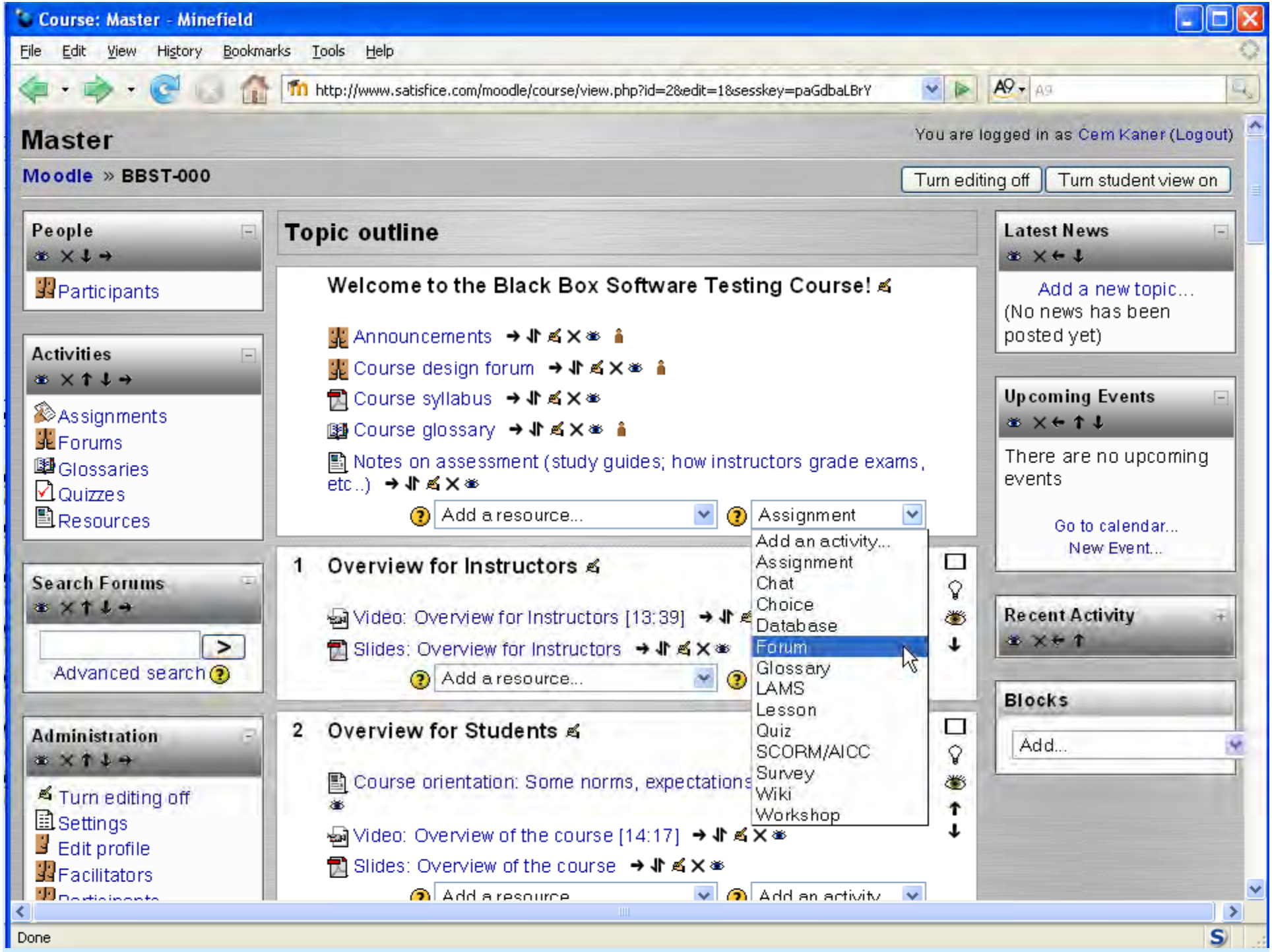

Course: Master - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=2&edit=1&sesskey=paGdbaLBrY
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000
Turn editing off
Turn student view on
People
Participants
Activities
Assignments
Forums
Glossaries
Quizzes
Resources
Search Forums
Advanced search
Administration
Turn editing off
Settings
Edit profile
Topic outline
Welcome to the Black Box Software Testing Course!
Announcements
Course design forum
Course syllabus
Course glossary
Notes on assessment (study guides; how instructors grade exams, etc..)
Add a resource...
Add an activity...
1 Overview for Instructors
Video: Overview for Instructors [13:39]
Slides: Overview for Instructors
Add a resource...
2 Overview for Students
Course orientation: Some norms, expectations
Video: Overview of the course [14:17]
Add an activity...
Assignment
Chat
Choice
Database
Forum
Glossary
LAMS
Lesson
Quiz
SCORM/AICC
Survey
Wiki
Workshop
Latest News
Add a new topic...
(No news has been posted yet)
Upcoming Events
There are no upcoming events
Go to calendar...
New Event...
Recent Activity
Blocks
Add...
Done

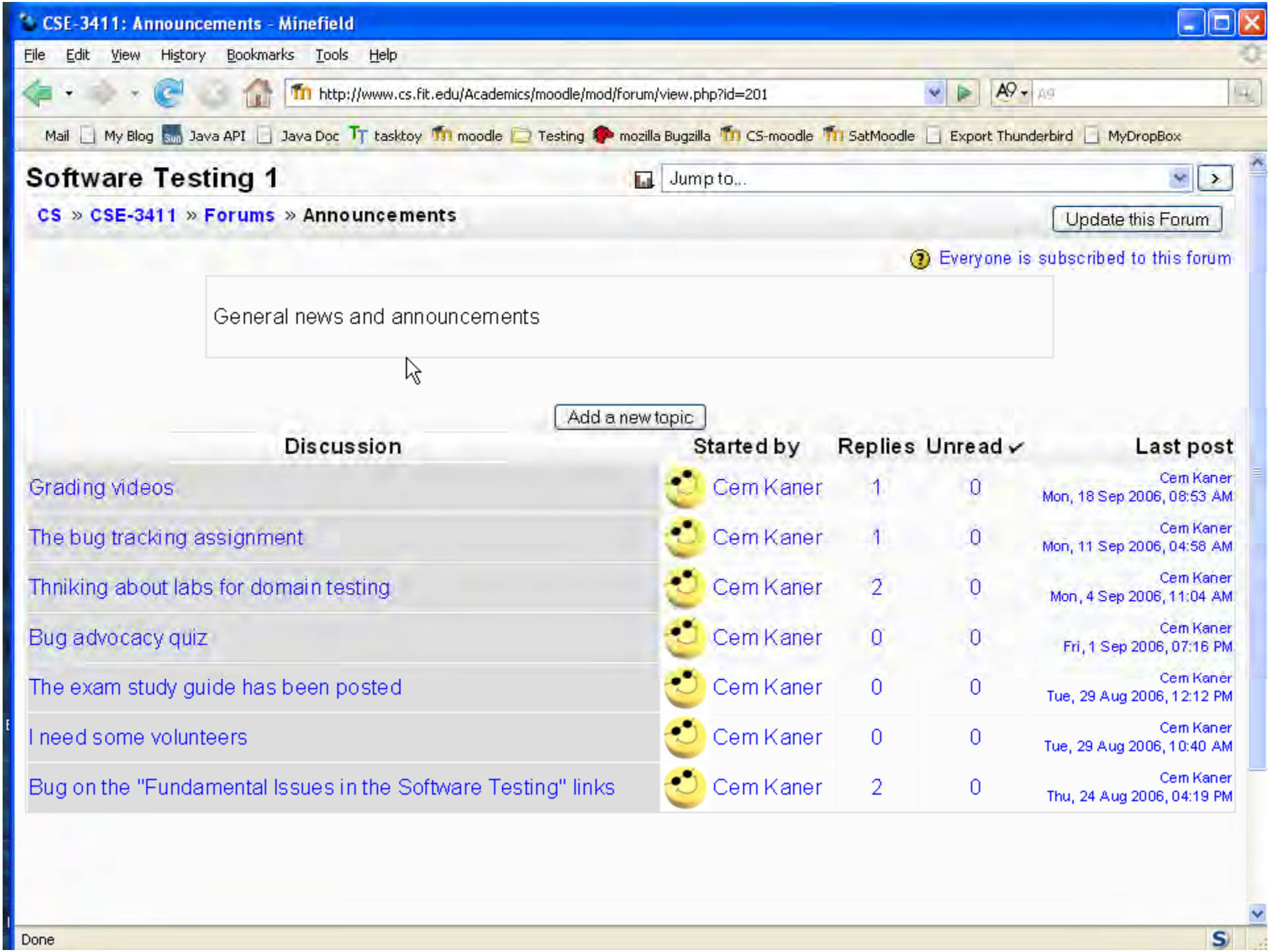

BBST-000: Editing Assignment - Minefield
File Edit View History Bookmarks Tools Help
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000 » Assignments » Editing Assignment
Adding a new Assignment
Assignment name:
Description:
Trebuchet
1 (8 pt)
Write carefully
Ask good questions
About the HTML editor
Path:
Grade:
100
Available from:
27 September 2006 - 08 35
Due date:
4 October 2006 - 08 35
Prevent late submissions:
No
Done

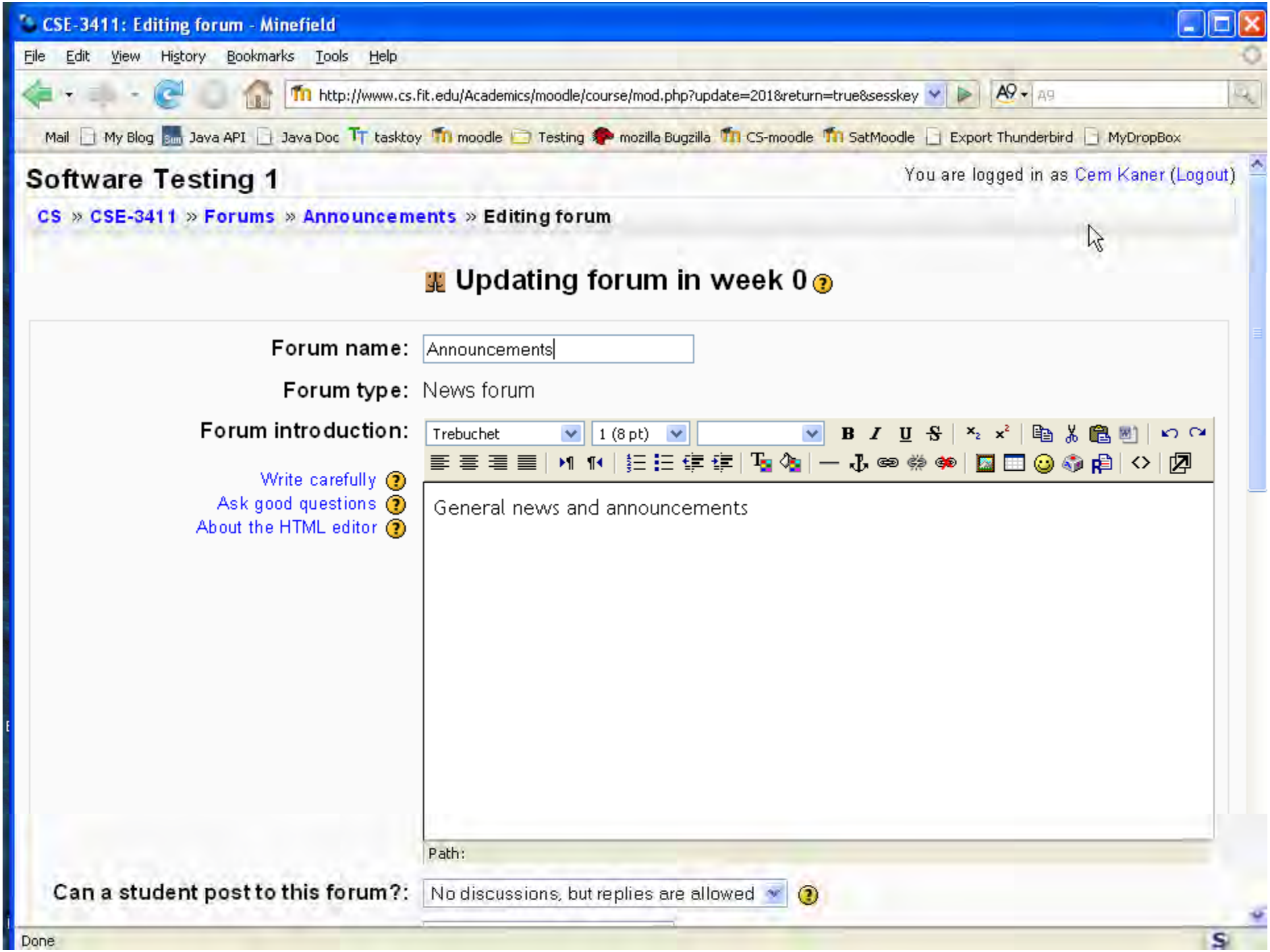

BBST-000: - Minefield
File Edit View History Bookmarks Tools Help
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000 » Assignments »
Upload a single file
This type of assignment allows each participant to upload a single file, of any type.
This might be a Word processor document, an image, a zipped web site, or anything you ask them to submit.
Maximum size: 2MB
Allow resubmitting: No
Email alerts to teachers: No
Continue
You are logged in as Cem Kaner (Logout)
BBST-000
Moodle Docs for this page
Done

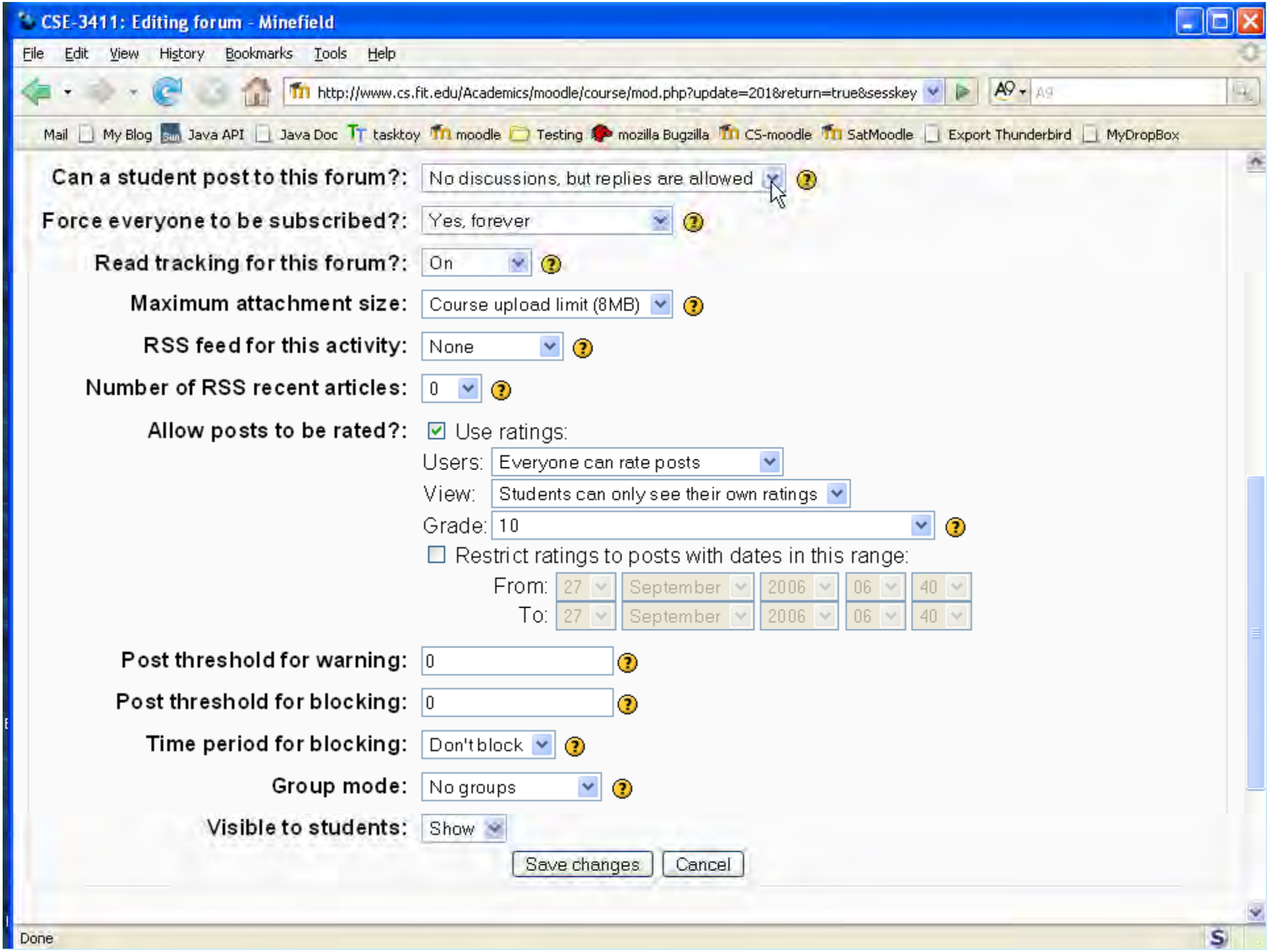

Course: Master - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=2&edit=1&sesskey=paGdbaLBrY
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000
Turn editing off
Turn student view on
People
Participants
Activities
Assignments
Forums
Glossaries
Quizzes
Resources
Search Forums
Advanced search
Administration
Turn editing off
Settings
Edit profile
Facilitators
Topic outline
Welcome to the Black Box Software Testing Course!
Announcements
Course design forum
Course syllabus
Course glossary
Notes on assessment (study guides; how instructors grade exams, etc..)
Add a resource...
Assignment
1 Overview for Instructors
Video: Overview for Instructors [13:39]
Slides: Overview for Instructors
Add a resource...
2 Overview for Students
Course orientation: Some norms, expectations
Video: Overview of the course [14:17]
Slides: Overview of the course
Add a resource
Add an activity...
Add an activity...
Assignment
Chat
Choice
Database
Forum
Glossary
LAMS
Lesson
Quiz
SCORM/AICC
Survey
Wiki
Workshop
Latest News
Add a new topic...
(No news has been posted yet)
Upcoming Events
There are no upcoming events
Go to calendar...
New Event...
Recent Activity
Blocks
Add...
Done

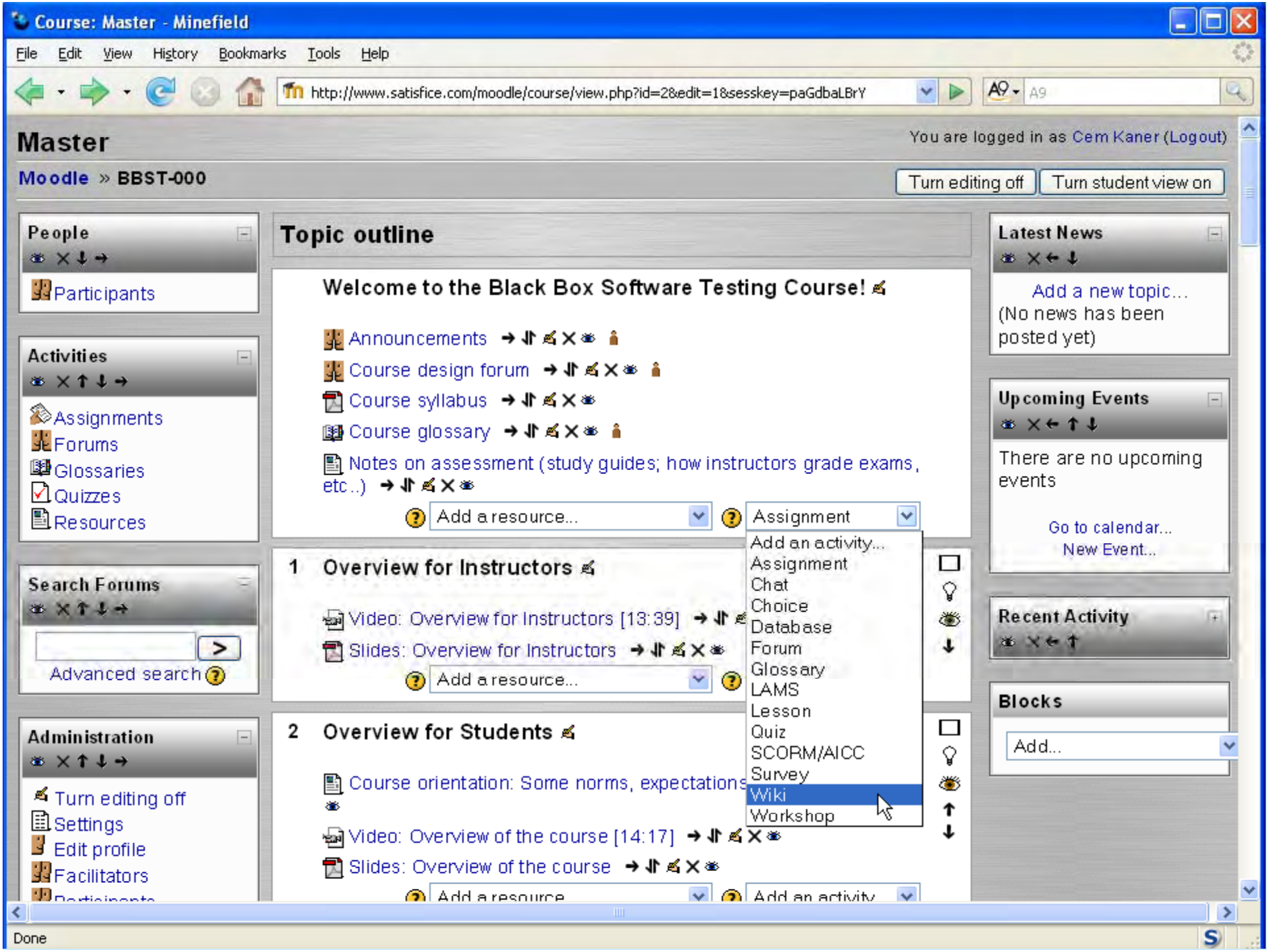

CSE1001-Kaner: Course Glossary - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/mod/glossary/view.php?id=197&mode=letter&hook=
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
CS 1001 A test-first introduction to Java programming
Jump to...
CS » CSE1001-Kaner » Glossaries » Course Glossary
Update this Glossary
Course Glossary
Search
Search full text
Browse by alphabet
Browse by category
Browse by date
Browse by Author
Add a new entry
Import entries
Export entries
Waiting approval
Browse the glossary using this index
Special | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O
P | Q | R | S | T | U | V | W | X | Y | Z | ALL
O
object
(Last edited: Saturday, 19 August 2006, 11:29 AM)
"an entity that has identity, type, and state; objects are created from classes"
Reference: Langr, Appendix A
"The principal building blocks of object-oriented programs. Each object is a programming unit consisting of data (instance variables) and functionality (instance methods). See also class."
Reference: http://java.sun.com/docs/books/tutorial/information/glossary.html
Done

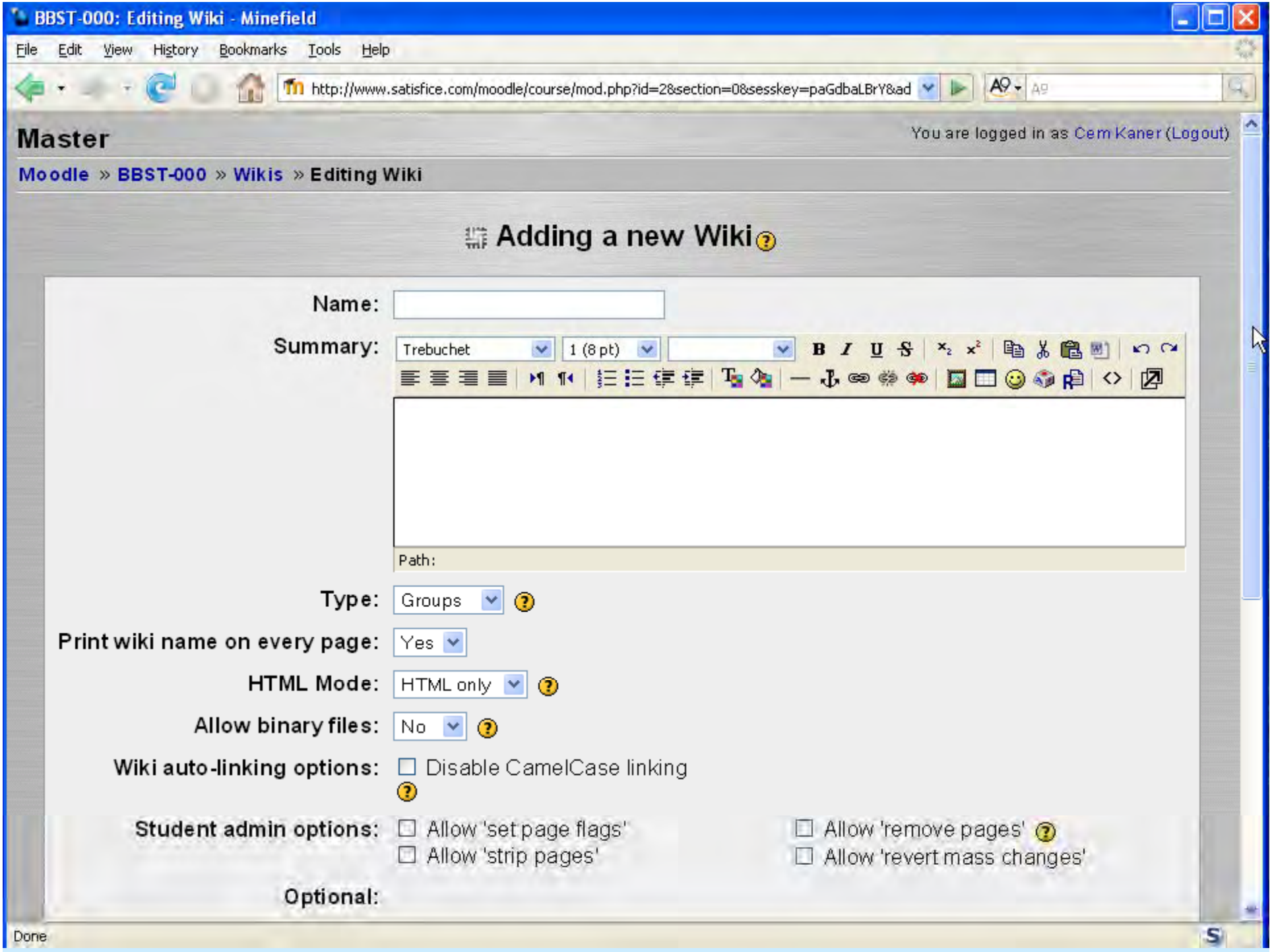

Course: Master - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=2&edit=1&sesskey=paGdbaLBrY
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000
Turn editing off
Turn student view on
People
Participants
Activities
Assignments
Forums
Glossaries
Quizzes
Resources
Search Forums
Advanced search
Administration
Turn editing off
Settings
Edit profile
Facilitators
Topic outline
Welcome to the Black Box Software Testing Course!
Announcements
Course design forum
Course syllabus
Course glossary
Notes on assessment (study guides; how instructors grade exams, etc..)
Add a resource...
Assignment
Add an activity...
Assignment
Chat
Choice
Database
Forum
Glossary
LAMS
Lesson
Quiz
SCORM/AICC
Survey
Wiki
Workshop
1 Overview for Instructors
Video: Overview for Instructors [13:39]
Slides: Overview for Instructors
Add a resource...
2 Overview for Students
Course orientation: Some norms, expectations
Video: Overview of the course [14:17]
Slides: Overview of the course
Add a resource
Add an activity
Latest News
Add a new topic...
(No news has been posted yet)
Upcoming Events
There are no upcoming events
Go to calendar...
New Event...
Recent Activity
Blocks
Add...
Done

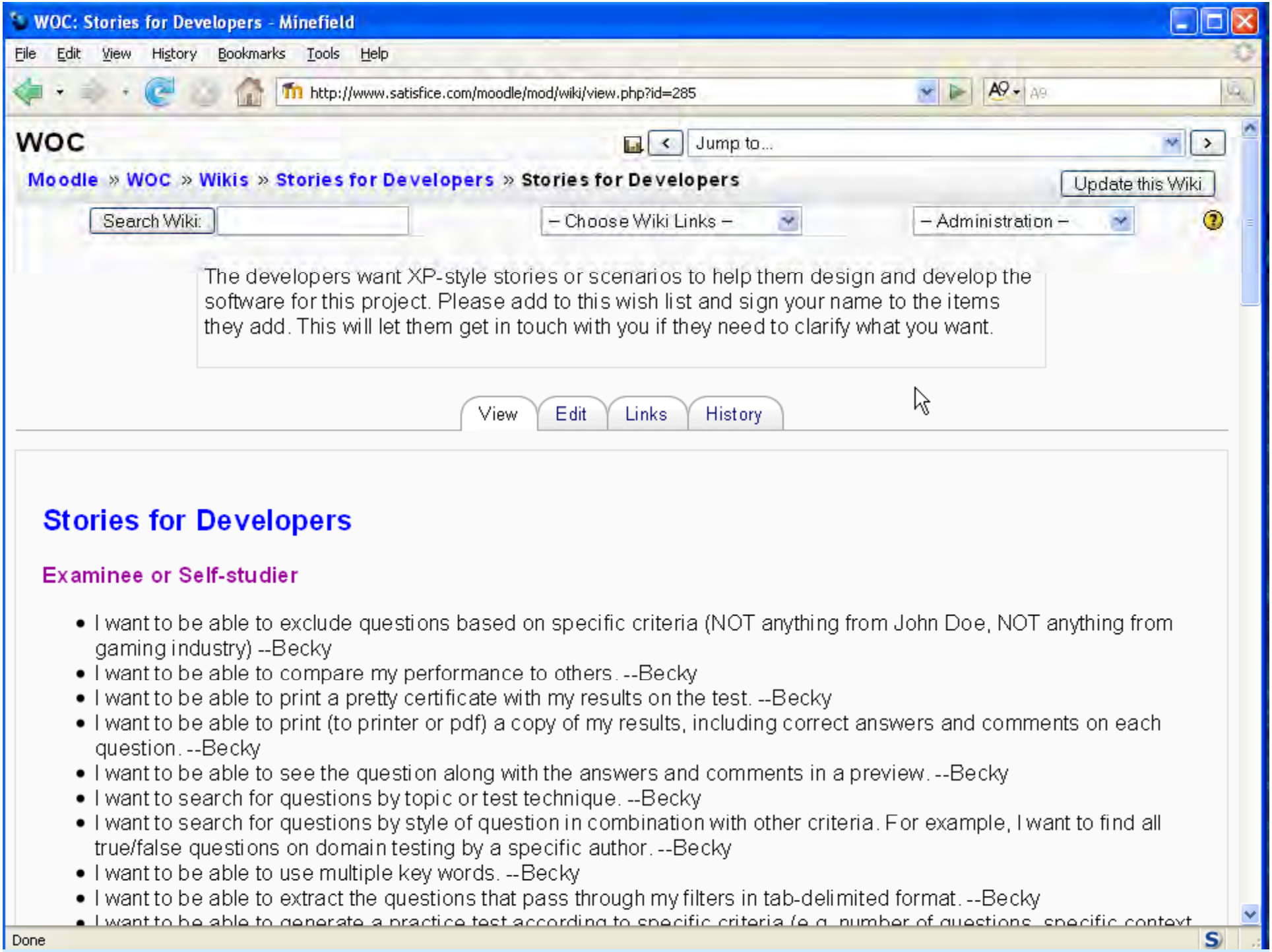

CSE-3411: Announcements - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/mod/forum/view.php?id=201
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Software Testing 1
Jump to...
CS » CSE-3411 » Forums » Announcements
Update this Forum
Everyone is subscribed to this forum
General news and announcements
Add a new topic
Discussion
Started by
Replies
Unread
Last post
Grading videos
Cem Kaner
1
0
Cem Kaner
Mon, 18 Sep 2006, 08:53 AM
The bug tracking assignment
Cem Kaner
1
0
Cem Kaner
Mon, 11 Sep 2006, 04:58 AM
Thniking about labs for domain testing
Cem Kaner
2
0
Cem Kaner
Mon, 4 Sep 2006, 11:04 AM
Bug advocacy quiz
Cem Kaner
0
0
Cem Kaner
Fri, 1 Sep 2006, 07:16 PM
The exam study guide has been posted
Cem Kaner
0
0
Cem Kaner
Tue, 29 Aug 2006, 12:12 PM
I need some volunteers
Cem Kaner
0
0
Cem Kaner
Tue, 29 Aug 2006, 10:40 AM
Bug on the "Fundamental Issues in the Software Testing" links
Cem Kaner
2
0
Cem Kaner
Thu, 24 Aug 2006, 04:19 PM
Done

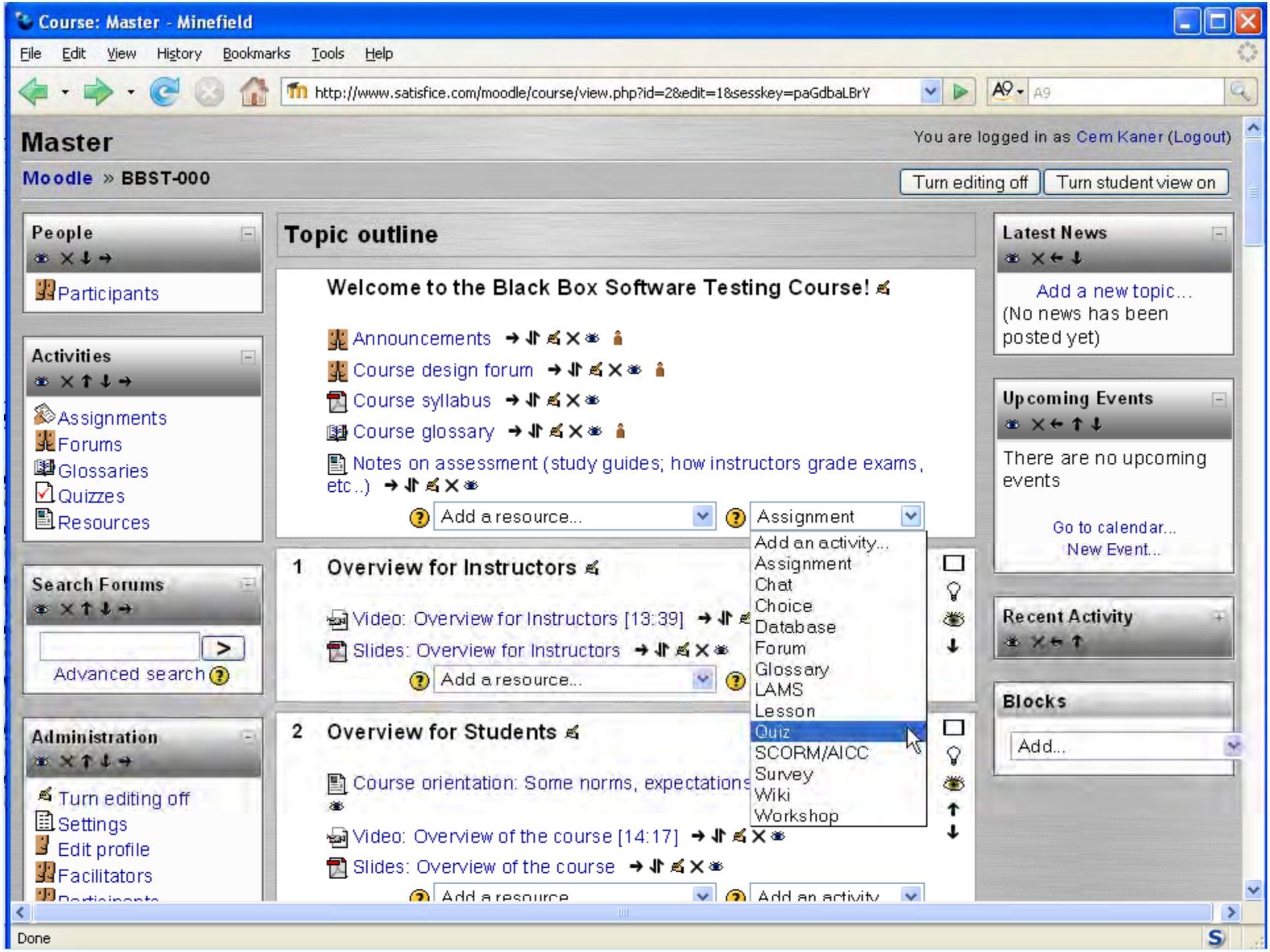

CSE-3411: Editing forum - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/course/mod.php?update=201&return=true&sesskey
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Software Testing 1
You are logged in as Cem Kaner (Logout)
CS » CSE-3411 » Forums » Announcements » Editing forum
Updating forum in week 0
Forum name: Announcements
Forum type: News forum
Forum introduction:
Trebuchet
1 (8 pt)
Write carefully
Ask good questions
About the HTML editor
General news and announcements
Path:
Can a student post to this forum?: No discussions, but replies are allowed
Done

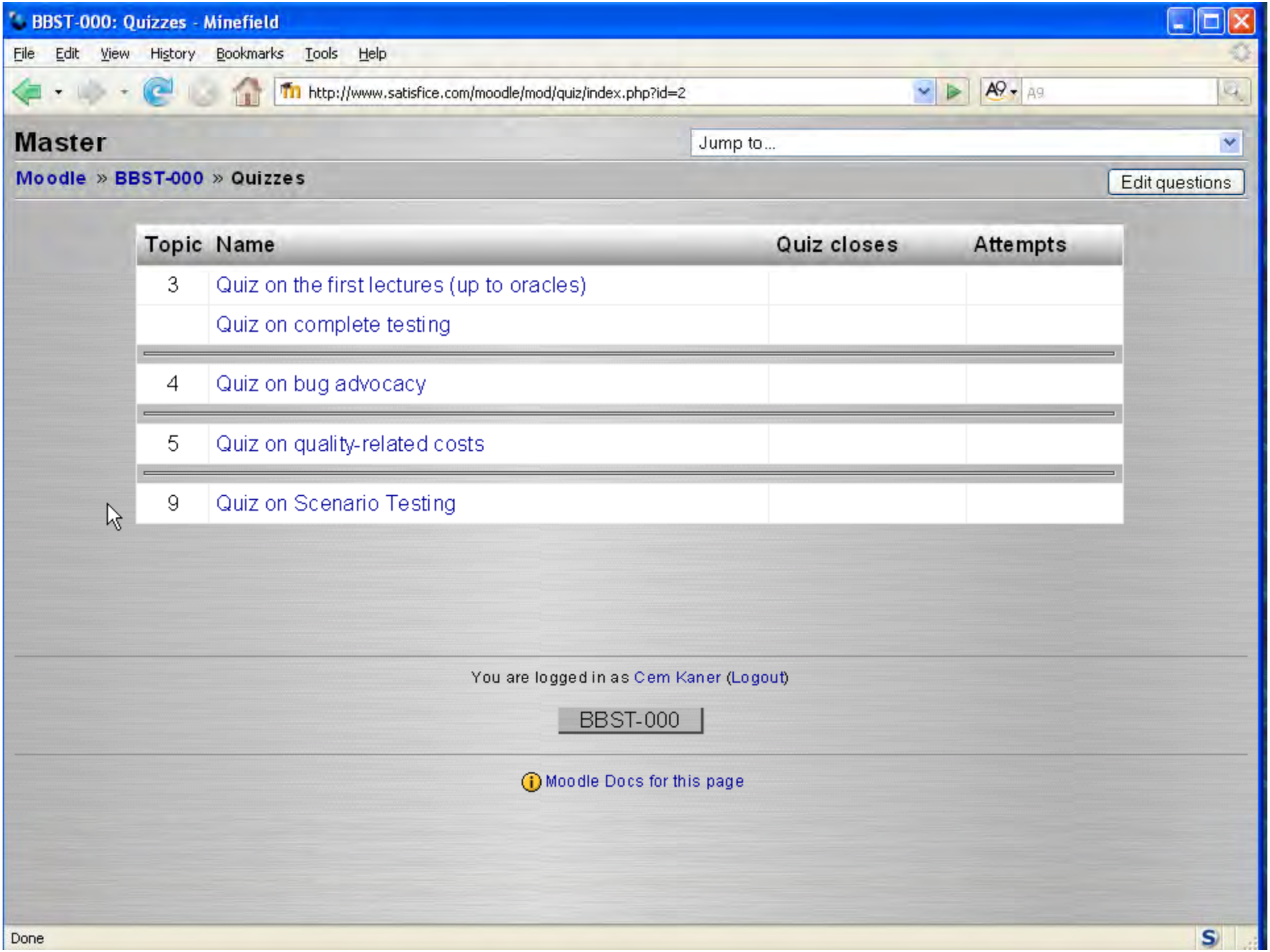

CSE-3411: Editing forum - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/course/mod.php?update=201&return=true&sesskey
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Can a student post to this forum?: No discussions, but replies are allowed
Force everyone to be subscribed?: Yes, forever
Read tracking for this forum?: On
Maximum attachment size: Course upload limit (8MB)
RSS feed for this activity: None
Number of RSS recent articles: 0
Allow posts to be rated?: Use ratings:
Users: Everyone can rate posts
View: Students can only see their own ratings
Grade: 10
Restrict ratings to posts with dates in this range:
From: 27 September 2006 06 40
To: 27 September 2006 06 40
Post threshold for warning: 0
Post threshold for blocking: 0
Time period for blocking: Don't block
Group mode: No groups
Visible to students: Show
Save changes
Cancel
Done

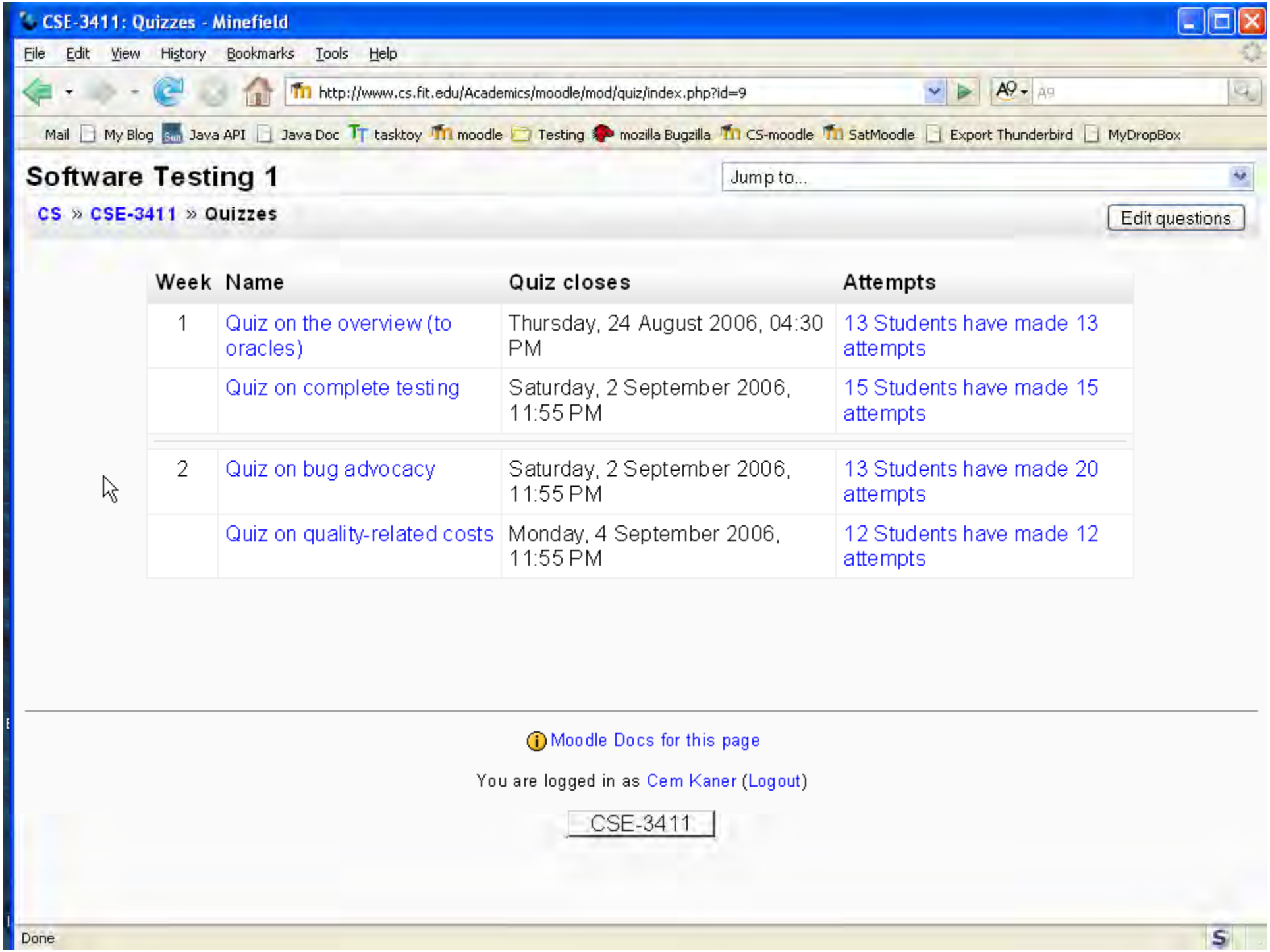

Course: Master - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=2&edit=1&sesskey=paGdbaLBrY
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000
Turn editing off
Turn student view on
People
Participants
Activities
Assignments
Forums
Glossaries
Quizzes
Resources
Search Forums
Advanced search
Administration
Turn editing off
Settings
Edit profile
Facilitators
Topic outline
Welcome to the Black Box Software Testing Course!
Announcements
Course design forum
Course syllabus
Course glossary
Notes on assessment (study guides; how instructors grade exams, etc..)
Add a resource...
Assignment
Add an activity...
Assignment
Chat
Choice
Database
Forum
Glossary
LAMS
Lesson
Quiz
SCORM/AICC
Survey
Wiki
Workshop
1 Overview for Instructors
Video: Overview for Instructors [13:39]
Slides: Overview for Instructors
Add a resource...
2 Overview for Students
Course orientation: Some norms, expectations
Video: Overview of the course [14:17]
Slides: Overview of the course
Add a resource
Add an activity
Latest News
Add a new topic...
(No news has been posted yet)
Upcoming Events
There are no upcoming events
Go to calendar...
New Event...
Recent Activity
Blocks
Add...
Done

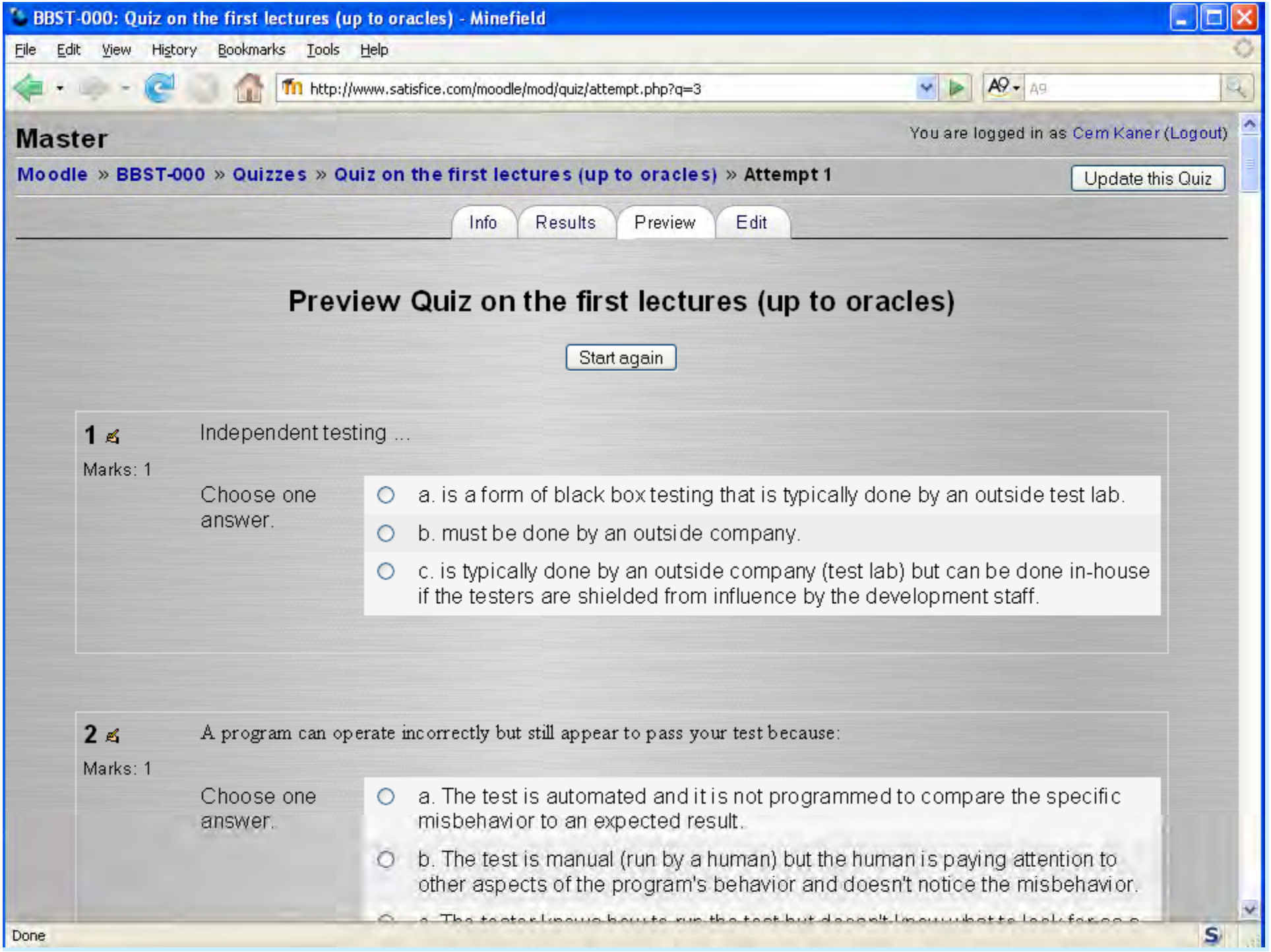

BBST-000: Editing Wiki - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/mod.php?id=2&section=0&sesskey=paGdbaLBrY&ad
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000 » Wikis » Editing Wiki
Adding a new Wiki
Name:
Summary:
Trebuchet
1 (8 pt)
Path:
Type:
Groups
Print wiki name on every page:
Yes
HTML Mode:
HTML only
Allow binary files:
No
Wiki auto-linking options:
Disable CamelCase linking
Student admin options:
Allow 'set page flags'
Allow 'strip pages'
Allow 'remove pages'
Allow 'revert mass changes'
Optional:
Done

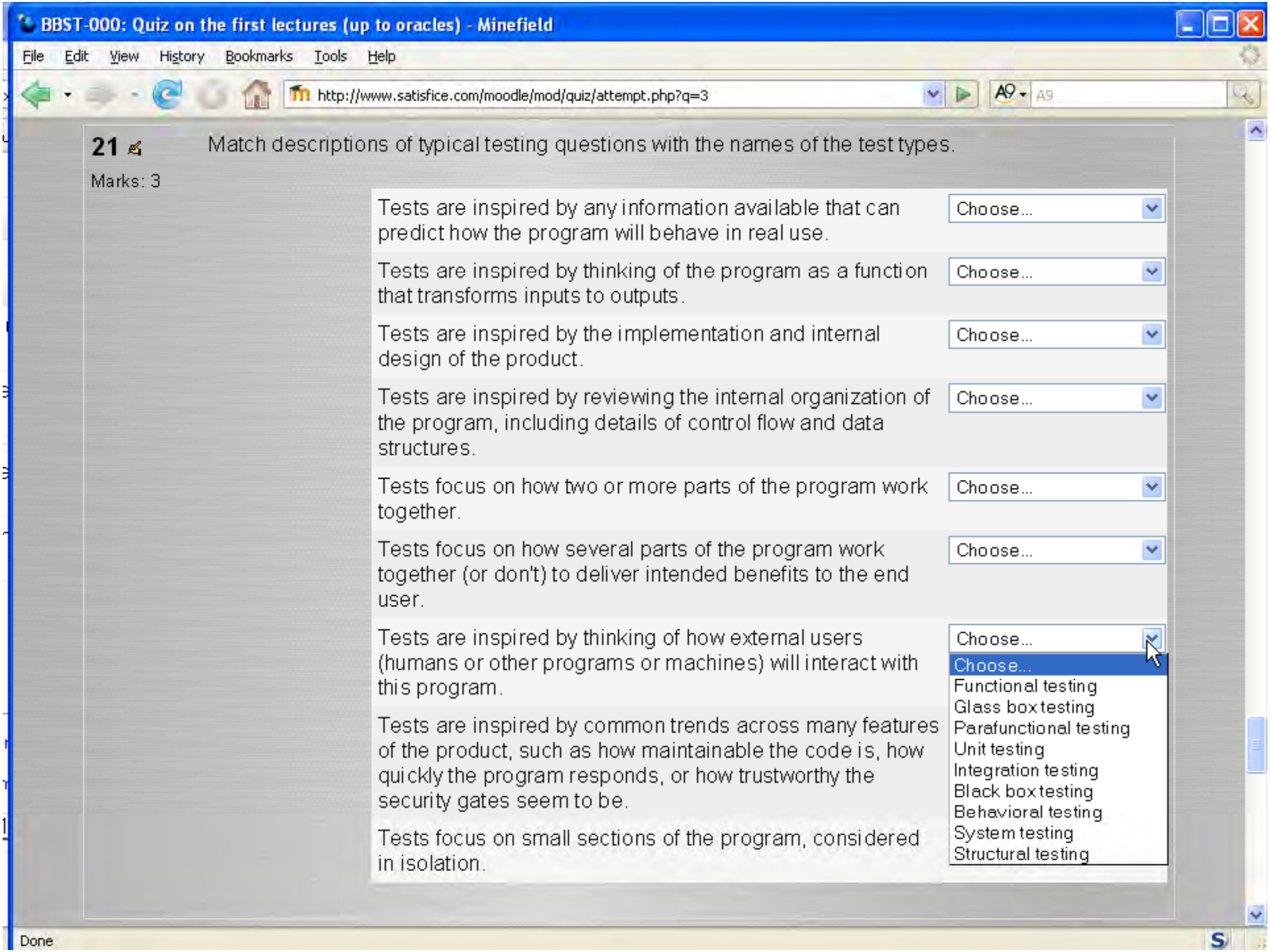

WOC: Stories for Developers - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/mod/wiki/view.php?id=285
WOC
Jump to...
Moodle » WOC » Wikis » Stories for Developers » Stories for Developers
Update this Wiki
Search Wiki:
– Choose Wiki Links –
– Administration –
The developers want XP-style stories or scenarios to help them design and develop the software for this project. Please add to this wish list and sign your name to the items they add. This will let them get in touch with you if they need to clarify what you want.
View
Edit
Links
History
Stories for Developers
Examinee or Self-studier
• I want to be able to exclude questions based on specific criteria (NOT anything from John Doe, NOT anything from gaming industry) --Becky
• I want to be able to compare my performance to others. --Becky
• I want to be able to print a pretty certificate with my results on the test. --Becky
• I want to be able to print (to printer or pdf) a copy of my results, including correct answers and comments on each question. --Becky
• I want to be able to see the question along with the answers and comments in a preview. --Becky
• I want to search for questions by topic or test technique. --Becky
• I want to search for questions by style of question in combination with other criteria. For example, I want to find all true/false questions on domain testing by a specific author. --Becky
• I want to be able to use multiple key words. --Becky
• I want to be able to extract the questions that pass through my filters in tab-delimited format. --Becky
Done

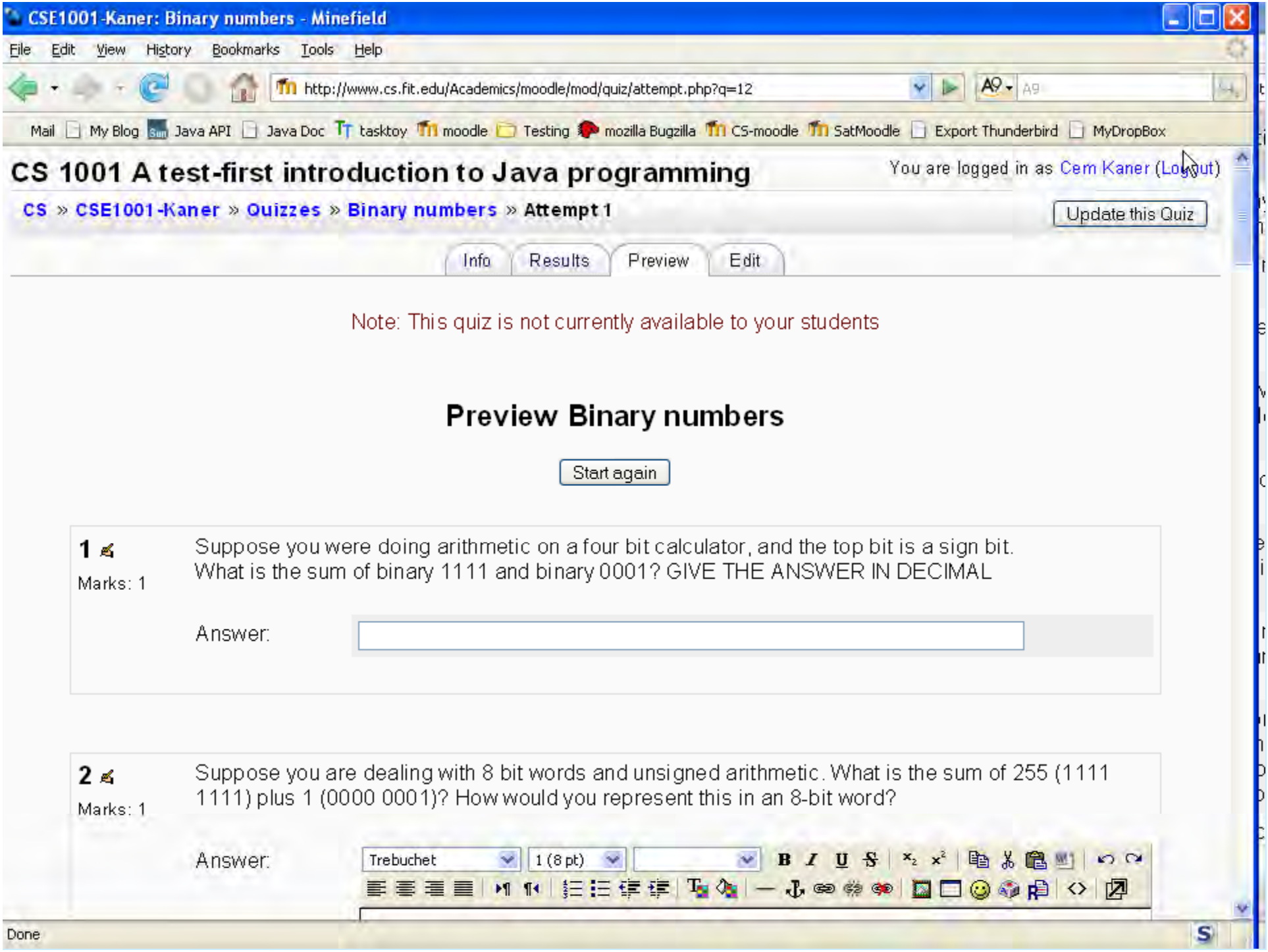

Course: Master - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=2&edit=1&sesskey=paGdbaLBrY
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000
Turn editing off
Turn student view on
People
Participants
Activities
Assignments
Forums
Glossaries
Quizzes
Resources
Search Forums
Advanced search
Administration
Turn editing off
Settings
Edit profile
Facilitators
Topic outline
Welcome to the Black Box Software Testing Course!
Announcements
Course design forum
Course syllabus
Course glossary
Notes on assessment (study guides; how instructors grade exams, etc..)
Add a resource...
Assignment
Add an activity...
Assignment
Chat
Choice
Database
Forum
Glossary
LAMS
Lesson
Quiz
SCORM/AICC
Survey
Wiki
Workshop
1 Overview for Instructors
Video: Overview for Instructors [13:39]
Slides: Overview for Instructors
Add a resource...
2 Overview for Students
Course orientation: Some norms, expectations
Video: Overview of the course [14:17]
Slides: Overview of the course
Latest News
Add a new topic...
(No news has been posted yet)
Upcoming Events
There are no upcoming events
Go to calendar...
New Event...
Recent Activity
Blocks
Add...
Done

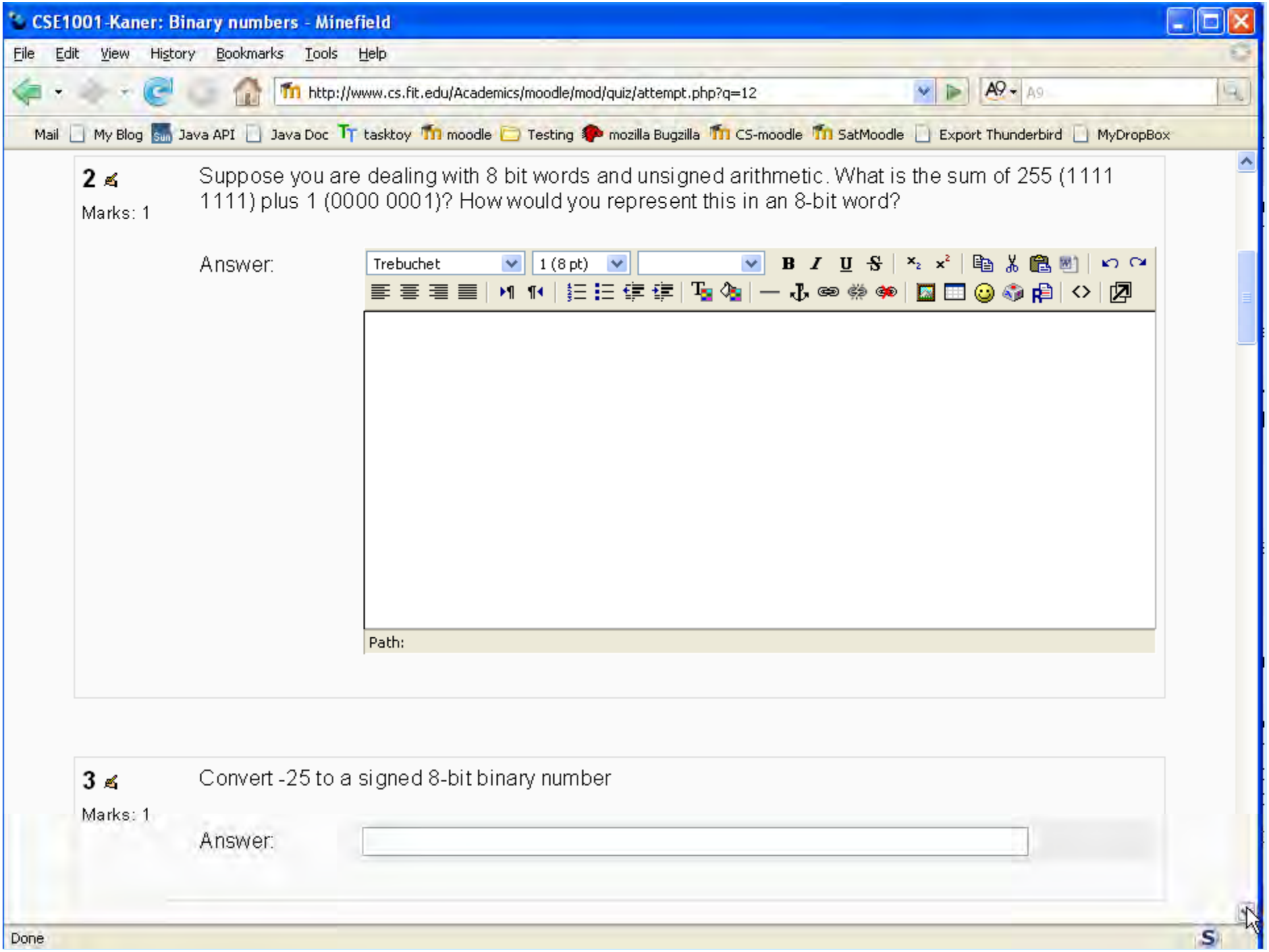

BBST-000: Quizzes - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/mod/quiz/index.php?id=2
Master
Jump to...
Moodle » BBST-000 » Quizzes
Edit questions
Topic Name Quiz closes Attempts
3 Quiz on the first lectures (up to oracles)
Quiz on complete testing
4 Quiz on bug advocacy
5 Quiz on quality-related costs
9 Quiz on Scenario Testing
You are logged in as Cem Kaner (Logout)
BBST-000
Moodle Docs for this page
Done


CSE-3411: Quizzes - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/mod/quiz/index.php?id=9
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
Software Testing 1
Jump to...
CS » CSE-3411 » Quizzes
Edit questions
Week Name Quiz closes Attempts
1 Quiz on the overview (to oracles) Thursday, 24 August 2006, 04:30 PM 13 Students have made 13 attempts
Quiz on complete testing Saturday, 2 September 2006, 11:55 PM 15 Students have made 15 attempts
2 Quiz on bug advocacy Saturday, 2 September 2006, 11:55 PM 13 Students have made 20 attempts
Quiz on quality-related costs Monday, 4 September 2006, 11:55 PM 12 Students have made 12 attempts
Moodle Docs for this page
You are logged in as Cem Kaner (Logout)
CSE-3411
Done


BBST-000: Quiz on the first lectures (up to oracles) - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/mod/quiz/attempt.php?q=3
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000 » Quizzes » Quiz on the first lectures (up to oracles) » Attempt 1
Update this Quiz
Info Results Preview Edit
Preview Quiz on the first lectures (up to oracles)
Start again
1
Independent testing ...
Marks: 1
Choose one answer.
a. is a form of black box testing that is typically done by an outside test lab.
b. must be done by an outside company.
c. is typically done by an outside company (test lab) but can be done in-house if the testers are shielded from influence by the development staff.
2
A program can operate incorrectly but still appear to pass your test because:
Marks: 1
Choose one answer.
a. The test is automated and it is not programmed to compare the specific misbehavior to an expected result.
b. The test is manual (run by a human) but the human is paying attention to other aspects of the program's behavior and doesn't notice the misbehavior.
Done


BBST-000: Quiz on the first lectures (up to oracles) - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/mod/quiz/attempt.php?q=3
21
Match descriptions of typical testing questions with the names of the test types.
Marks: 3
Tests are inspired by any information available that can predict how the program will behave in real use.
Tests are inspired by thinking of the program as a function that transforms inputs to outputs.
Tests are inspired by the implementation and internal design of the product.
Tests are inspired by reviewing the internal organization of the program, including details of control flow and data structures.
Tests focus on how two or more parts of the program work together.
Tests focus on how several parts of the program work together (or don't) to deliver intended benefits to the end user.
Tests are inspired by thinking of how external users (humans or other programs or machines) will interact with this program.
Tests are inspired by common trends across many features of the product, such as how maintainable the code is, how quickly the program responds, or how trustworthy the security gates seem to be.
Tests focus on small sections of the program, considered in isolation.
Choose...
Choose...
Functional testing
Glass box testing
Parafunctional testing
Unit testing
Integration testing
Black box testing
Behavioral testing
System testing
Structural testing
Done


CSE1001-Kaner: Binary numbers - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/mod/quiz/attempt.php?q=12
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
CS 1001 A test-first introduction to Java programming
You are logged in as Cem Kaner (Logout)
CS » CSE1001-Kaner » Quizzes » Binary numbers » Attempt 1
Update this Quiz
Info Results Preview Edit
Note: This quiz is not currently available to your students
Preview Binary numbers
Start again
1
Marks: 1
Suppose you were doing arithmetic on a four bit calculator, and the top bit is a sign bit. What is the sum of binary 1111 and binary 0001? GIVE THE ANSWER IN DECIMAL
Answer:
2
Marks: 1
Suppose you are dealing with 8 bit words and unsigned arithmetic. What is the sum of 255 (1111 1111) plus 1 (0000 0001)? How would you represent this in an 8-bit word?
Answer:
Trebuchet
1 (8 pt)
Done


CSE1001-Kaner: Binary numbers - Minefield
File Edit View History Bookmarks Tools Help
http://www.cs.fit.edu/Academics/moodle/mod/quiz/attempt.php?q=12
Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox
2
Marks: 1
Suppose you are dealing with 8 bit words and unsigned arithmetic. What is the sum of 255 (1111 1111) plus 1 (0000 0001)? How would you represent this in an 8-bit word?
Answer:
Trebuchet
1 (8 pt)
Path:
3
Marks: 1
Convert -25 to a signed 8-bit binary number
Answer:
Done

CSE1001-Kaner: Binary numbers - Minefield

File Edit View History Bookmarks Tools Help

http://www.cs.fit.edu/Academics/moodle/mod/quiz/attempt.php?q=12

Mail My Blog Java API Java Doc tasktoy moodle Testing mozilla Bugzilla CS-moodle SatMoodle Export Thunderbird MyDropBox

**12** Marks: 1

Suppose we are dealing with signed binary numbers? What is the decimal value of 1111 1111?

Choose one answer.

- a. -1
- b. 255
- c. The question doesn't provide enough information. If the word size is 8 bits then -1 is correct. If the word size is bigger than 8, then 255 is correct.

**13** Marks: 1

What is the smallest number of bits needed to store the decimal number 7?

Answer:

**14** Marks: 1

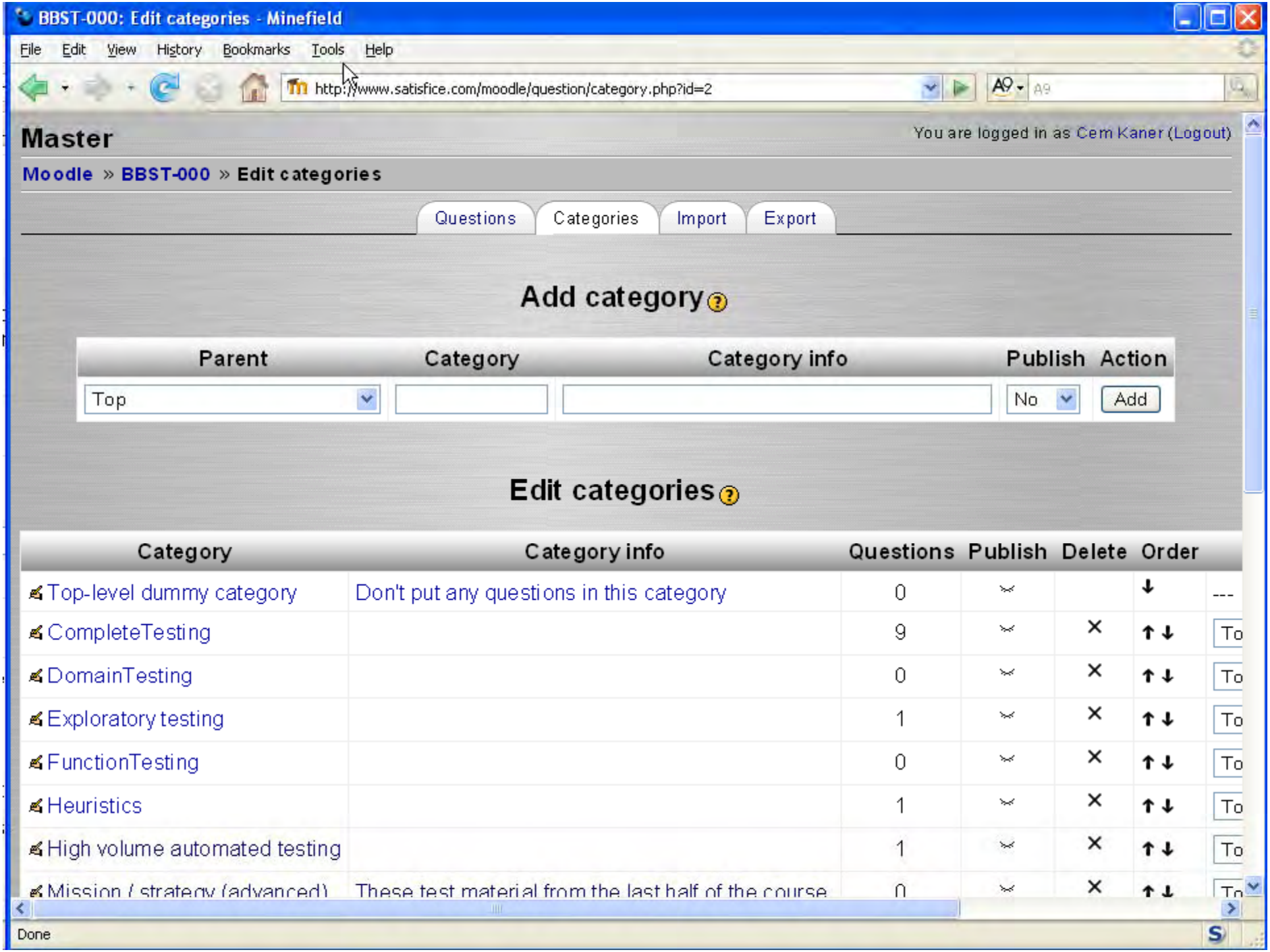
Suppose you decided to assign codes to every letter in the alphabet. We'll make "A" be 1, "B" be 2, and "a" be 27. What is the smallest number of binary digits you would need to be able to represent letters in binary?

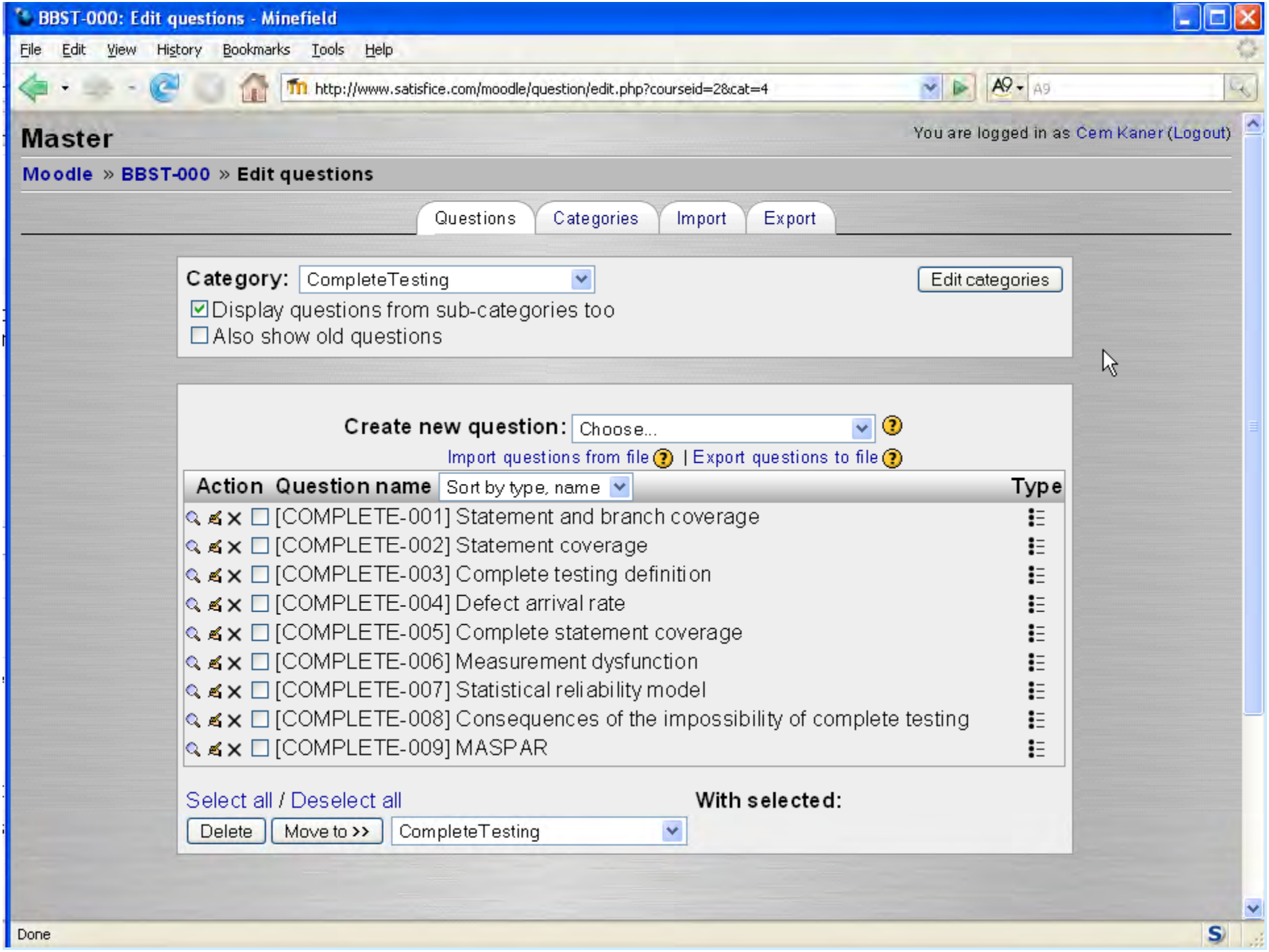
Choose one answer.

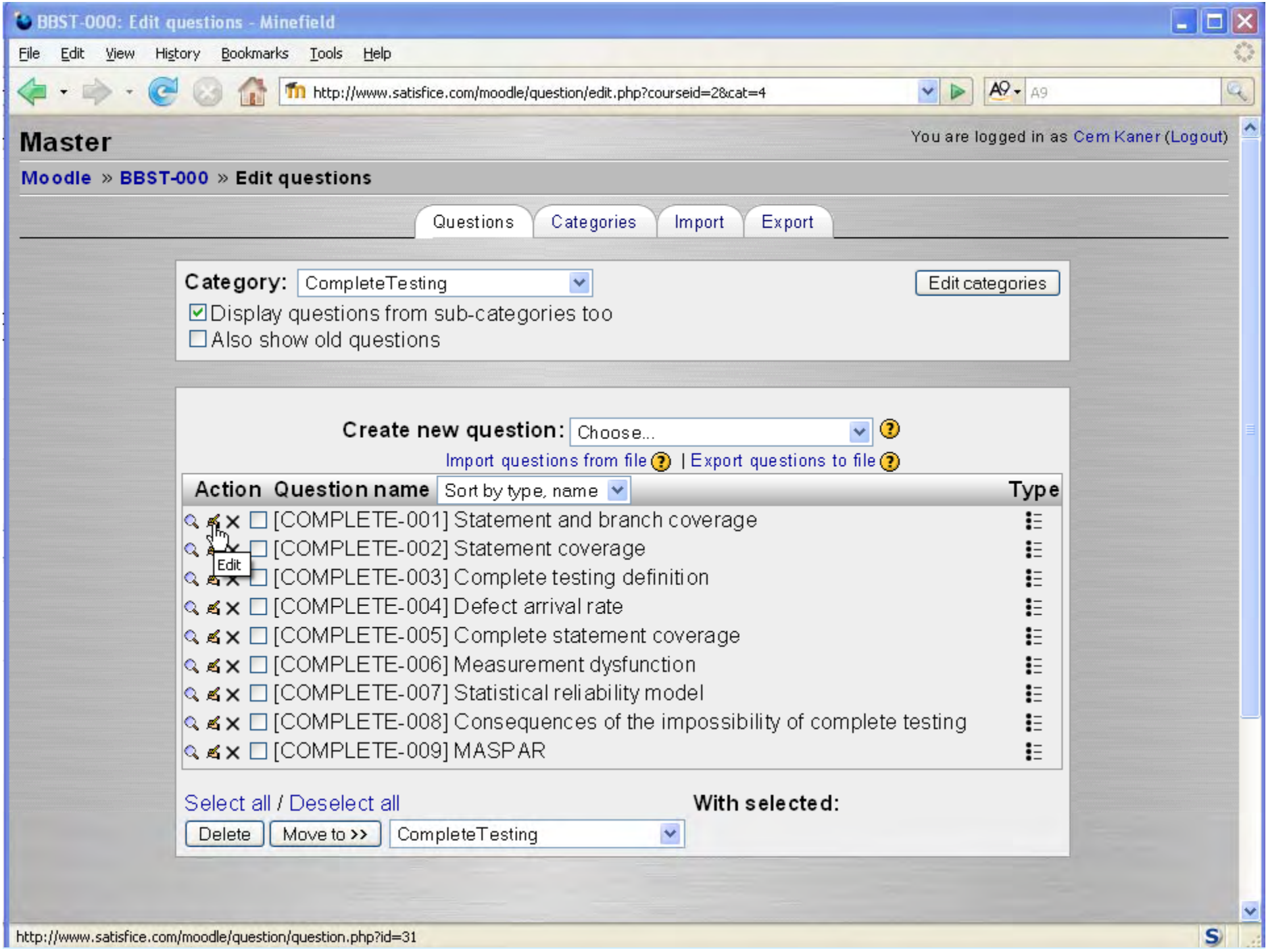
- a. Six, because the largest letter is "z" and its code would be 52 (110100)
- b. 26, because you could store lower case as zero and upper case as 1
- c. Six, because the largest letter is "z" and its code would be 52 (110101)
- d. 52, one for each letter

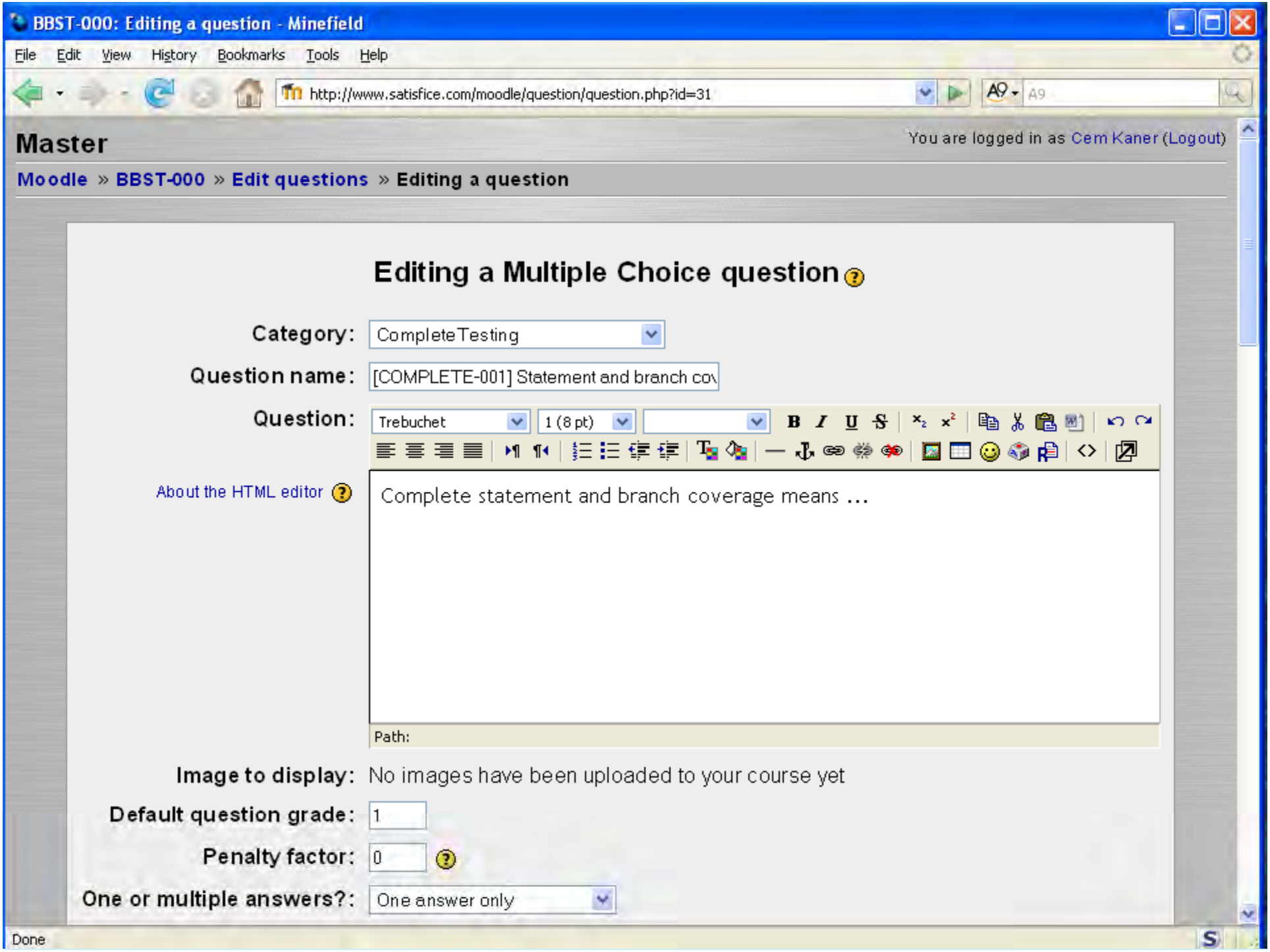
Done

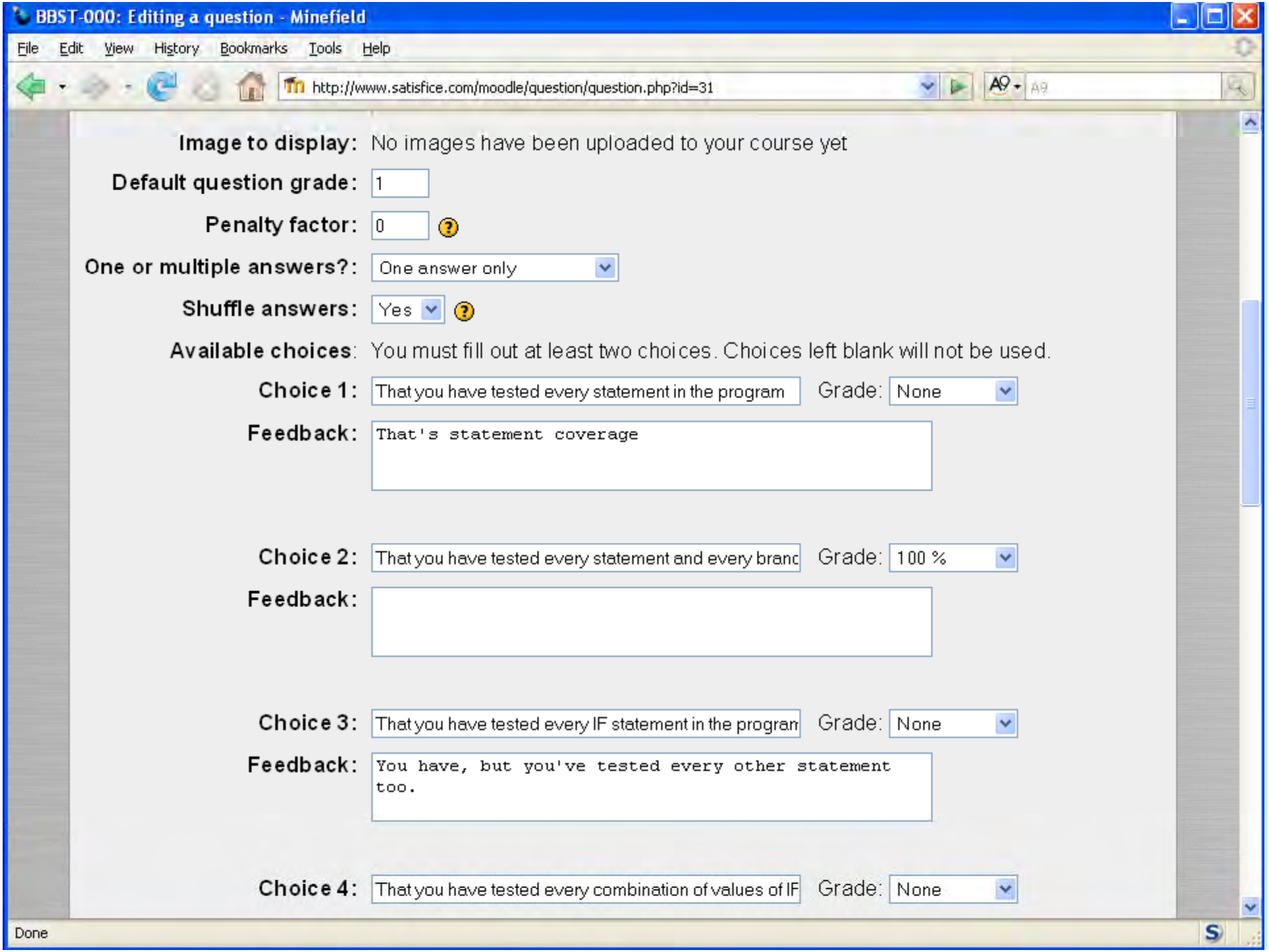

BBST-000: Edit questions - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/question/edit.php?courseid=2&clean=true
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000 » Edit questions
Questions
Categories
Import
Export
Category: Top-level dummy category
Edit categories
Display questions from sub-categories too
Also show old questions
Don't put any questions in this category
Create new question:
Choose...
Import questions from file
Export questions to file
No questions have been added yet
You are logged in as Cem Kaner (Logout)
BBST-000
Moodle Docs for this page
Done

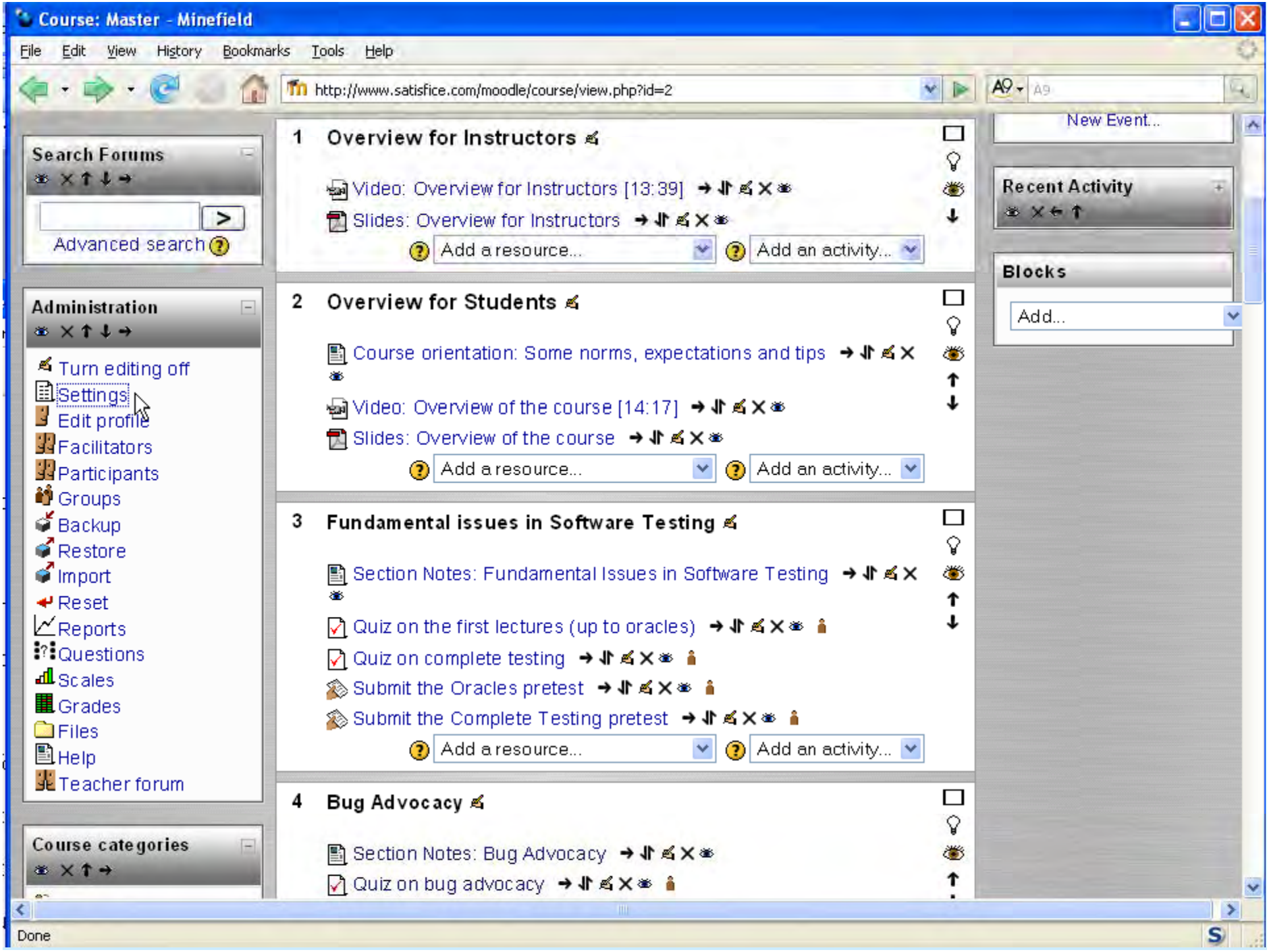

BBST-000: Edit categories - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/question/category.php?id=2
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000 » Edit categories
Questions Categories Import Export
Add category
Parent Category Category info Publish Action
Top
No
Add
Edit categories
Category Category info Questions Publish Delete Order
Top-level dummy category Don't put any questions in this category 0 ---
CompleteTesting 9 To
DomainTesting 0 To
Exploratory testing 1 To
FunctionTesting 0 To
Heuristics 1 To
High volume automated testing 1 To
Mission / strategy (advanced) These test material from the last half of the course 0 To
Done

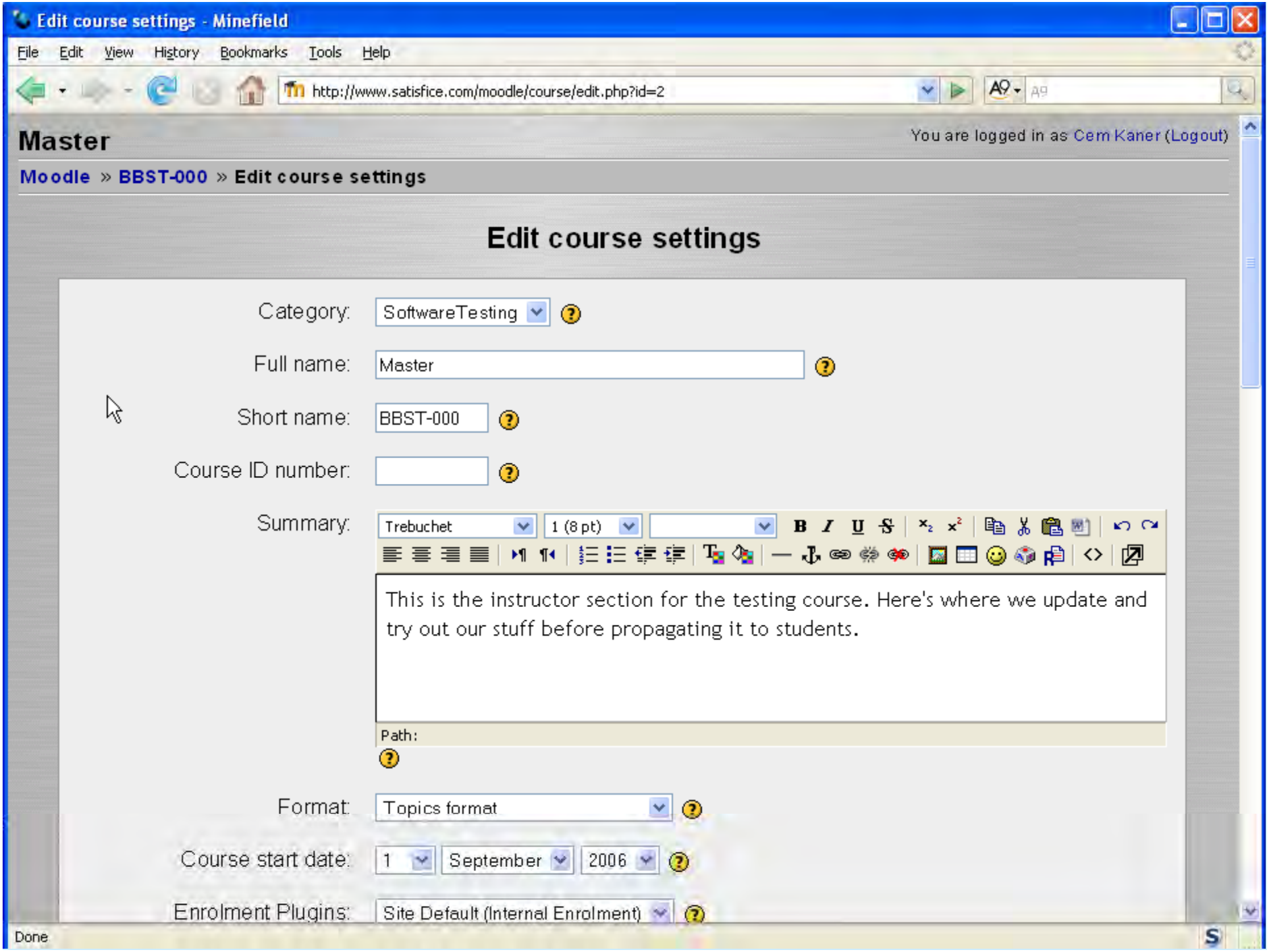

BBST-000: Edit questions - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/question/edit.php?courseid=2&cat=4
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000 » Edit questions
Questions
Categories
Import
Export
Category: CompleteTesting
Edit categories
Display questions from sub-categories too
Also show old questions
Create new question: Choose...
Import questions from file | Export questions to file
Action Question name Sort by type, name Type
[COMPLETE-001] Statement and branch coverage
[COMPLETE-002] Statement coverage
[COMPLETE-003] Complete testing definition
[COMPLETE-004] Defect arrival rate
[COMPLETE-005] Complete statement coverage
[COMPLETE-006] Measurement dysfunction
[COMPLETE-007] Statistical reliability model
[COMPLETE-008] Consequences of the impossibility of complete testing
[COMPLETE-009] MASPAR
Select all / Deselect all
With selected:
Delete
Move to >>
CompleteTesting
Done

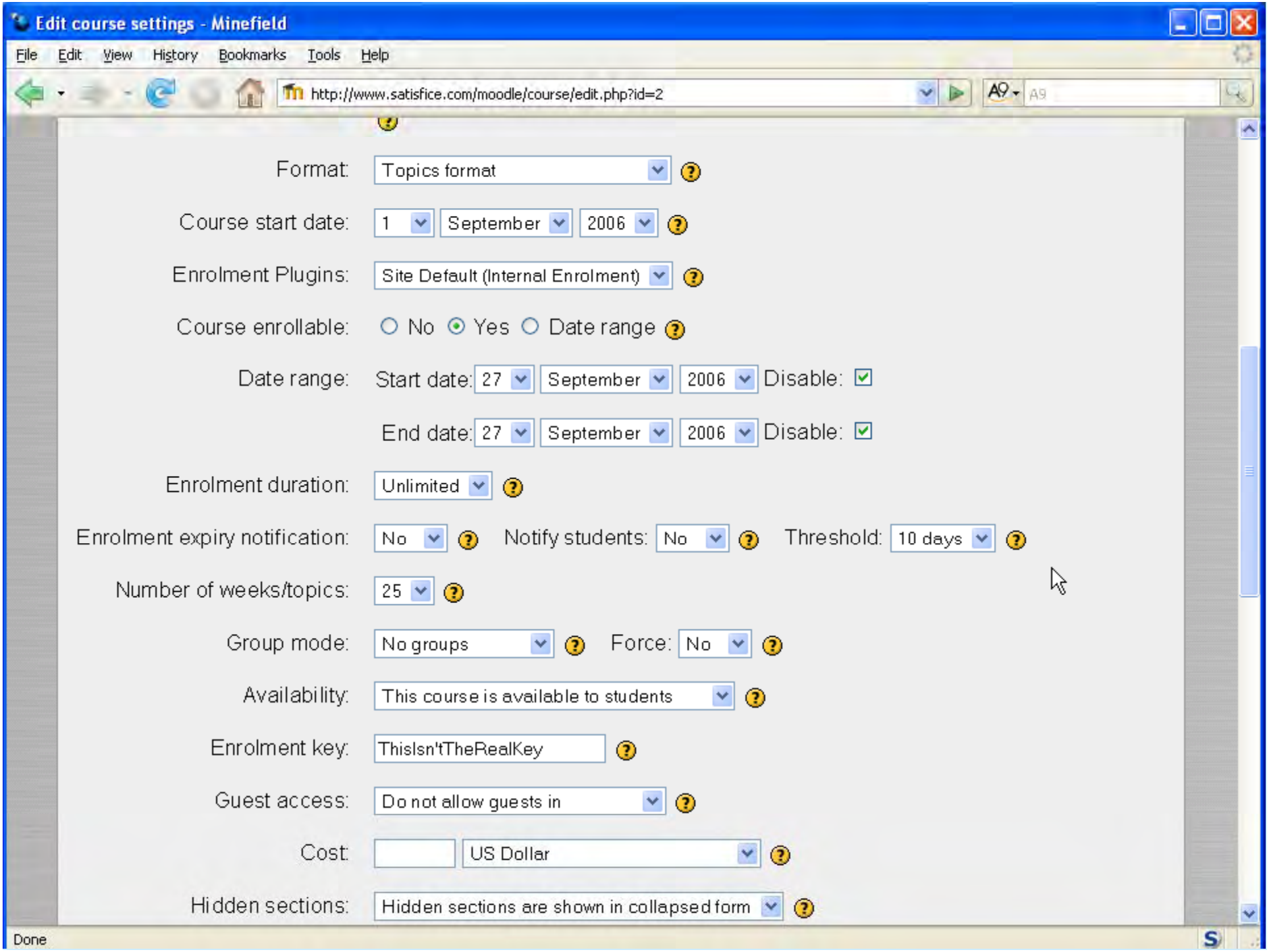

BBST-000: Edit questions - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/question/edit.php?courseid=2&cat=4
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000 » Edit questions
Questions
Categories
Import
Export
Category: CompleteTesting
Edit categories
Display questions from sub-categories too
Also show old questions
Create new question: Choose...
Import questions from file | Export questions to file
Action Question name Sort by type, name Type
[COMPLETE-001] Statement and branch coverage
[COMPLETE-002] Statement coverage
[COMPLETE-003] Complete testing definition
[COMPLETE-004] Defect arrival rate
[COMPLETE-005] Complete statement coverage
[COMPLETE-006] Measurement dysfunction
[COMPLETE-007] Statistical reliability model
[COMPLETE-008] Consequences of the impossibility of complete testing
[COMPLETE-009] MASPAR
Edit
Select all / Deselect all
With selected:
Delete
Move to >>
CompleteTesting
http://www.satisfice.com/moodle/question/question.php?id=31


BBST-000: Editing a question - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/question/question.php?id=31
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000 » Edit questions » Editing a question
Editing a Multiple Choice question
Category: CompleteTesting
Question name: [COMPLETE-001] Statement and branch co
Question:
Trebuchet
1 (8 pt)
About the HTML editor
Complete statement and branch coverage means ...
Path:
Image to display: No images have been uploaded to your course yet
Default question grade: 1
Penalty factor: 0
One or multiple answers?: One answer only
Done


BBST-000: Editing a question - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/question/question.php?id=31
Image to display: No images have been uploaded to your course yet
Default question grade: 1
Penalty factor: 0
One or multiple answers?: One answer only
Shuffle answers: Yes
Available choices: You must fill out at least two choices. Choices left blank will not be used.
Choice 1: That you have tested every statement in the program
Grade: None
Feedback: That's statement coverage
Choice 2: That you have tested every statement and every branc
Grade: 100 %
Feedback:
Choice 3: That you have tested every IF statement in the program
Grade: None
Feedback: You have, but you've tested every other statement too.
Choice 4: That you have tested every combination of values of IF
Grade: None
Done


Course: Master - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=2
Search Forums
Advanced search
Administration
Turn editing off
Settings
Edit profile
Facilitators
Participants
Groups
Backup
Restore
Import
Reset
Reports
Questions
Scales
Grades
Files
Help
Teacher forum
Course categories
1 Overview for Instructors
Video: Overview for Instructors [13:39]
Slides: Overview for Instructors
Add a resource...
Add an activity...
2 Overview for Students
Course orientation: Some norms, expectations and tips
Video: Overview of the course [14:17]
Slides: Overview of the course
Add a resource...
Add an activity...
3 Fundamental issues in Software Testing
Section Notes: Fundamental Issues in Software Testing
Quiz on the first lectures (up to oracles)
Quiz on complete testing
Submit the Oracles pretest
Submit the Complete Testing pretest
Add a resource...
Add an activity...
4 Bug Advocacy
Section Notes: Bug Advocacy
Quiz on bug advocacy
New Event...
Recent Activity
Blocks
Add...
Done


Edit course settings - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/edit.php?id=2
Master
You are logged in as Cem Kaner (Logout)
Moodle » BBST-000 » Edit course settings
Edit course settings
Category: SoftwareTesting
Full name: Master
Short name: BBST-000
Course ID number:
Summary: Trebuchet 1 (8 pt)
This is the instructor section for the testing course. Here's where we update and try out our stuff before propagating it to students.
Path:
Format: Topics format
Course start date: 1 September 2006
Enrolment Plugins: Site Default (Internal Enrolment)
Done


Edit course settings - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/edit.php?id=2
Format: Topics format
Course start date: 1 September 2006
Enrolment Plugins: Site Default (Internal Enrolment)
Course enrollable: No Yes Date range
Date range: Start date: 27 September 2006 Disable:
End date: 27 September 2006 Disable:
Enrolment duration: Unlimited
Enrolment expiry notification: No Notify students: No Threshold: 10 days
Number of weeks/topics: 25
Group mode: No groups Force: No
Availability: This course is available to students
Enrolment key: ThisIsn'tTheRealKey
Guest access: Do not allow guests in
Cost: US Dollar
Hidden sections: Hidden sections are shown in collapsed form
Done

# Course tour (continued)

1. Tour of the Moodle course management system
2. Tour of the Black Box Software Testing Course on Moodle

# How the Course Works

Students watch the video before coming to class

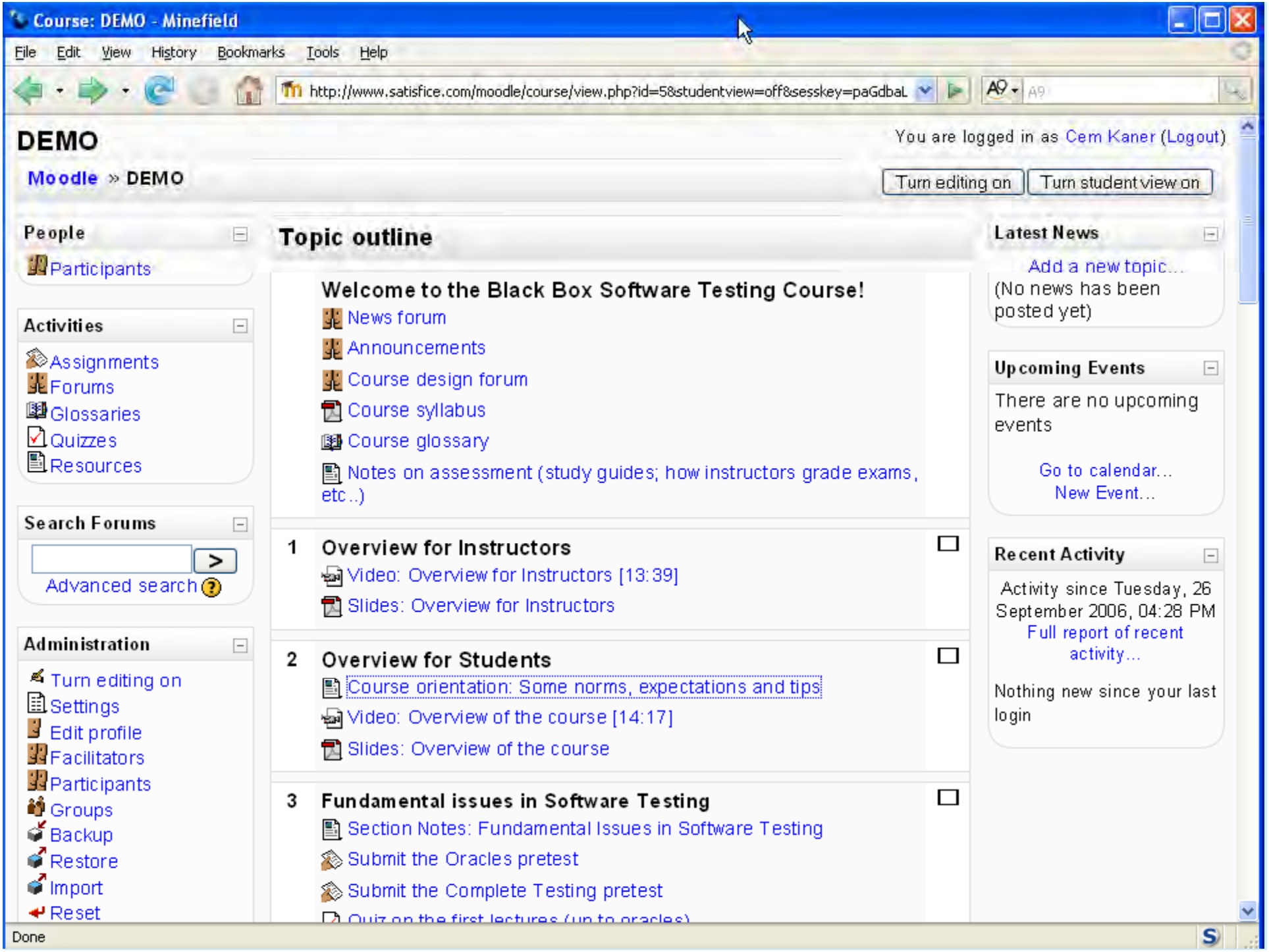
Students often work through an open-book quiz before coming to class

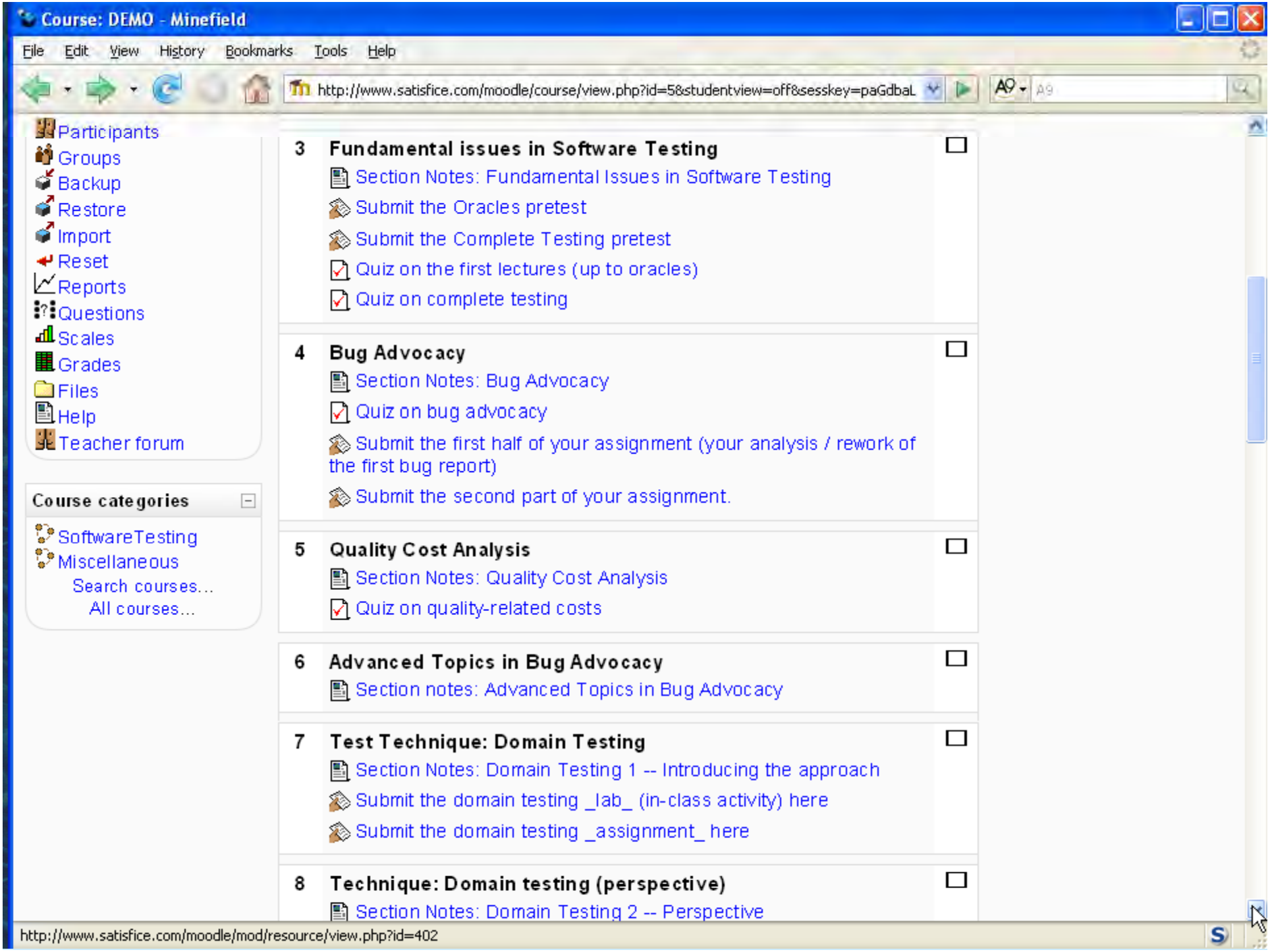
We spend classroom time on

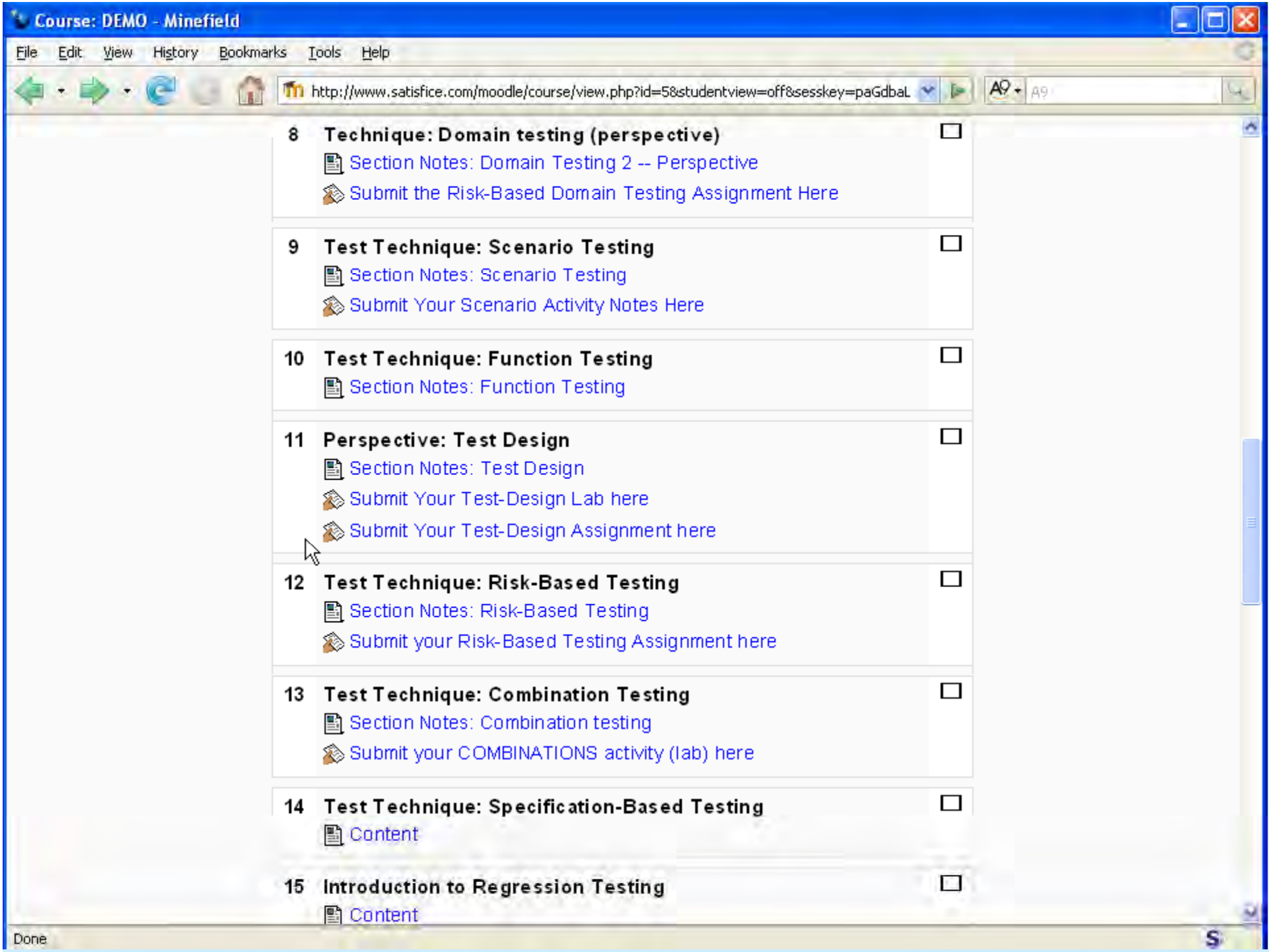
- coached activities
- facilitated discussions
- group feedback (lecture) when I see a class-wide problem

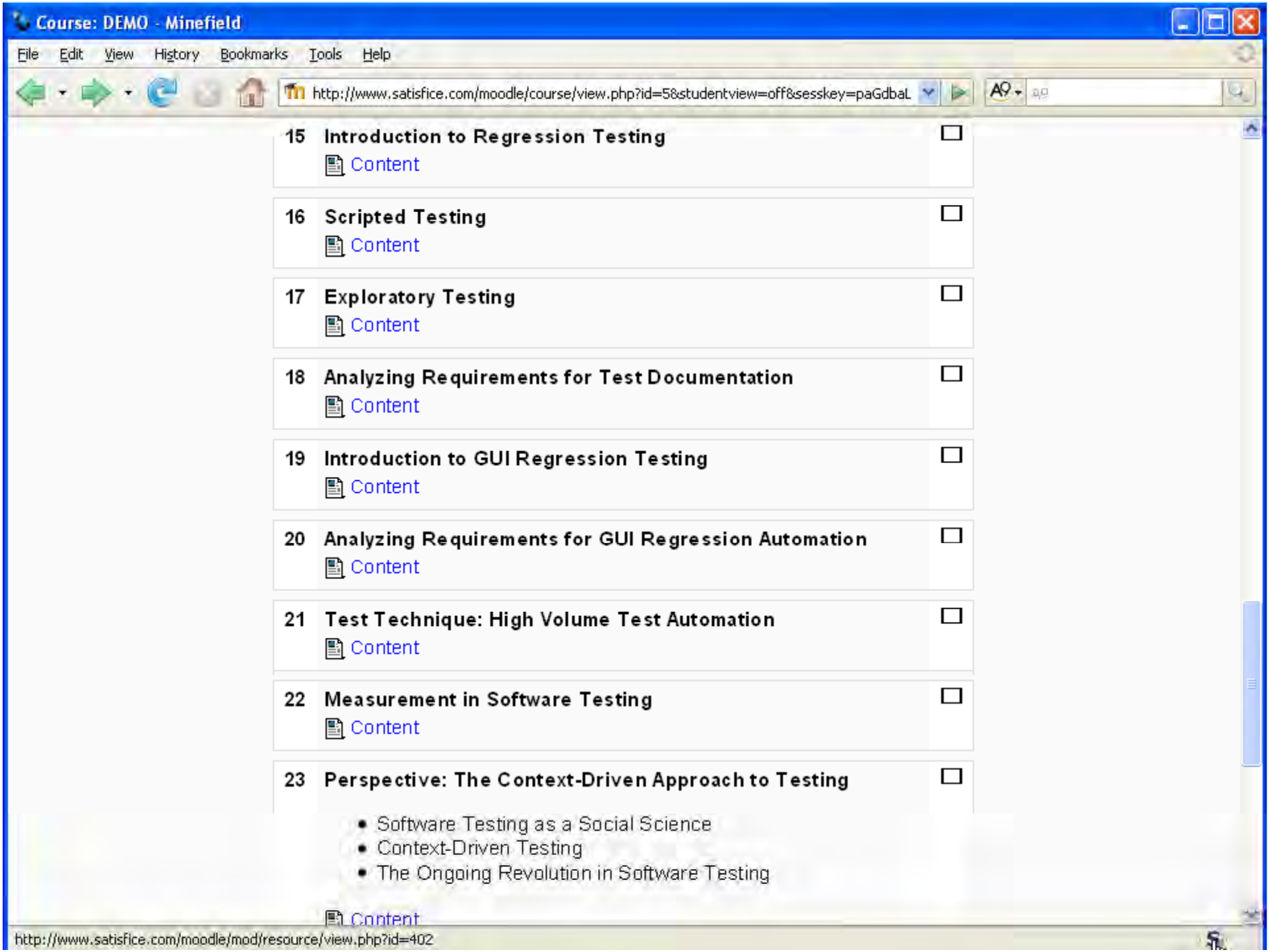
We apply the material in

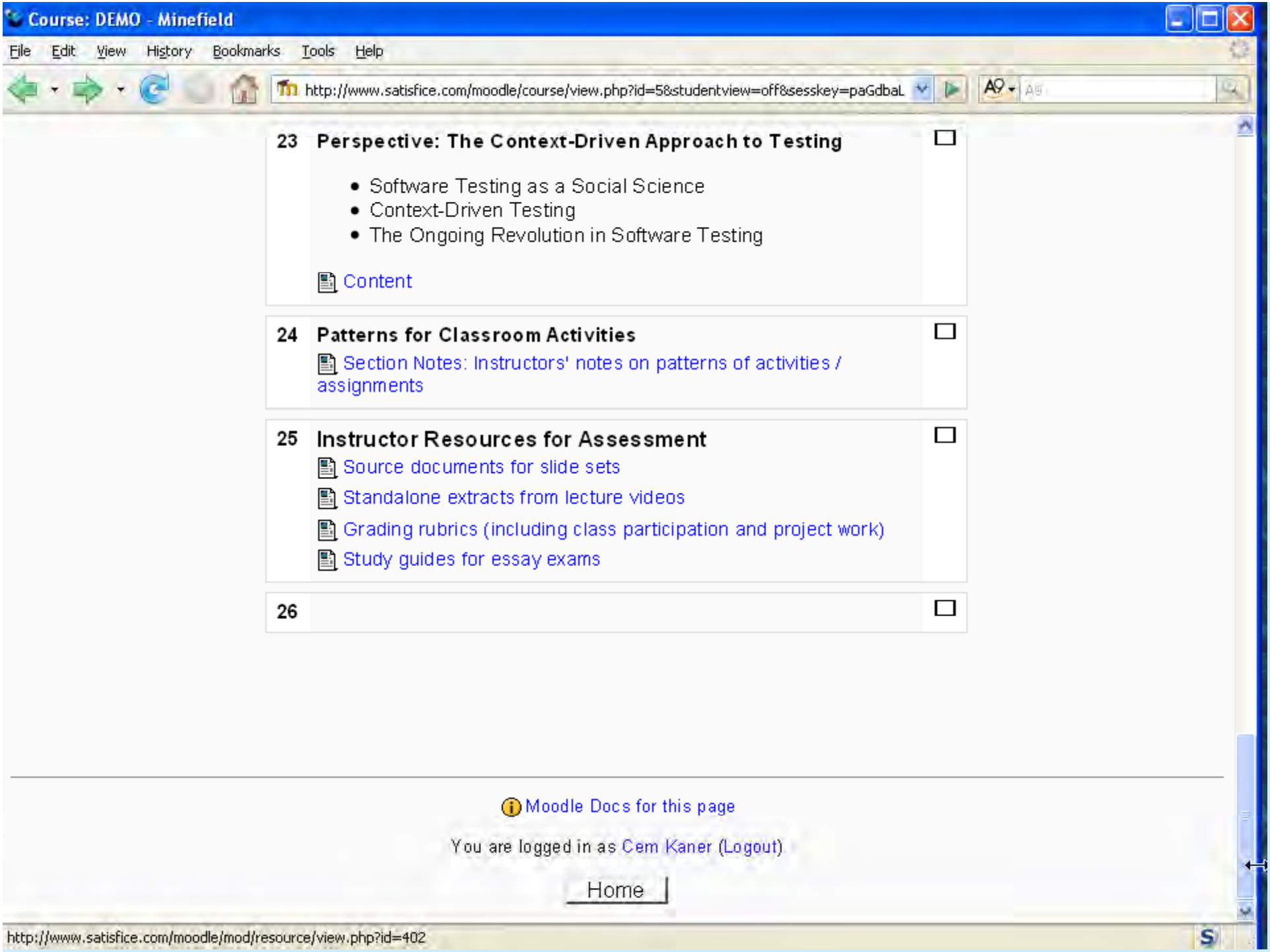
- in-class activities
- out-of-class assignments

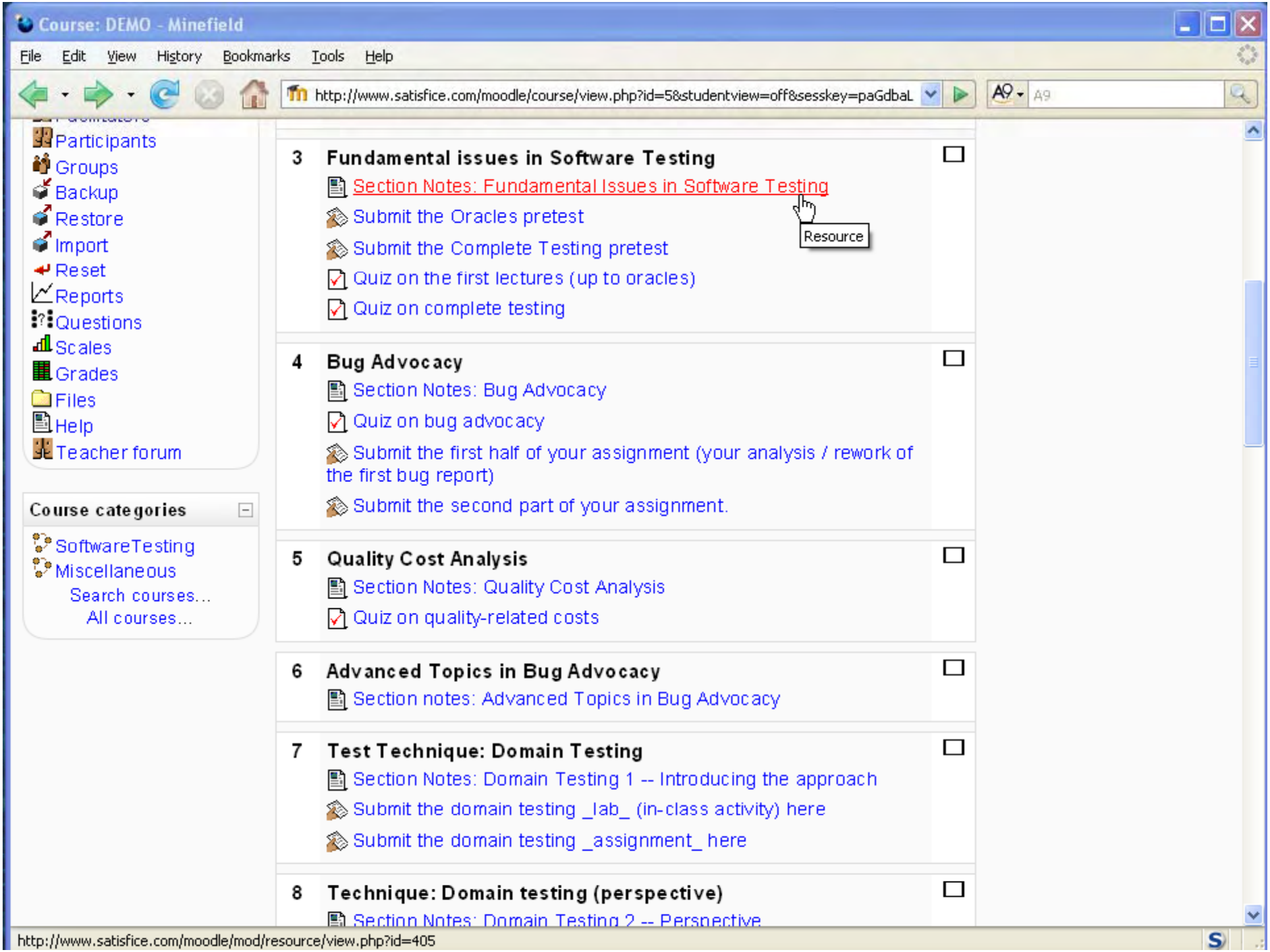

Course: DEMO - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=5&studentview=off&sesskey=paGdbaL
DEMO
You are logged in as Cem Kaner (Logout)
Moodle » DEMO
Turn editing on
Turn student view on
People
Participants
Activities
Assignments
Forums
Glossaries
Quizzes
Resources
Search Forums
Advanced search
Administration
Turn editing on
Settings
Edit profile
Facilitators
Participants
Groups
Backup
Restore
Import
Reset
Topic outline
Welcome to the Black Box Software Testing Course!
News forum
Announcements
Course design forum
Course syllabus
Course glossary
Notes on assessment (study guides; how instructors grade exams, etc..)
1 Overview for Instructors
Video: Overview for Instructors [13:39]
Slides: Overview for Instructors
2 Overview for Students
Course orientation: Some norms, expectations and tips
Video: Overview of the course [14:17]
Slides: Overview of the course
3 Fundamental issues in Software Testing
Section Notes: Fundamental Issues in Software Testing
Submit the Oracles pretest
Submit the Complete Testing pretest
Latest News
Add a new topic...
(No news has been posted yet)
Upcoming Events
There are no upcoming events
Go to calendar...
New Event...
Recent Activity
Activity since Tuesday, 26 September 2006, 04:28 PM
Full report of recent activity...
Nothing new since your last login
Done

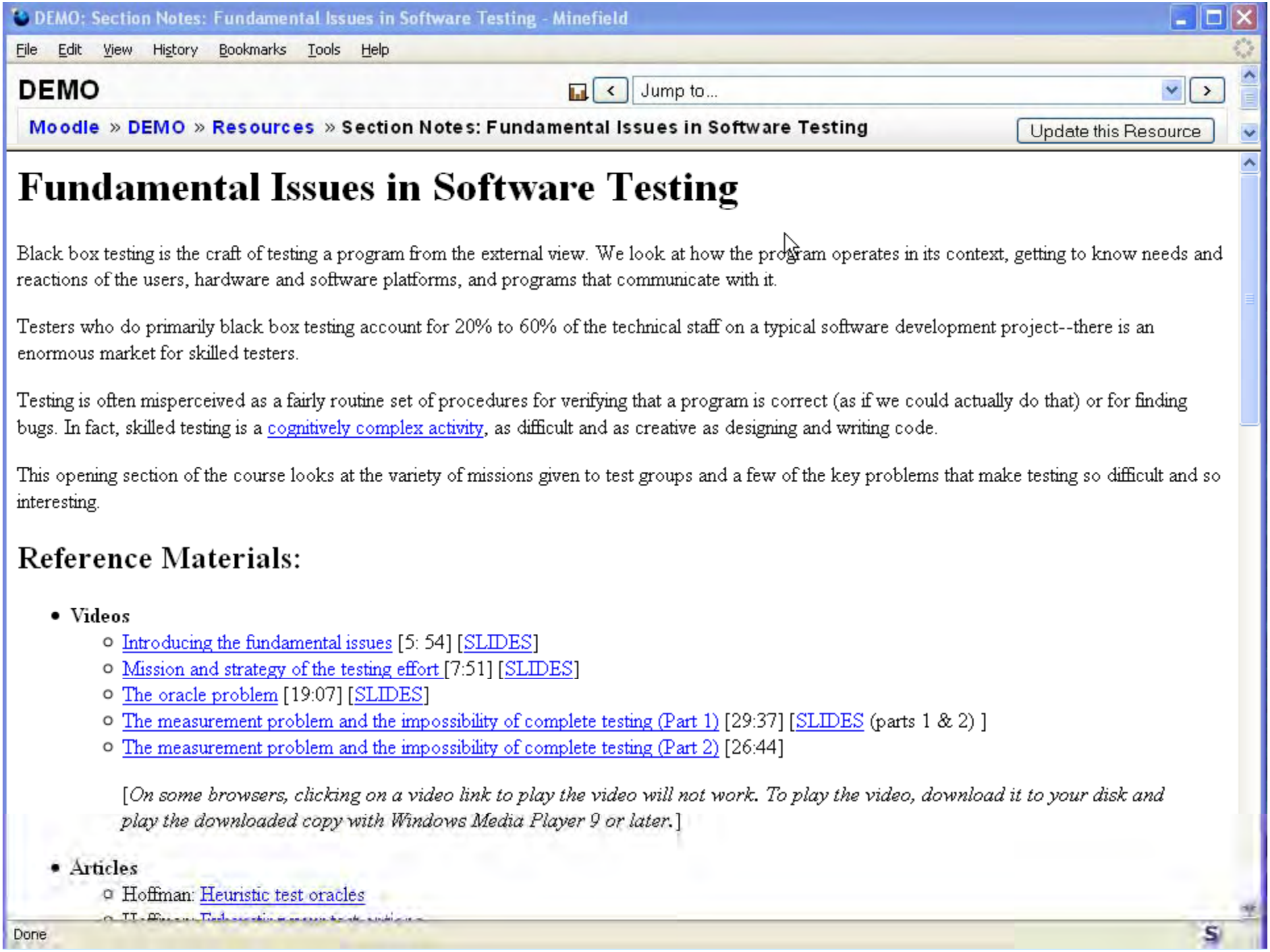

Course: DEMO - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=5&studentview=off&sesskey=paGdbaL
Participants
Groups
Backup
Restore
Import
Reset
Reports
Questions
Scales
Grades
Files
Help
Teacher forum
Course categories
SoftwareTesting
Miscellaneous
Search courses...
All courses...
3 Fundamental issues in Software Testing
Section Notes: Fundamental Issues in Software Testing
Submit the Oracles pretest
Submit the Complete Testing pretest
Quiz on the first lectures (up to oracles)
Quiz on complete testing
4 Bug Advocacy
Section Notes: Bug Advocacy
Quiz on bug advocacy
Submit the first half of your assignment (your analysis / rework of the first bug report)
Submit the second part of your assignment.
5 Quality Cost Analysis
Section Notes: Quality Cost Analysis
Quiz on quality-related costs
6 Advanced Topics in Bug Advocacy
Section notes: Advanced Topics in Bug Advocacy
7 Test Technique: Domain Testing
Section Notes: Domain Testing 1 -- Introducing the approach
Submit the domain testing _lab_ (in-class activity) here
Submit the domain testing _assignment_ here
8 Technique: Domain testing (perspective)
Section Notes: Domain Testing 2 -- Perspective
http://www.satisfice.com/moodle/mod/resource/view.php?id=402

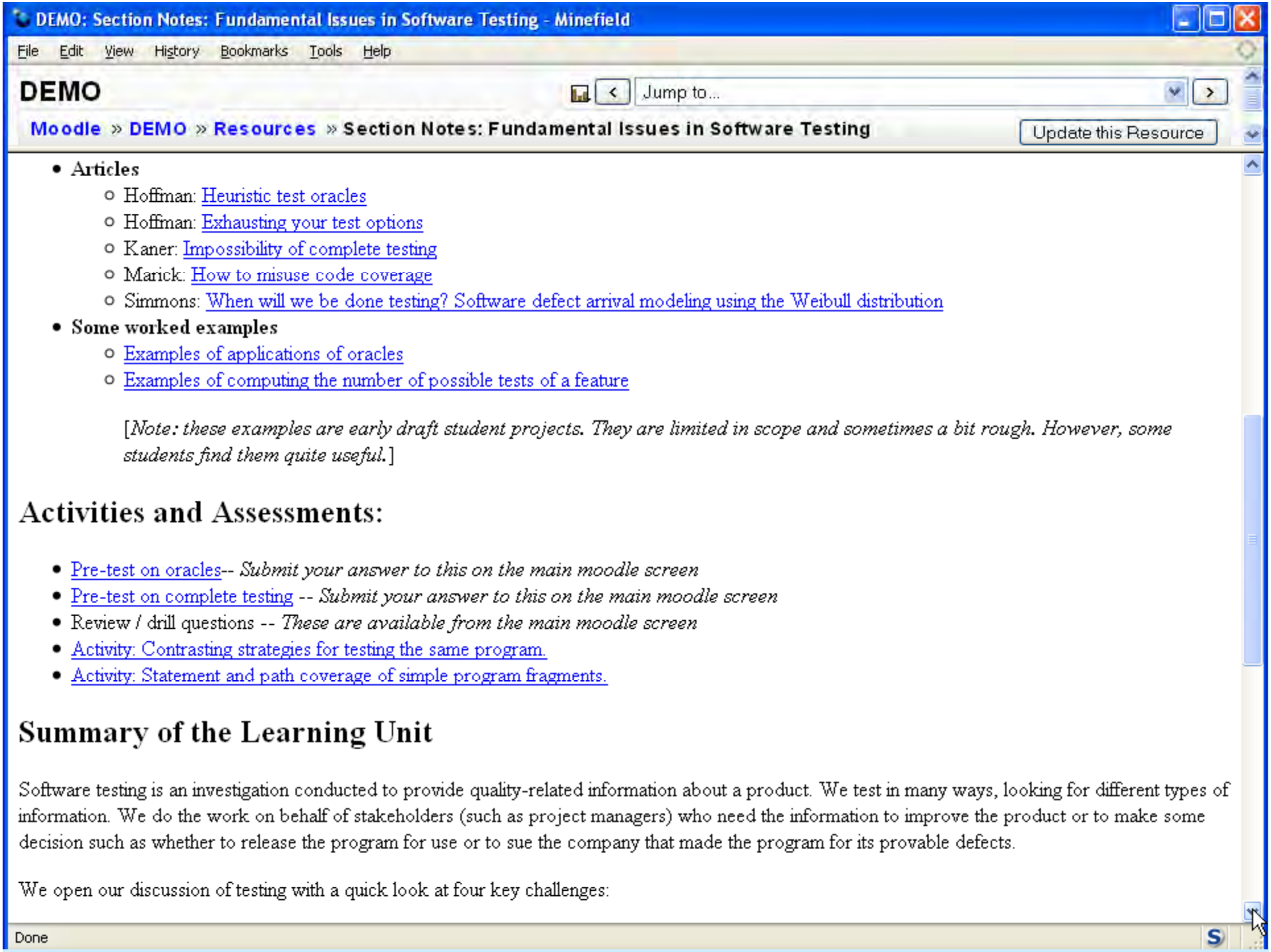

Course: DEMO - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=5&studentview=off&sesskey=paGdbaL
8 Technique: Domain testing (perspective)
Section Notes: Domain Testing 2 -- Perspective
Submit the Risk-Based Domain Testing Assignment Here
9 Test Technique: Scenario Testing
Section Notes: Scenario Testing
Submit Your Scenario Activity Notes Here
10 Test Technique: Function Testing
Section Notes: Function Testing
11 Perspective: Test Design
Section Notes: Test Design
Submit Your Test-Design Lab here
Submit Your Test-Design Assignment here
12 Test Technique: Risk-Based Testing
Section Notes: Risk-Based Testing
Submit your Risk-Based Testing Assignment here
13 Test Technique: Combination Testing
Section Notes: Combination testing
Submit your COMBINATIONS activity (lab) here
14 Test Technique: Specification-Based Testing
Content
15 Introduction to Regression Testing
Content
Done

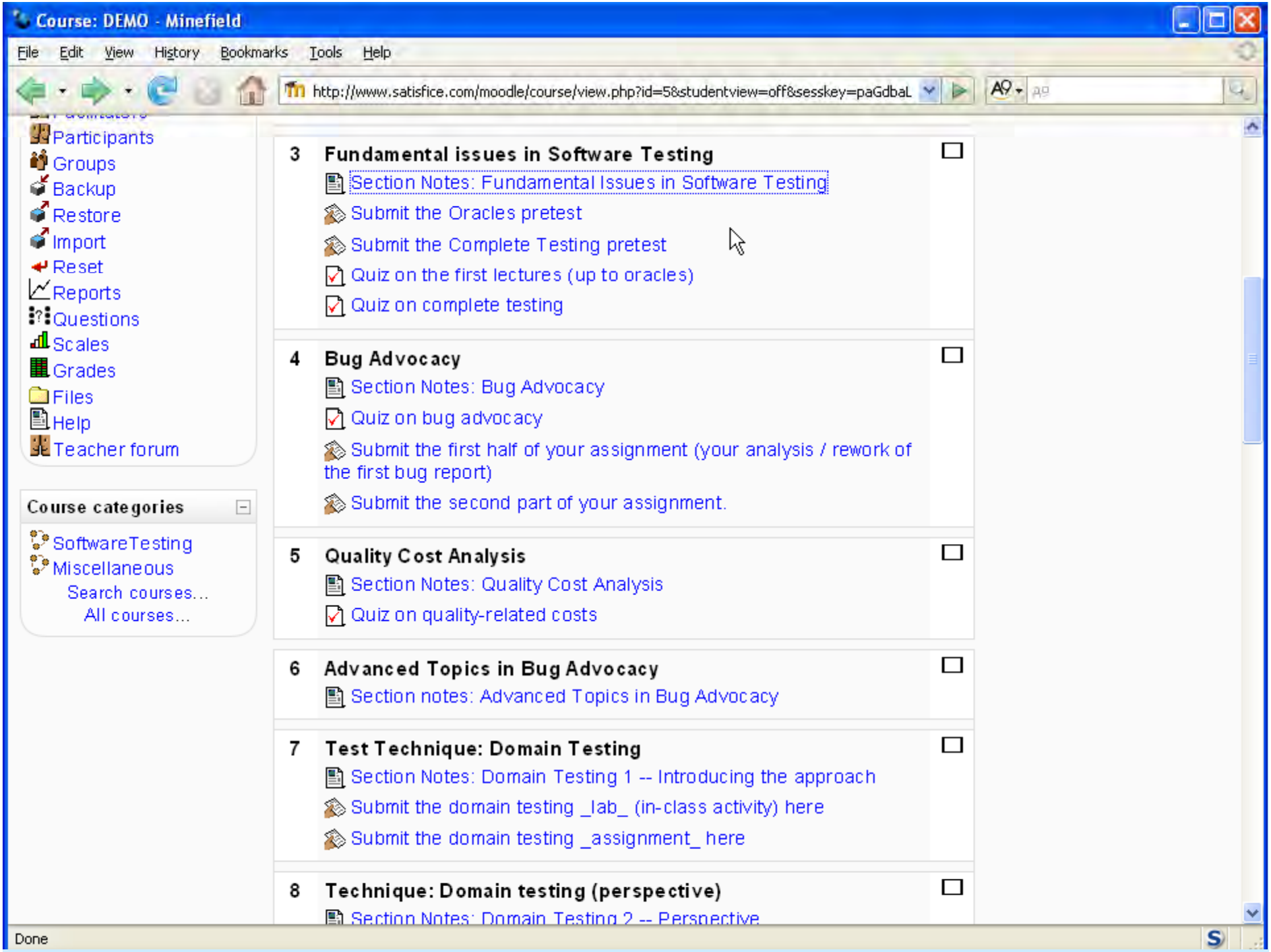

Course: DEMO - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=5&studentview=off&sesskey=paGdbaL
15 Introduction to Regression Testing
Content
16 Scripted Testing
Content
17 Exploratory Testing
Content
18 Analyzing Requirements for Test Documentation
Content
19 Introduction to GUI Regression Testing
Content
20 Analyzing Requirements for GUI Regression Automation
Content
21 Test Technique: High Volume Test Automation
Content
22 Measurement in Software Testing
Content
23 Perspective: The Context-Driven Approach to Testing
Software Testing as a Social Science
Context-Driven Testing
The Ongoing Revolution in Software Testing
Content
http://www.satisfice.com/moodle/mod/resource/view.php?id=402

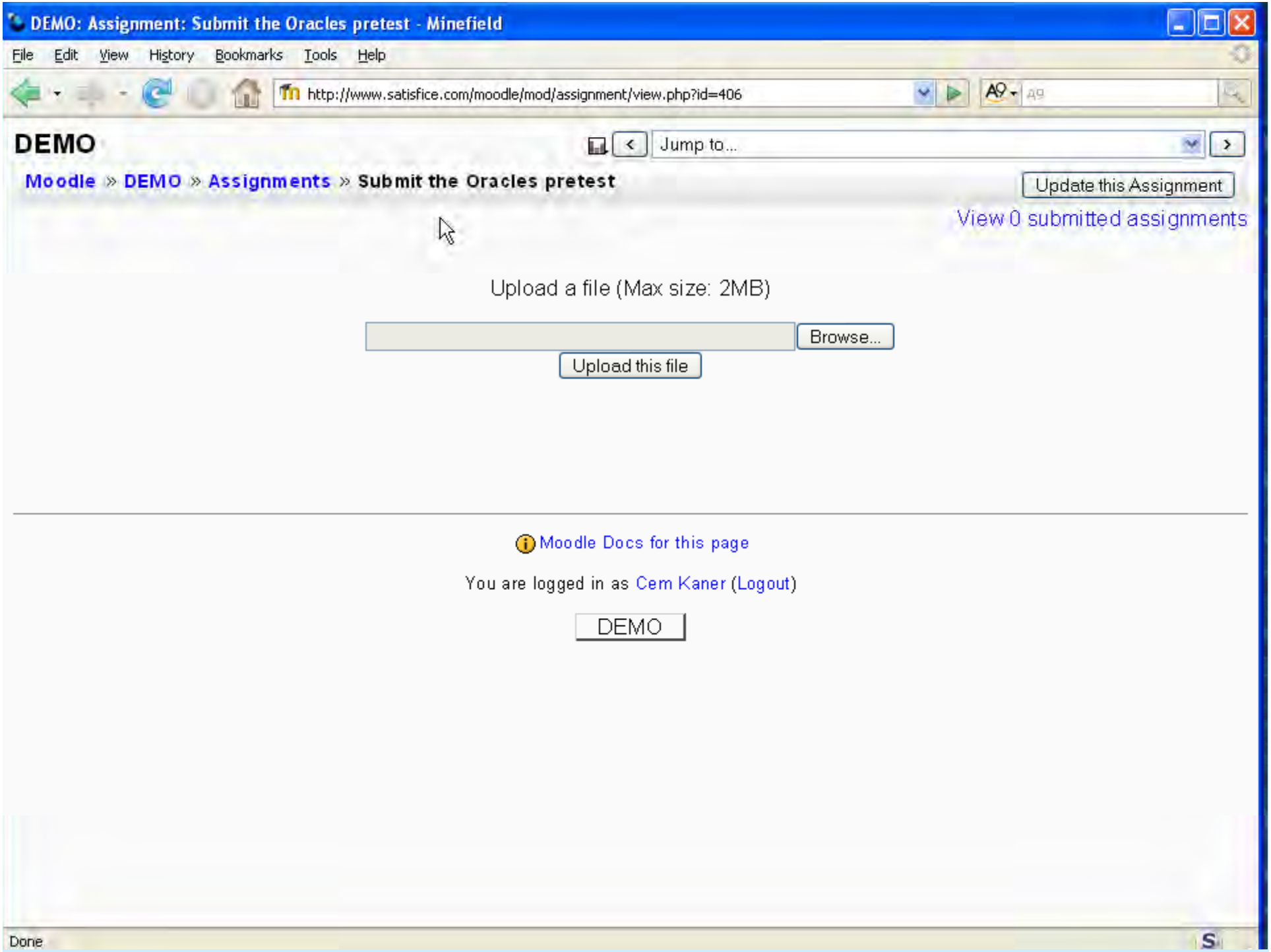

Course: DEMO - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=5&studentview=off&sesskey=paGdbaL
23 Perspective: The Context-Driven Approach to Testing
Software Testing as a Social Science
Context-Driven Testing
The Ongoing Revolution in Software Testing
Content
24 Patterns for Classroom Activities
Section Notes: Instructors' notes on patterns of activities / assignments
25 Instructor Resources for Assessment
Source documents for slide sets
Standalone extracts from lecture videos
Grading rubrics (including class participation and project work)
Study guides for essay exams
26
Moodle Docs for this page
You are logged in as Cem Kaner (Logout)
Home
http://www.satisfice.com/moodle/mod/resource/view.php?id=402

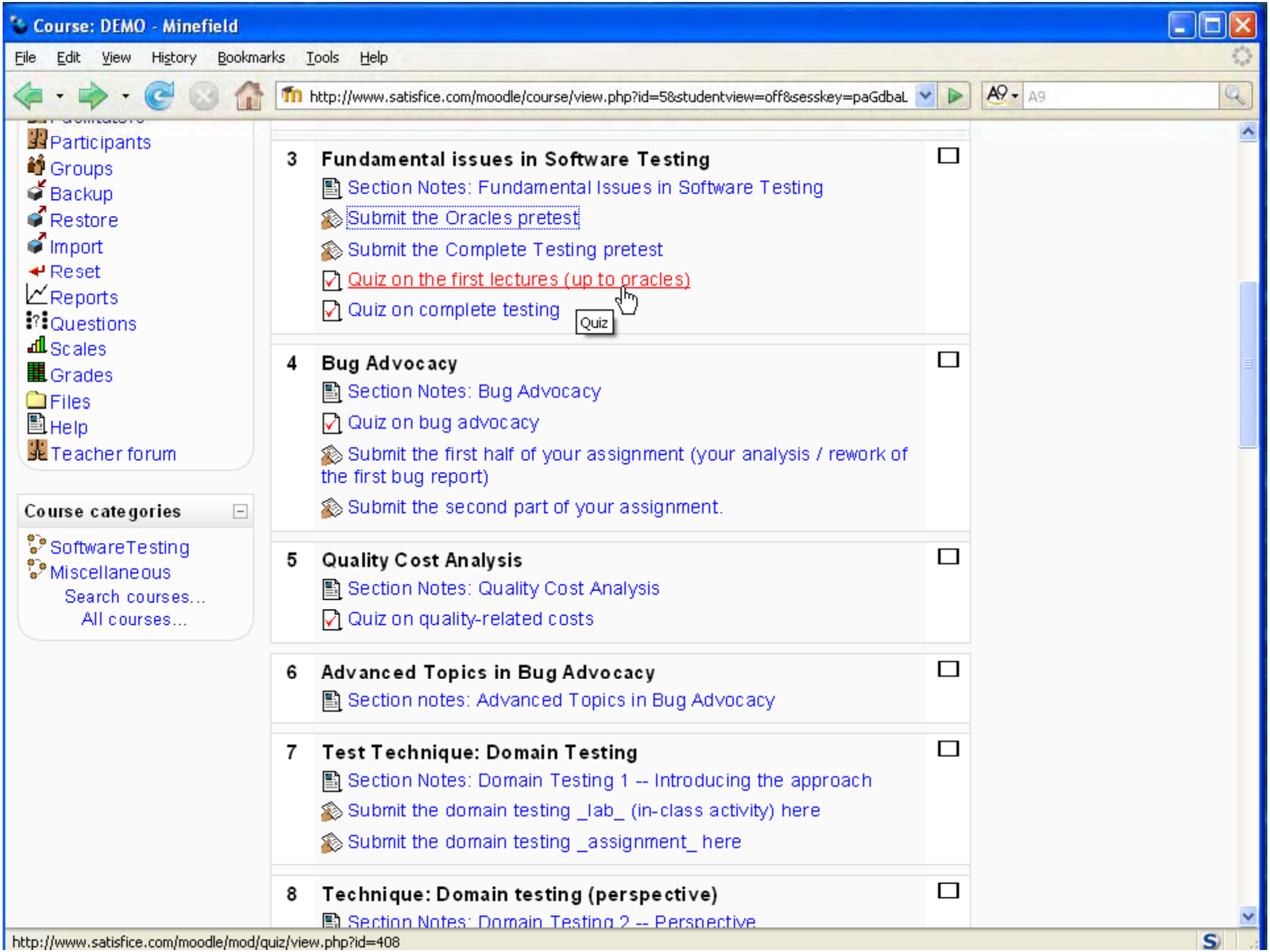

Course: DEMO - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=5&studentview=off&sesskey=paGdbaL
Participants
Groups
Backup
Restore
Import
Reset
Reports
Questions
Scales
Grades
Files
Help
Teacher forum
Course categories
SoftwareTesting
Miscellaneous
Search courses...
All courses...
3 Fundamental issues in Software Testing
Section Notes: Fundamental Issues in Software Testing
Submit the Oracles pretest
Submit the Complete Testing pretest
Quiz on the first lectures (up to oracles)
Quiz on complete testing
Resource
4 Bug Advocacy
Section Notes: Bug Advocacy
Quiz on bug advocacy
Submit the first half of your assignment (your analysis / rework of the first bug report)
Submit the second part of your assignment.
5 Quality Cost Analysis
Section Notes: Quality Cost Analysis
Quiz on quality-related costs
6 Advanced Topics in Bug Advocacy
Section notes: Advanced Topics in Bug Advocacy
7 Test Technique: Domain Testing
Section Notes: Domain Testing 1 -- Introducing the approach
Submit the domain testing _lab_ (in-class activity) here
Submit the domain testing _assignment_ here
8 Technique: Domain testing (perspective)
Section Notes: Domain Testing 2 -- Perspective
http://www.satisfice.com/moodle/mod/resource/view.php?id=405


DEMO: Section Notes: Fundamental Issues in Software Testing - Minefield
File Edit View History Bookmarks Tools Help
DEMO
Jump to...
Moodle » DEMO » Resources » Section Notes: Fundamental Issues in Software Testing
Update this Resource
Fundamental Issues in Software Testing
Black box testing is the craft of testing a program from the external view. We look at how the program operates in its context, getting to know needs and reactions of the users, hardware and software platforms, and programs that communicate with it.
Testers who do primarily black box testing account for 20% to 60% of the technical staff on a typical software development project--there is an enormous market for skilled testers.
Testing is often misperceived as a fairly routine set of procedures for verifying that a program is correct (as if we could actually do that) or for finding bugs. In fact, skilled testing is a cognitively complex activity, as difficult and as creative as designing and writing code.
This opening section of the course looks at the variety of missions given to test groups and a few of the key problems that make testing so difficult and so interesting.
Reference Materials:
Videos
Introducing the fundamental issues [5: 54] [SLIDES]
Mission and strategy of the testing effort [7:51] [SLIDES]
The oracle problem [19:07] [SLIDES]
The measurement problem and the impossibility of complete testing (Part 1) [29:37] [SLIDES (parts 1 & 2) ]
The measurement problem and the impossibility of complete testing (Part 2) [26:44]
[On some browsers, clicking on a video link to play the video will not work. To play the video, download it to your disk and play the downloaded copy with Windows Media Player 9 or later.]
Articles
Hoffman: Heuristic test oracles
Done


DEMO: Section Notes: Fundamental Issues in Software Testing - Minefield
File Edit View History Bookmarks Tools Help
DEMO
Jump to...
Moodle » DEMO » Resources » Section Notes: Fundamental Issues in Software Testing
Update this Resource
Articles
Hoffman: Heuristic test oracles
Hoffman: Exhausting your test options
Kaner: Impossibility of complete testing
Marick: How to misuse code coverage
Simmons: When will we be done testing? Software defect arrival modeling using the Weibull distribution
Some worked examples
Examples of applications of oracles
Examples of computing the number of possible tests of a feature
[Note: these examples are early draft student projects. They are limited in scope and sometimes a bit rough. However, some students find them quite useful.]
Activities and Assessments:
Pre-test on oracles-- Submit your answer to this on the main moodle screen
Pre-test on complete testing -- Submit your answer to this on the main moodle screen
Review / drill questions -- These are available from the main moodle screen
Activity: Contrasting strategies for testing the same program.
Activity: Statement and path coverage of simple program fragments.
Summary of the Learning Unit
Software testing is an investigation conducted to provide quality-related information about a product. We test in many ways, looking for different types of information. We do the work on behalf of stakeholders (such as project managers) who need the information to improve the product or to make some decision such as whether to release the program for use or to sue the company that made the program for its provable defects.
We open our discussion of testing with a quick look at four key challenges:
Done


Course: DEMO - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=5&studentview=off&sesskey=paGdbaL
Participants
Groups
Backup
Restore
Import
Reset
Reports
Questions
Scales
Grades
Files
Help
Teacher forum
Course categories
SoftwareTesting
Miscellaneous
Search courses...
All courses...
3 Fundamental issues in Software Testing
Section Notes: Fundamental Issues in Software Testing
Submit the Oracles pretest
Submit the Complete Testing pretest
Quiz on the first lectures (up to oracles)
Quiz on complete testing
4 Bug Advocacy
Section Notes: Bug Advocacy
Quiz on bug advocacy
Submit the first half of your assignment (your analysis / rework of the first bug report)
Submit the second part of your assignment.
5 Quality Cost Analysis
Section Notes: Quality Cost Analysis
Quiz on quality-related costs
6 Advanced Topics in Bug Advocacy
Section notes: Advanced Topics in Bug Advocacy
7 Test Technique: Domain Testing
Section Notes: Domain Testing 1 -- Introducing the approach
Submit the domain testing _lab_ (in-class activity) here
Submit the domain testing _assignment_ here
8 Technique: Domain testing (perspective)
Section Notes: Domain Testing 2 -- Perspective
Done


DEMO: Assignment: Submit the Oracles pretest - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/mod/assignment/view.php?id=406
DEMO
Jump to...
Moodle » DEMO » Assignments » Submit the Oracles pretest
Update this Assignment
View 0 submitted assignments
Upload a file (Max size: 2MB)
Browse...
Upload this file
Moodle Docs for this page
You are logged in as Cem Kaner (Logout)
DEMO
Done


Course: DEMO - Minefield
File Edit View History Bookmarks Tools Help
http://www.satisfice.com/moodle/course/view.php?id=5&studentview=off&sesskey=paGdbaL
Participants
Groups
Backup
Restore
Import
Reset
Reports
Questions
Scales
Grades
Files
Help
Teacher forum
Course categories
SoftwareTesting
Miscellaneous
Search courses...
All courses...
3 Fundamental issues in Software Testing
Section Notes: Fundamental Issues in Software Testing
Submit the Oracles pretest
Submit the Complete Testing pretest
Quiz on the first lectures (up to oracles)
Quiz on complete testing
Quiz
4 Bug Advocacy
Section Notes: Bug Advocacy
Quiz on bug advocacy
Submit the first half of your assignment (your analysis / rework of the first bug report)
Submit the second part of your assignment.
5 Quality Cost Analysis
Section Notes: Quality Cost Analysis
Quiz on quality-related costs
6 Advanced Topics in Bug Advocacy
Section notes: Advanced Topics in Bug Advocacy
7 Test Technique: Domain Testing
Section Notes: Domain Testing 1 -- Introducing the approach
Submit the domain testing _lab_ (in-class activity) here
Submit the domain testing _assignment_ here
8 Technique: Domain testing (perspective)
Section Notes: Domain Testing 2 -- Perspective
http://www.satisfice.com/moodle/mod/quiz/view.php?id=408

# SUCCESS FACTORS

# Success factors

1. Strong content
2. Story-based teaching
3. Detailed examples
4. Video lectures
5. In-class activities that tie to the lecture
6. Application to a real product under test
7. Orientation exercises
8. Open book quizzes
9. Study-guide based exam
10. Challenging but focused assignments
11. Task scaffolding
12. Peer review
13. Explicit discussions of learning issues in the course design.
14. Instruction on test-taking skills
15. SALG feedback
16. Open discussion of (employment) value of the material and the work
17. Organic evolution of the class (rather than process-constrained)
18. Enthusiasm and ongoing renewal (Hawthorne effect)

# Strong content

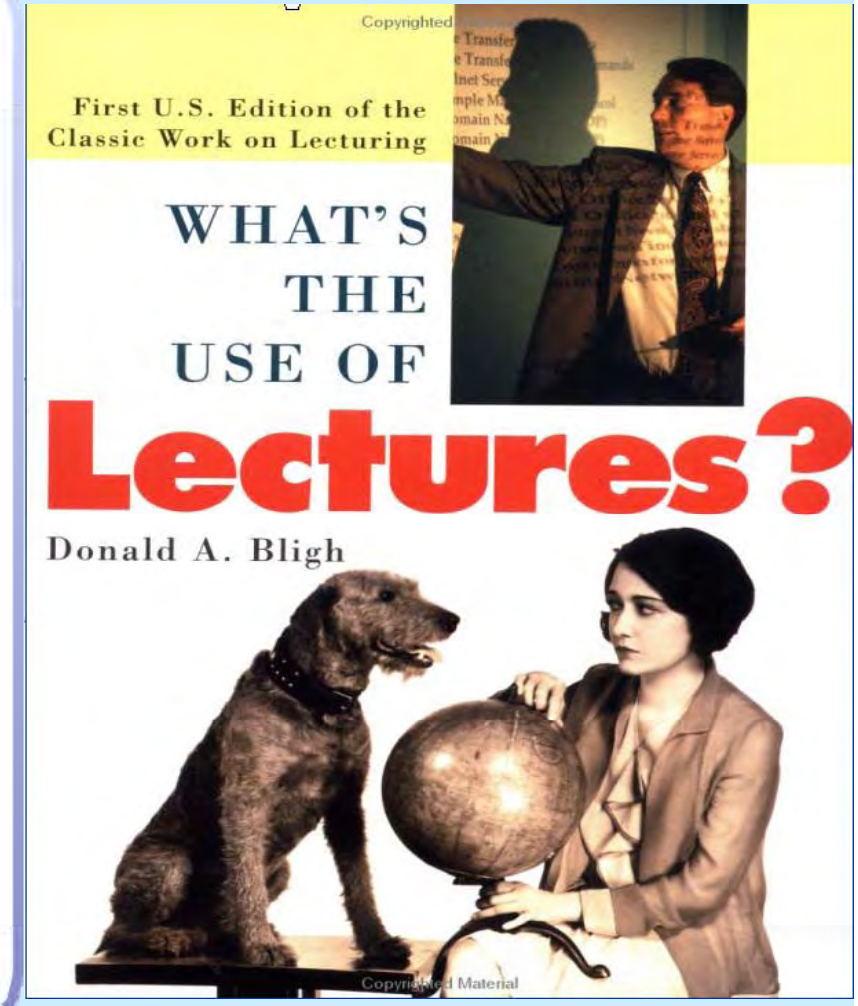
- Course development started in 1983
  - people were teaching QAI/DoD/SWEBOK-style testing rather than Silicon Valley style, totally counterproductive for my staff
  - Outcomes from 1983-1992 were not course notes, they were TCS 1st ed (1987) and 2nd ed (1993) (became best seller in the field)
- Commercial teaching 1993-2004
  - 100+ teachings
  - Extensive peer review (alpha/beta teachings, co-teaching, mergers with other courses)
- Academic 2000-2007

# Commercial Teaching Style

Primary communication style was lecture

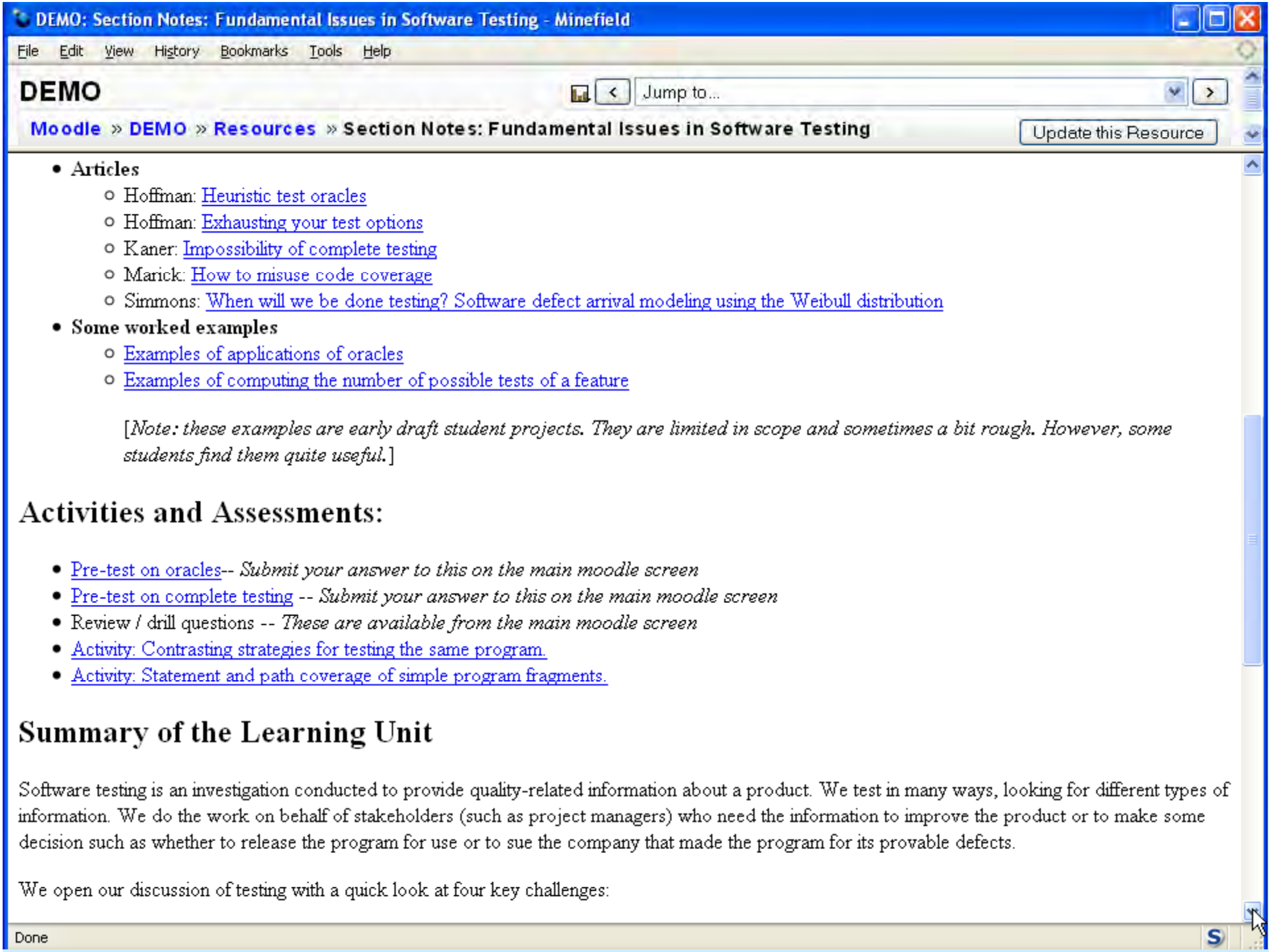
- Real-life examples
  - Motivating
  - Memorable
  - Illustrate applications
  - Illustrate complexity

Lectures can be excellent for conveying basic knowledge, but they are weak for developing higher order cognitive skills



# Detailed examples

- For "each" technique, we provide written examples (with screen shots) or video demonstrations that illustrate the application of the technique to a shipping product.
- Worked examples can be powerful teaching tools, especially when motivated by real-life situations. They are fundamental for some learning styles.
- The lasting popularity of problem books, such as the *Schaum's Outline series* and more complex texts like *Sveshnikov [148]* attests to the value of example-driven learning, at least for some learners.
- At this time, we don't have examples for every course section, and what we do have are variable in quality. However, several students have told us the examples that we do provide helped them learn. We intend to create more and better examples.


DEMO: Section Notes: Fundamental Issues in Software Testing - Minefield
File Edit View History Bookmarks Tools Help
DEMO
Jump to...
Moodle » DEMO » Resources » Section Notes: Fundamental Issues in Software Testing
Update this Resource
Articles
Hoffman: Heuristic test oracles
Hoffman: Exhausting your test options
Kaner: Impossibility of complete testing
Marick: How to misuse code coverage
Simmons: When will we be done testing? Software defect arrival modeling using the Weibull distribution
Some worked examples
Examples of applications of oracles
Examples of computing the number of possible tests of a feature
[Note: these examples are early draft student projects. They are limited in scope and sometimes a bit rough. However, some students find them quite useful.]
Activities and Assessments:
Pre-test on oracles-- Submit your answer to this on the main moodle screen
Pre-test on complete testing -- Submit your answer to this on the main moodle screen
Review / drill questions -- These are available from the main moodle screen
Activity: Contrasting strategies for testing the same program.
Activity: Statement and path coverage of simple program fragments.
Summary of the Learning Unit
Software testing is an investigation conducted to provide quality-related information about a product. We test in many ways, looking for different types of information. We do the work on behalf of stakeholders (such as project managers) who need the information to improve the product or to make some decision such as whether to release the program for use or to sue the company that made the program for its provable defects.
We open our discussion of testing with a quick look at four key challenges:
Done

# Story-based teaching

- Exemplars (prototypic cases or events) play an important role in the development and recollection of simple and complex concepts.
- Stories are like examples, but they probably also describe something that actually happened and include human-interest information that makes the situation more memorable and more motivating.
- Well chosen stories can enhance the credibility of the teacher with the students and can shape the attitudes of the students.
- Poorly chosen, poorly researched, dishonestly told, or poorly presented stories can kill the credibility of the teacher.

# Video lectures

- We created a variety of out-of-classroom activities, such as homework (with application to real products) and group study sessions
- Students praised the (by now, well polished) lectures
- But they often told us that they learned the most from the out-of-class activities
- In many cases, the most effective (our subjective assessment) student-and-instructor interactions happened out of class, such as discussions at the local cafe.
- So we turned the class inside out
  - **Lectures out** of the classroom
  - **Activities** (including discussion) **in** the classroom

# Video lectures

- *Stored lectures* are common in distance learning programs.
- Some students prefer live lectures but on average, students learn as well from video as live lecture.
- Students can replay videos which can help students whose first language is not English.
- Web-based lecture segments supplement some computer science courses. Studio-taped, rehearsed lectures with synchronously presented slides (like ours) have been done before.
- Many instructors tape live lectures, but some (including us) report their students prefer studio-produced lectures over recorded live lectures.
- We prefer studio-produced lectures because they have no unscripted interruptions and we can edit them to remove errors and digressions.

# Lectures On-Line

http://www.testingeducation.org/BBST

Video lectures

- Students watch them before class
- Take simple quiz that checks that they watched the video and paid attention
- Then we do in-class activities

The results seem good

- Good student satisfaction
- Not enough time for the activities
  - **In an in-house course, time is not constrained by the same type of schedule. It's constrained by value to the project and the staff.**

# In-class activities that tie to the lecture

- At Florida Tech, we teach in a lab with one computer per student. Students work in groups. Activities are open book, open web. The teacher moves from group to group asking questions, giving feedback, or offering supplementary readings that relate to the direction taken by an individual group.
- Classroom activities vary. Students might apply ideas, practice skills, try out a test tool, explore ideas from lecture, or debate a question from the study guide.
- Students may present results to the class in the last 15 minutes of the 75-minute class.
- They often hand in work for (sympathetic) grading: we use activity grades to get attention and give feedback, not for high-stakes assessment. We want students laughing together about their mistakes in activities, not mourning their grades
- Developing good activities is sometimes easy, sometimes very difficult. We need to develop a large pool of activities and activity ideas.

# Sample Activity: Contrasting Missions

Your group is testing a spreadsheet / database. Please consider what your testing strategy should be and what types of test documentation to deliver.

Different groups consider this question:

- Traditional end-of-cycle test group
- Development support near start of project
- Testing a character database for a game
- Testing a custom application for a medical device maker

Groups report back, either by report/discussion to full group or by rotation of group representatives into discussion groups

# Application to a real product under test

- Like service learning, but not as heavy a commitment for the students or for me
  - We pick a well-known product
  - Students apply what they learn to that product
  - Typically, I use an open source product because it avoids NDA problems, students can show their work at interviews
  - Facilitates student learning (application level and above)
  - Facilitates student transfer of skills / knowledge to the workplace because students are doing the same tasks and facing the same problems as with commercial products
  - Work-products (results of assignments) are often credible work examples for employment interviews

# Application to a real product

As long as the assignments:

- **are not too far beyond the skill and knowledge level of the learner**,
- authentic assignments yield positive effects on retention, motivation, and transfer [48, 52, 119, 153].

# Orientation exercises

This is a special type of classroom activity:

- The task addresses an example or a task or a problem presented in the lecture that the student is about to watch.
- The expectation is that the student will not be able to complete the task correctly before seeing the lecture, but he can make some progress and gain insight into the underlying challenges of the problem.
- The intent is to ready the student to appreciate the solution presented in the lecture:
  - Cognitive readiness
  - Motivational readiness

# Open-book quizzes

These count very little toward the final grade, just enough to keep grade-conscious students motivated.

The instructional objectives of these quizzes:

- Help the student notice and understand key concepts and definitions
- Help the student check her understanding of the key concepts and definitions
- Raise an alarm to the student who is reading / watching but not understanding the materials.

Many testing concepts have conflicting definitions or applications. Students are "expected" to know the one from lecture. Quiz discussion forums can help students challenge that one true lecture definition, which builds their knowledge.

# Study-guide based exam

- http://www.testingeducation.org/k04/BBSTreviewfall2005.htm
- 100 questions, include all candidates for mid-term and final exam
- Students prepare answers together, assess each other's work
- I can require well-organized, thoughtful answers
- Fosters strategic preparation
- Reduces disadvantage of students whose native language is not English
- Creates cooperative learning tasks that should help limited-English-proficiency students improve language skills

# Study guide results

- Students inexperienced with these, often blow the first test
- Make-up mid-terms
  - Replace grade, not average, not best 1 of 2 results
  - Students who take it improve more (1st test compared to final exam) than students who did not take it
    - Practice effect, motivation confound
- Writing is better, answers are better, I have greater freedom to grade less forgivingly
- Many students told me this was the most valuable learning experience in the course, and the most time-consuming

# Study Guides

In-house use:

- Focus discussion of course materials
- Potential interview questions, especially if you revise them to apply to your class of product

# Challenging but focused assignments

I tend to give 4-6 smaller scale assignments, rather than 1-3 larger scale ones.

Ideally, each assignment would be:

- Linkable to an instructional unit
- Authentic (motivating)
- Peer-reviewable
- Appropriately complex

# Task scaffolding

- Scaffolding helps the student understand what is required in a task and how to do it. We provide:
  - Grading rubrics for some assignments
  - Study guides
  - Lecture examples that are similar to the example under test
  - Peer coaching (it's OK to share notes / ideas)

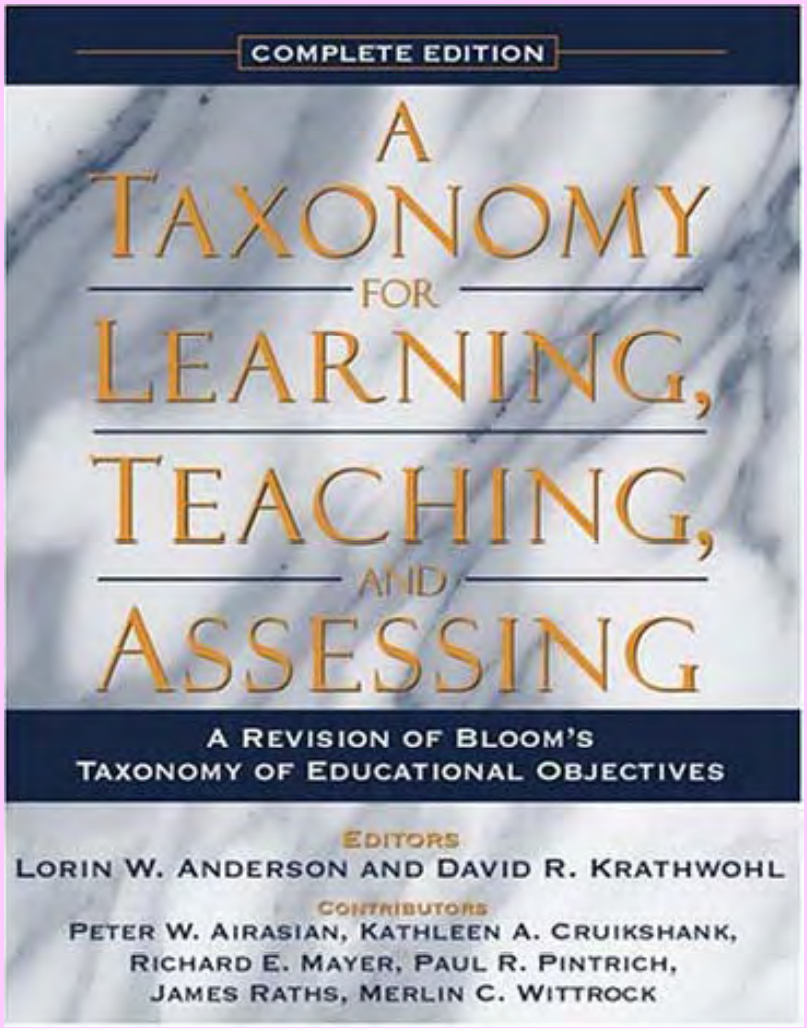
# Peer review

- Every assignment
- Every exam
- If you submit something (e.g. for grading), it gets peer-reviewed
- We often provide the reviewers with rubrics to guide their reviewing.

# Explicit discussion of learning issues

- I tell people what the learning objectives are for the course
- When we talk about exams, assignments or other tasks, I tie these to the learning objectives
  - I explain that quizzes are open book because the objective of the quiz is to help the student focus her reading, gain familiarity-level understanding of the material. The quiz acts as a study guide.
  - I explain the study-guide exam structure in terms of allowing time to develop higher-level answers, give examples of top-graded answers that acknowledged but then blasted my views. Unlimited time to prepare these..

# Characterizing Cognitive Complexity




Anderson & Krathwohl (2001) provide a modern update to Bloom's (1956) taxonomy

# Anderson Krathwohl update to Bloom's taxonomy, modified slightly for software testing

| **Knowledge dimension** | **Cognitive Process Dimension** | | | | | |
|---|---|---|---|---|---|---|
| | Remember | Understand | Apply | Analyze | Evaluate | Create |
| Facts | Lecture | Lecture | | | | |
| Concepts | Lecture | Lecture | | | | |
| Procedures | Lecture | Lecture | | | | |
| Cognitive strategies | Lecture | Lecture | | | | |
| Models | Lecture | Lecture | | | | |
| Skills | | | | | | |
| Attitudes | Lecture | Lecture | | | | |
| Metacognition | Lecture | Lecture | | | | |

# Variation for Testing: Facts

A "statement of fact" is a statement that can be unambiguously proved true or false. For example, "James Bach was born in 1623" is a statement of fact. (But not true, for the James Bach we know and love.) A fact is the subject of a true statement of fact.

Facts include such things as:

- Tidbits about famous people
- Famous examples (the example might also be relevant to a concept, procedure, skill or attitude)
- Items of knowledge about devices (for example, a description of an interoperability problem between two devices)

# Variation for Testing: Concepts

A concept is a general idea. "Concepts are abstract in that they omit the differences of things in their extension, treating them as if they were identical." (wikipedia: Concept).

In practical terms, we treat the following kinds of things as "concepts" in this taxonomy:

- definitions
- descriptions of relationships between things
- descriptions of contrasts between things
- description of the idea underlying a practice, process, task, heuristic (whatever)

Here's a distinction that you might find useful.

- Consider the oracle heuristic, "Compare the behavior of this program with a respected competitor and report a bug if this program's behavior seems inconsistent with and possibly worse than the competitor's."
  - If I am merely describing the heuristic, I am giving you a concept.
  - If I tell you to make a decision based on this heuristic, I am giving you a

Sometimes, a rule is a concept.

- A rule is an imperative ("Stop at a red light") or a causal relationship ("Two plus two yields four") or a statement of a norm ("Don't wear undershorts outside of your pants at formal meetings").
- The description / definition of the rule is the concept
- Applying the rule in a straightforward way is application of a concept
- The decision to puzzle through the value or applicability of a rule is in the realm of cognitive strategies.
- The description of a rule in a formalized way is probably a model.

# Variation for Testing: Procedures

"Procedures" are algorithms. They include a reproducible set of steps for achieving a goal.

Consider the task of reporting a bug. Imagine that someone has

- broken this task down into subtasks (simplify the steps, look for more general conditions, write a short descriptive summary, etc.)
- and presented the tasks in a sequential order.

This description is intended as a procedure if the author expects you to do all of the steps in exactly this order every time.

This description is a cognitive strategy if it is meant to provide a set of ideas to help you think through what you have to do for a given bug, with the understanding that you may do different things in different orders each time, but find this a useful reference point as you go.

# Variation for Testing: Cognitive Strategies

"Cognitive strategies are guiding procedures that students can use to help them complete less-structured tasks such as those in reading comprehension and writing. The concept of cognitive strategies and the research on cognitive strategies represent the third important advance in instruction.

There are some academic tasks that are "well-structured." These tasks can be broken down into a fixed sequence of subtasks and steps that consistently lead to the same goal. The steps are concrete and visible. There is a specific, predictable algorithm that can be followed, one that enables students to obtain the same result each time they perform the algorithmic operations. These well-structured tasks are taught by teaching each step of the algorithm to students. The results of the research on teacher effects are particularly relevant in helping us learn how teach students algorithms they can use to complete well-structured tasks.

In contrast, reading comprehension, writing, and study skills are examples of less- structured tasks -- tasks that cannot be broken down into a fixed sequence of subtasks and steps that consistently and unfailingly lead to the goal. Because these tasks are less-structured and difficult, they have also been called higher-level tasks. These types of tasks do not have the fixed sequence that is part of well-structured tasks. One cannot develop algorithms that students can use to complete these tasks."

Gleefully pilfered from: Barak Rosenshine, Advances in Research on Instruction, Chapter 10 in J.W. Lloyd, E.J. Kameanui, and D. Chard (Eds.) (1997) Issues in educating students with disabilities. Mahwah, N.J.: Lawrence Erlbaum: Pp. 197-221. http://epaa.asu.edu/barak/barak.html

In cognitive strategies, we include:

- heuristics (fallible but useful decision rules)
- guidelines (fallible but common descriptions of how to do things)
- good (rather than "best" practices)

The relationship between cognitive strategies and models:

- deciding to apply a model and figuring out how to apply a model involve cognitive strategies
- deciding to create a model and figuring out how to create models to represent or simplify a problem involve cognitive strategies

BUT

- the model itself is a simplified representation of something, done to give you insight into the thing you are modeling.

We aren't sure that the distinction between models and the use of them is worthwhile, but it seems natural to us so we're making it.

# Variation for Testing: Models

A model is

- A simplified representation created to make something easier to understand, manipulate or predict some aspects of the modeled object or system.
- Expression of something we don't understand in terms of something we (think we) understand.

A state-machine representation of a program is a model.

Deciding to use a state-machine representation of a program as a vehicle for generating tests is a cognitive strategy.

Slavishly following someone's step-by-step catalog of best practices for generating a state- machine model of a program in order to derive scripted test cases for some fool to follow is a procedure.

This definition of a model is a concept.

The assertion that Harry Robinson publishes papers on software testing and models is a statement of fact.

Sometimes, a rule is a model.

- A rule is an imperative ("Stop at a red light") or a causal relationship ("Two plus two yields four") or a statement of a norm ("Don't wear undershorts outside of your pants at formal meetings").
- A description / definition of the rule is probably a concept
- A symbolic or generalized description of a rule is probably a model.

# Variation for Testing: Skills

Skills are things that improve with practice.

- Effective bug report writing is a skill, and includes several other skills.
- Taking a visible failure and varying your test conditions until you find a simpler set of conditions that yields the same failure is skilled work. You get better at this type of thing over time.

Entries into this section will often be triggered by examples (in instructional materials) that demonstrate skilled work, like "Here's how I use this technique" or "Here's how I found that bug."

The "here's how" might be classed as a:

- procedure
- cognitive strategy, or
- skill

In many cases, it would be accurate and useful to class it as both a skill and a cognitive strategy.

# Variation for Testing: Attitudes

"An attitude is a persisting state that modifies an individual's choices of action." Robert M. Gagne, Leslie J. Briggs & Walter W. Wager (1992) "Principles of Instructional Design" (4th Ed),, p. 48.

Attitudes are often based on beliefs (a belief is a proposition that is held as true whether it has been verified true or not).

Instructional materials often attempt to influence the student's attitudes.

For example, when we teach students that complete testing is impossible, we might spin the information in different ways to influence student attitudes toward their work:

- given the impossibility, testers must be creative and must actively consider what they can do at each moment that will yield the highest informational return for their project
- given the impossibility, testers must conform to the carefully agreed procedures because these reflect agreements reached among the key stakeholders rather than diverting their time to the infinity of interesting alternatives

Attitudes are extremely controversial in our field and refusal to acknowledge legitimate differences (or even the existence of differences) has been the source of a great deal of ill will.

In general, if we identify an attitude or an attitude-related belief as something to include as an assessable item, we should expect to create questions that:

- define the item without requiring the examinee to agree that it is true or valid
- contrast it with a widely accepted alternative, without requiring the examinee to agree that it is better or preferable to the alternative
- adopt it as the One True View, but with discussion notes that reference the controversy about this belief or attitude and make clear that this item will be accepted for some exams and bounced out of others.

# Variation for Testing: Metacognition

Metacognition refers to the executive process that is involved in such tasks as:

- planning (such as choosing which procedure or cognitive strategy to adopt for a specific task)
- estimating how long it will take (or at least, deciding to estimate and figuring out what skill / procedure / slave-labor to apply to obtain that information)
- monitoring how well you are applying the procedure or strategy
- remembering a definition or realizing that you don't remember it and rooting through Google for an adequate substitute

Much of context-driven testing involves metacognitive questions:

- which test technique would be most useful for exposing what information that would be of what interest to who?
- what areas are most critical to test next, in the face of this information about risks, stakeholder priorities, available skills, available resources?

Questions / issues that should get you thinking about metacognition are:

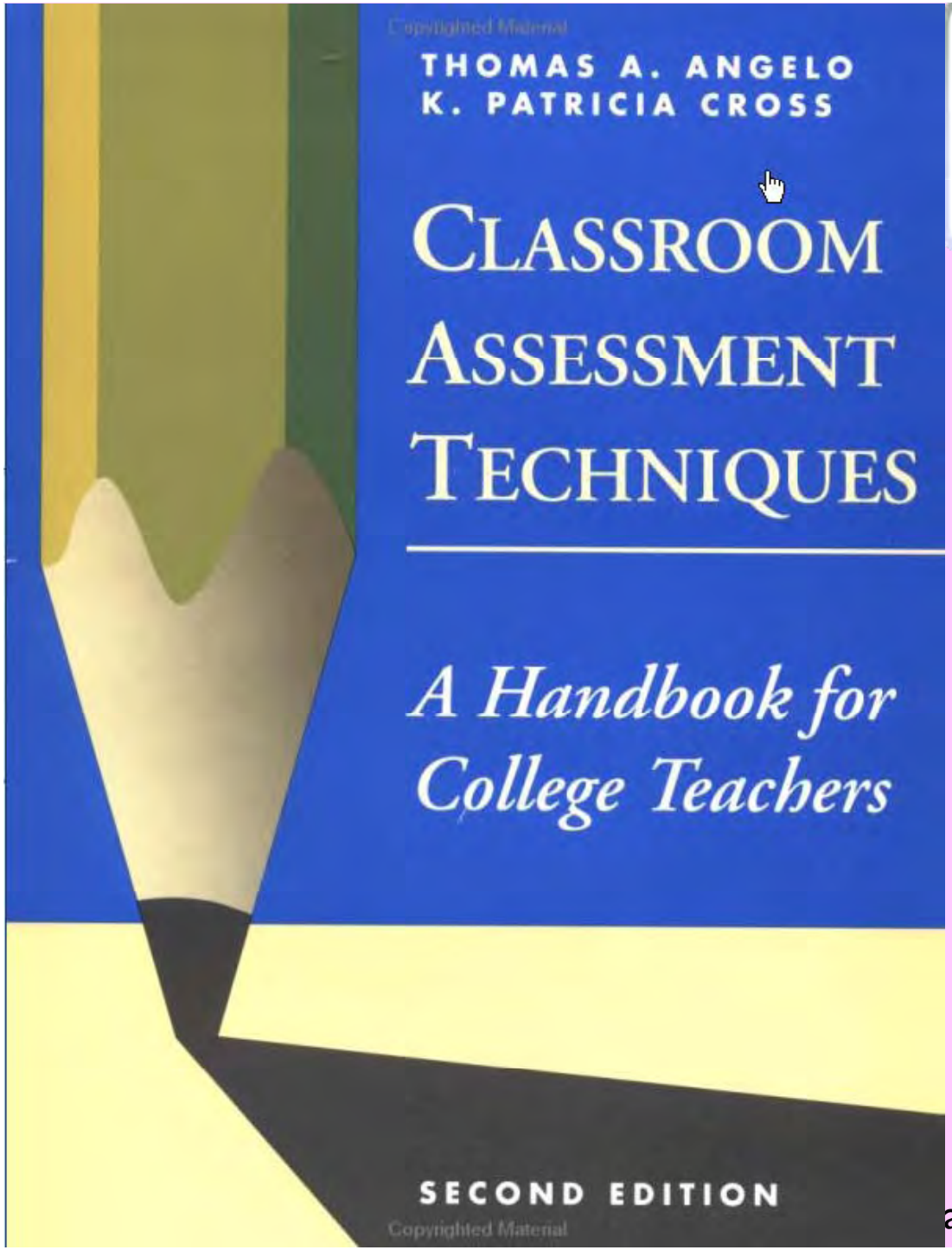
- How to think about ...
- How to learn about ...
- How to talk about ...

In the BBST course, the section on specification analysis includes a long metacognitive digression into active reading and strategies for getting good information value from the specification fragments you encounter, search for, or create.

# Assessment

1. Assessment at one level (e.g. facts / concepts) is not informative with respect to another level (e.g. evaluation)
2. "Authentic assessment" – assessment with simplified or artificial tasks is uninformative with respect to what can actually be done in real circumstances (flip side of the transfer problem)
3. Assessment that is apparently at a higher level is often reducible to lower level via:
    1. Study strategies
    2. Question-answering strategies

    This is part of the strong success of exam-review courses.




# We set objectives

- Cognitively complex material
- We need to develop skill, judgment, and attitudes, not just knowledge of facts and definitions
- We face the usual (for science education) transfer problems
- Set a few explicit learning objectives
- And assess against them

# My Learning Objectives

- Learn many test techniques well enough to know how, when, and why to use them
- Foster strategic thinking--prioritization, designing tests/reports for specific audiences , assess the requirements for complex testing tasks (such as test automation, test documentation)
- Apply (and further develop) communication skills (e.g. for bug reporting, status reporting, specification analysis)
- Improve and apply teamwork skills (peer reviews, paired testing, shared analysis of challenging problems)
- Gain (and document) experiences that can improve the student's chances of getting a job in testing

# Instruction on test-taking skills

Individual guidance to students on how to do specific tasks involved in a test, such as:

- What is the "call of the question"?
- Separating relevant and incidental (confuser) facts
- Ways of critically reading multiple-choice questions
- How to organize essay answers

We also provide a course video that shows the grading of several student answers' to a complex essay question and explains:

- how we decompose the question into gradable components
- how we grade these simpler parts of the question.

# SALG feedback

- Student Assessment of Learning Gains
  http://www.flaguide.org/cat/salg/
- Measures student perceptions of their 'gains' in learning
- Customizable, we customize it heavily
- Administered online
  - We find it easier to summarize data if we use our own form and Survey Monkey, rather than the original site
- FREE
- Beats the standard course evaluation form!
- Students each spent over an hour providing their evaluation.

# Employment value of the material and the work

We specifically advise students to create course portfolios that show off the workproducts from this or other courses.

When an assignment has been particularly practical, or has intrigued hiring managers in the past,

- We point this out to students
- We discuss what aspects of the work might be more relevant to the employers

# Organic evolution of the class

- Course development is not driven by an inflexible heavyweight process.
- It is driven by what is most valuable, most capable, of allowing imports and expertises.

# Enthusiasm and ongoing renewal (Hawthorne effect)

- Hawthorne effect: workers performed better and were more enthusiastic, in response to the combination of (a) change and (b) management attention.
- Usually discussed as an experimenters' trap (experimenter effects)
- Rather than seeing this as a blocking problem, we take advantage it, describe the changes we're making, show that we're interested in how they handle the changed cases, etc.

Opportunities for Improvement

# SHOULD-BE SUCCESS FACTORS

# Should be success factors

I know these should make the course better, but I haven't succeeded in figuring out how:

1. Drill / problem sets, to help students
   - Experience worked examples
   - Develop skills through practice
   - Experience an underlying common core when there is a lot of more superficial variation
2. Paired testing
3. Testing competitions
4. Student presentations
5. Employer / famous-person visitors

# Drill / problem sets

We want to help students:

- Experience worked examples
- Develop skills through practice
- Experience an underlying common core when there is a lot of more superficial variation

# Example: Domain Testing

Most widely taught testing technique

- For details, see http://www.testingeducation.org/BBST/Domain.html
- Easy to explain the basic concepts
- Classic examples widely taught
- Students quickly signal that they understand it
- But when you give them exercises under slightly new circumstances
  - They blow it
    - And then they blow the next one
      - And the next one . . .

# Common errors

## Consider an integer that can take on values from -999 to 999 inclusively

- **Doesn't spot a boundary.**
- **Offers excess values.** Students offer 998 as well as the appropriate 999 and 1000.
- **Doesn't spot a dimension.** (a) how many characters should this field handle? Same for positive and negative numbers? (b) if you delay after entering the first character, is there a risk of time-out? What delay durations should you test? Boundaries?
- **Doesn't articulate a risk.** Suppose we explicitly ask students to identify a risk and then identify relevant variable(s) and a powerful test appropriate to the risk. Rather than describe how the program might fail, the student might reiterate the test or make vague statements, like "fail to process this value correctly."
- **Doesn't explain how a test case relates to a stated risk.** When an assignment calls for such an explanation, students may respond inarticulately or irrelevantly.
- **Doesn't consider a consequence.** In real life (and in some of our test questions), the tester can determine more information than the bare range of an input field. The program will do something with the data entered. It is important, for each of those uses, to check whether the bounds imposed by the input filter are appropriate to the later use, and what consequence will result if they are not.
- **Poor generalization.** In more complex questions than the integer example here, students often pick inappropriate variables for analysis, such as treating each value of a binary variable as the best representative of its own 1-member class.

# Common errors

Students have learned the basic idea

- Bloom's taxonomy lower levels: know / explain

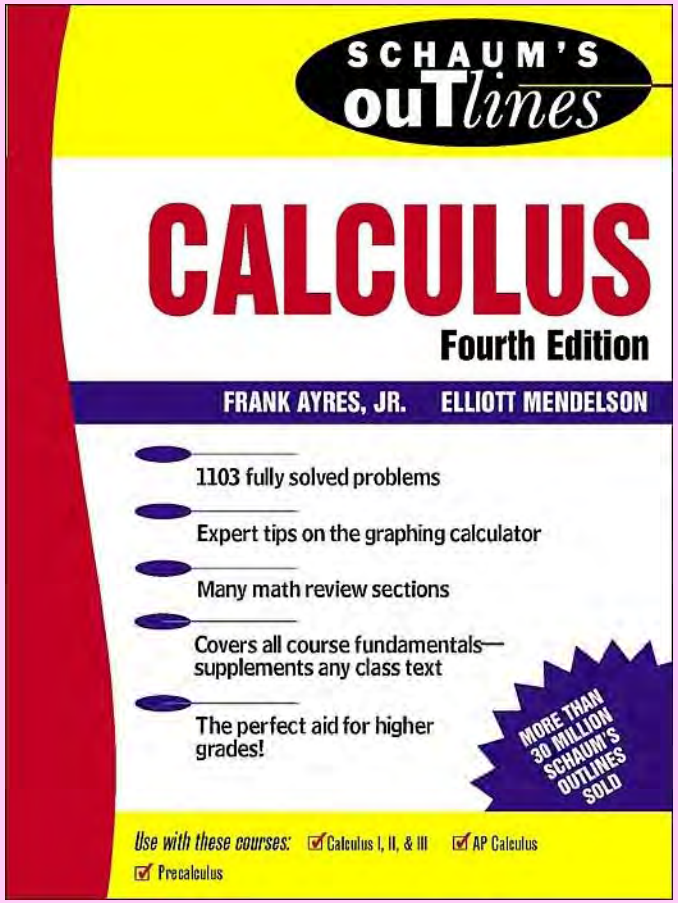
Students don't have a higher-level understanding

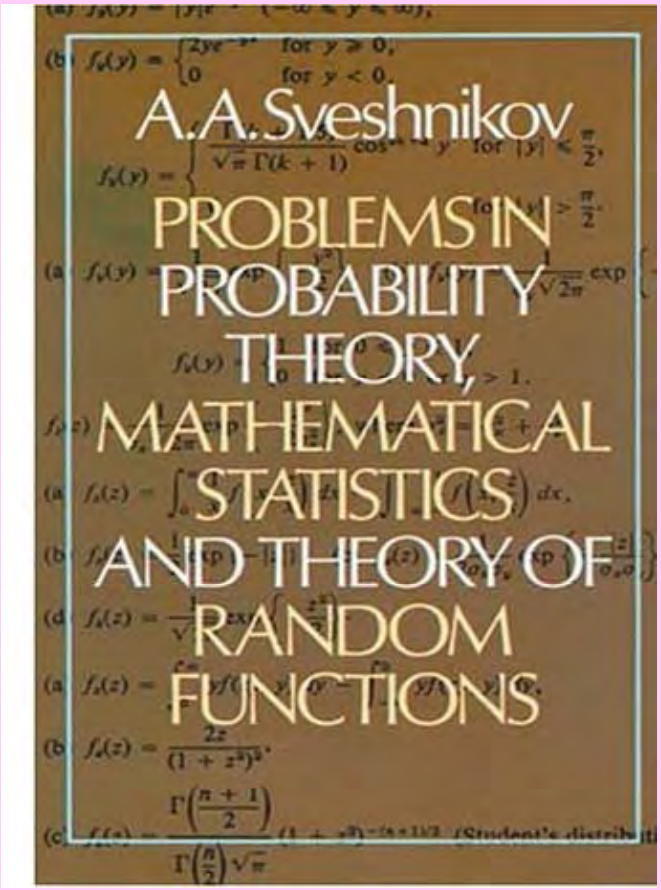
- apply / analyze / think through what they are learning

How can we increase their depth of understanding?

# Analogy to studying mathematics

Lots of practice exercises, like we used to do (and love, of course) as math students.



# I Tried This With Commercial Students

Many (often, most) of them needed a lot of practice under changing circumstances

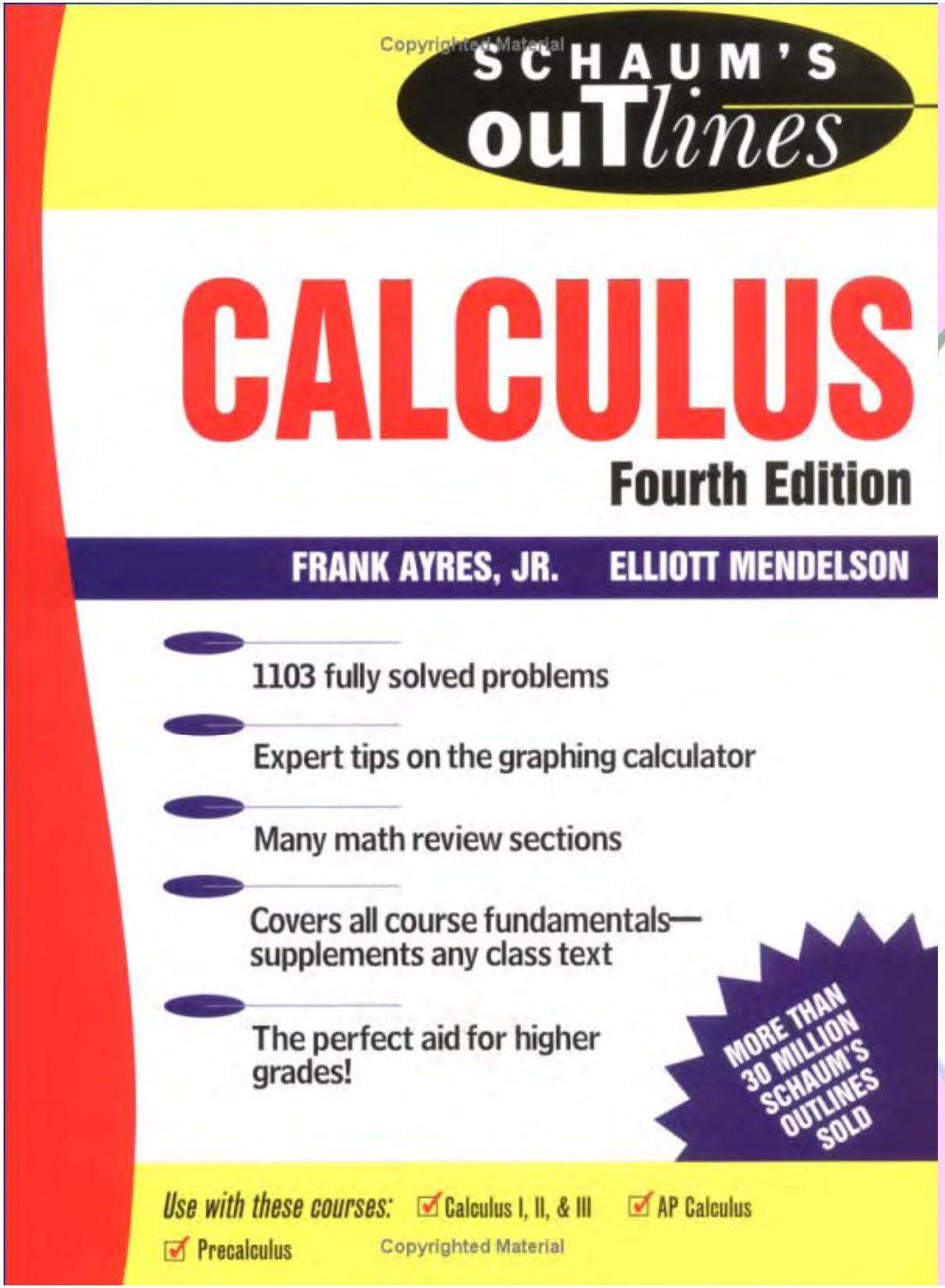
But the perceived slow pace of the course made them anxious

---

After 16 years of sending my staff to training, training my own staff, and training strangers . . .

. . . I realized two things:

1. This wasn't working (not for me, not for the field)
2. In terms of commercial training, I didn't know how to make it work




# Now, at Florida Tech!

Academic course

- Lots of practice exercises
- Like we used to do as math students
- It was impractical in commercial training
- **Now, at last, we can try it on university students.**

# Padmanabhan's Thesis: Practice on Domain Testing

- 18 classroom hours of lecture plus examples plus practice, practice, practice. Lots of procedural instruction and drill
- Students mastered every procedure
- Final exam

– Applied what they knew to similar questions (near transfer)

➢ **They aced them**

– Applied what they knew to a problem that was beyond their practice (not beyond the lecture) (a little bit farther transfer)

➢ **They all failed miserably**

- Successful transfer of learning requires more than procedural training and practice **(Of course, YOU already know that ...)**

# Paired testing

- Two-person group projects:
  - File bug reports together
  - Edit each other's reports
  - Split tasks so that each plays 1 or (a very few) roles.
- Bug-hunting competitions

- In practice, I've seen remarkable resistance to this, and little benefit to few students.

# Testing competitions

Students create test-related work products (bug reports, etc.) under time pressure.

How well / how much do they actually do?

We've done very few of these. We've seen them done very successfully by a colleague but don't yet feel as though we understand the ingredients for success.

# Student presentations

- 5 – 15 minutes
- Learn a lot about a topic by teaching it
- The other students in the room are typically bored, sometimes rude

# Employer / famous person visits

- Good stories (memorable examples)
- Can provide authoritative answers
- Can credibly resolve some differences of opinion among students (visitor must have involvement with the issue under dispute)
- Enhance the reputation of the instructor (cool to be connected to these people)
- Opportunity to create Hawthorne-effect application for the course.

# CURRENT CHALLENGES

# Current challenges

1. Disappointing essay exams
2. Getting students to watch videos in advance
3. Getting students to do preparatory exercises
4. Coping with an unstandardizable vocabulary
5. Classroom time management (discussion versus lab time)
6. Videos feature one white man
7. No active discussion in the videos
8. On-the-record videos make some storytelling difficult
9. Synchrony is important when students rely on each other
10. Grading time is substantial
11. Activities are hard to design
12. Multiple choice pool is small
13. We need better assigned readings

# Disappointing essay exams

- Many answers are
  - Shallow
  - Poorly focused
  - Not directly tied to the call of the question
- Weak work on first midterm is common.
  - Give students instruction on
    - Study strategies
    - Test answering strategies
  - Some students do much better
  - Others do about the same as before
- I think we're making progress on this, with the grading video and better coaching
- We did a blind regrading of 10 courses' final exams (three graders, consensus ranking from best to worst answers):
  - More recent exams are statistically better
  - Subjectively, the improvement is less than we had hoped.

# Getting students to watch videos in advance

- Age-old problem
- Like getting the students to do required reading before class
  - Lawyers use fear of humiliation as the motivator for reading the texts and doing homework on time
- Day & Foley used short essay questions; we used multiple choice questions. We don't have quizzes for every lecture:
  - Our questions are open book; we encourage challenges to the grading.
  - We close the quiz just before class, so students do it before class or they get 0.
- After we switched to open book, we started getting positive feedback from students: requests to add more quizzes. This is an encouraging event.

# Getting students to do preparatory exercises

I've been lucky; in most courses, most students just do these exercises when I ask them to. However:

- I don't have many such exercises. Once the novelty wears off, perhaps fewer people would do the tasks
- Some students don't understand the motivation for the task, even when the task motivator has explained why this type of task / this particular task is instructionally valuable.
  - The idea of setting up readiness for learning is a hard-to-grasp concept for many students, especially engineering students
- Some students resent being asked questions that the instructor knows the student(s) can't (yet) solve.

# Unstandardizable vocabulary

What's a "test case"? For that matter, what's the definition of a "software error"?

- There are several definitions in common use. There have been efforts to standardize the vocabulary around One True Definition, but how can you call a word usage incorrect if it is widely used among practitioners and useful to them?
  - Michael Bolton is starting a project to create an Oxford-English style dictionary for testing terms, i.e. a project to document usage as it exists rather than to advocate for specific definitions. We will help him in that.
- The adoption of definitions in a way that is explicitly course-specific, paying respect to some alternatives, is very disturbing for some students and very confusing for others.
  - The quizzes help a bit with this.

# Classroom time management

Imagine coming into class:

- Students have questions about the video they watched last night
  - The students who ask the questions obviously did watch the video and the questions are not bad
  - How much time should the instructor spend answering the questions or leading a discussion in class?
    - It's a pleasure to discuss with engaged students
    - If you spend too long, there's no time for the planned activity
- In one semester, some students figured out that they could escape activities by creating discussions.
  - A few students would watch the video and create a discussion.
  - Students swapped the responsibility for class prep from class to class.
  - They avoided a lot of activities (too many) that term.

# Videos feature one (1) white man.

Currently, I am the only lecturer.

- We want to redo several videos to feature other role models, reflecting the diversity of leaders in the field in the process.
- Coordinating this work is difficult. A couple of pilot efforts have failed so far.

# No active discussions in videos

Infomercials often feature 2-4 people talking about the product being sold, its attributes and benefits. Often, someone plays the skeptic. Content is reinforced with personalized stories, with listeners asking questions and expressing interest.

This is a very different kind of "lecture" format from one talking head.

We are beginning to map out a discussion video for one segment, Bug Advocacy. If we can get this to work, we'll try it for others.

For now, we see this as a supplementary retelling of the lecture rather than as a replacement for the lecture. When we get better at it…

# On-the-record video makes some storytelling difficult.

In my face-to-face classes, I have some great stories to tell. This was one of the marks of the BBST commercial class—I had stories for every segment. Some were part of the main lecture, others came out in responses to questions.

None of these violated nondisclosure agreements, but some were embarrassing. They were OK for oral presentation to a limited-group classroom setting, but I would never post them to a public website.

It is possible to anonymize some of the stories to some degree, but they lose some of their power in the process. Hearing that Hewlett-Packard (maybe 3/4 of the students in the room have a computer or printer by HP) had a problem is different from hearing that some obviously-stupid anonymous company had a problem.

# Synchrony is important when students rely on each other

In a traditional class, I expect some students to hand in late work. There are lots of reasons, some good. Unforgiving assignment deadlines has not been my style.

- In classes with peer reviews, late assignments by Student 1:
  - Student 1 cannot/should not review Student 2's work before submitting her own work. Student 2 gets late feedback. If Student 2 gets to submit a 2nd draft, this delay is unfair.
  - Student 3 cannot review Student 1's work until Student 1 submits it, and this interferes with Student 3's work scheduling.
- I now enforce deadlines, and students are unhappy with this

# Grading time is substantial

Grading classroom assignments and essay exam answers takes a lot of time.

In the AST classes, taught by volunteer instructors, we expected to drive grading time to near-zero by relying primarily on peer review. In practice, in the AST Foundations class:

- 3 weeks lecture, 1 week exam
- 20 students, attrition to 15 students, working 8-12 hours per week on the class
- 3 volunteer instructors spend 10-15 hours each, plus extra time at very end (pass-fail decisions). Perhaps half of this is assessment (grading, comments) related.

# Activities are hard to design

We have several individual exercises that we think are excellent.

We have collections of activities from other fields that we know will serve as guides / data to help us create some generalized patterns for activities.

We have a few patterns.

Most of this work is yet to do.

Our biggest practical problem is finding a graduate student(s) to support who can pick up this task.

# Multiple choice pool is small

We have about 100 questions.

I now have a thorough discussion of Multiple Choice drafting guidelines, at http://www.satisfice.com/kaner/?p=24. Several colleagues helped with this, at the last Workshop on Open Certification.

We will develop about 30-40 more questions per learning unit as each unit is adapted for the AST course series

# We need better assigned readings

The lectures are useful but they are not a substitute for a textbook or assigned readings.

- University courses, relatively easy to assign texts and coursebooks of readings from IEEE/ACM/etc journals
- Public courses, much harder to assign proprietary readings. Few practitioners have access to IEEE/ACM digital libraries and local editions of several books are unavailable in many countries.
- This problem is persisting as we design AST versions of the classes, but we might be able to make progress against it as we get more certified instructors per class. This might be their way of contributing improvements to the class.

# ADDITIONAL ACTIVITIES UNDER THE GRANT

# Additional activity

- BBST instructors course
- AST adaptation of BBST
- Tester certification

# BBST Instructors course

Rebecca Fiedler is heading up the project to create the BBST instructors course.

- Free course
- Entirely self-directed
- Students can do assignments, save them, submit them as workproduct examples later (this is needed if they want to be AST instructors)
- Student work is not graded (apart from some open book multiple choice exams)
- For current status, see
  - http://www.bbstinstructors.org/

# AST adoption of BBST

Association for Software Testing provided a letter of support for the original project and is dedicating significant resource now that the grant has been awarded, including perhaps ½ of the time of the Executive Director.

- The BBST university course is 15 weeks, about 15 learning units (45 contact hours, plus expected 150 homework/study hours). We have about 20 learning units' material for the course.
- AST BBST creates a new course for each learning unit:
  - Quizzes, preparatory exercises, homework, group project, final exam, everything peer-reviewed
  - Students agree to work 8 hours / week, actually spend 10-12
  - 3 weeks of lecture, etc., last week is final exam
- AST expects to offer 20 courses, including some not on BBST list (other experts have volunteered to add units)
- AST certifies instructors in each course, and will certify testers who have completed 10 AST courses.
  - AST offers course free to its members. AST-certified instructors can offer the course they are certified in, for fee or for free.

# Tester certification

**Overview**

- Ultimate certifications are:
  - Degree → so why is there no degree in SW testing?
  - Professional license
- Given that we are going to do professional certification rather than degree or licensing,
  - How should we assess knowledge and skill to qualify for certification?
- The Open Certification Project

# Why is there no undergraduate degree in testing?

- Florida Tech's curriculum design
  - Meet ACM standards for CS curriculum
  - Meet ACM/IEEE standards for SE curriculum
  - 3 required core courses in testing (BBST / test-first / tools) plus options
  - Psychology, human factors, discrete math, logic, metrics, requirements, model-based (UML) design
  - Plus the CS/SE courses
- **Abandoned curriculum after employer review**, adopted test-heavy SE curriculum instead
- Resistance to testing courses
  - Field seen as trivial by many academics

# Challenges of dealing with testing certifications

Issues "in principle"

- assessed level of knowledge
- imposed orthodoxy via body of knowledge
- uncertain meaning of the certificate (often pushed as evidence of competence or even expertise. Two of them have been compared by marketers to M.Sc. Degrees.)
- public impression of the profession (how hard can this profession be if you can be certified as competent after a 16-hour review course?)

We (Kaner and several well-known practitioner colleagues) are dissatisfied with multiple-choice-based certification exams by ISTQB, ASQ, QAI and others. We question the motivation (in some cases, we perceive the goal is sales of review courses rather than assessment of knowledge/skill of the people who will be certified), the assessment vehicle (multiple choice is a weak format for skill or higher level knowledge) and the body of knowledge (simplified for easy review-course teaching to inexperienced people with money)

# Challenges of dealing with testing certifications

Issues "in practice"

- Cheating
  - M/C questions often posted to the Net, so some examinees have studied from the questions and others not. For some certs, the review courses prep students for some specific questions that typically appear on the exam
- Inter-grader reliability
  - For essay questions, this is a huge problem

Cost

- proprietary study materials
- expensive courses and exams

# Standard BOK?

Widespread disagreement in the field

- I started writing TCS because I felt that then-current views of good testing were impractical. Today's BoKs aren't much different.
  - Risk-based testing?
  - V-model?
  - Value of the factory approach?
  - What is appropriate test documentation?
  - What is a test?
  - Independence vs collaboration (programmer / tester)?
  - Scope of testing?
  - Manufacturing metaphor vs investigation metaphor vs police enforcement metaphor

# Anderson Krathwohl update to Bloom's taxonomy, modified (slightly) for software testing

| Knowledge dimension | Cognitive Process Dimension | | | | | |
|---|---|---|---|---|---|---|
| | Remember | Understand | Apply | Analyze | Evaluate | Create |
| Facts | Multiple choice questions typically test at this level of knowledge | | | Testing practice typically operates at this level of knowledge | | |
| Concepts | | | | | | |
| Procedures | | | | | | |
| Cognitive strategies | | | | | | |
| Models | | | | | | |
| Skills | | | | | | |
| Attitudes | | | | | | |
| Metacognition | | | | | | |

# Assessment (refer back to Bloom taxonomy): Concerns about current certification exams

1. Assessment at one level (e.g. facts / concepts) is not informative with respect to another level (e.g. evaluation)
2. "Authentic assessment" – assessment with simplified or artificial tasks is uninformative with respect to what can actually be done in real circumstances (flip side of the transfer problem)
3. Assessment that is apparently at a higher level is often reducible to lower level via:
    1. Study strategies
    2. Question-answering strategies

    This is part of the strong success of review courses.

# Overview of the Open Certification Project

- Open to the public (anyone can view / comment. AST members can create questions and exams)
- Open source (everything is Creative Commons or GPL)
- Open questions (visibility / comments)
- Open exams (multiple exams reflecting different visions of testing)
- Open body of knowledge (questions based on free-access or lightly-restricted access on the web; possible reference to books)
- No-cost administration
- Full feedback

# Job interview scenario

- Jane applies for a job at Bank of Americanadia.
- Resume claim:
  - "Open Certification in Software Testing: (Context-Driven)"
  - Certificate: Suggests retest of the exam with instructions
- At the job site:
  - Takes the exam (free)
  - Gets 38 / 50 (is this good or bad?)
  - Printout to employer:
    - Questions
    - Answers
    - Commentary
  - Discussion

# Update (February 2008)

- We described the Open Certification project in the NSF proposal
- Satisfice Inc. promised to donate $10K to open cert project if the main project was approved by NSF, this would be the only NSF-grant-related funding to go to open cert (no money directly from grant)
- AST adopted Open Cert and provided some funding
- HOWEVER:
  - We had ongoing concerns about the Open Cert approach because it was still a multiple choice exam (no testing of skill) and because there are few universally-accepted definitions, drafts of questions had to footnote to reference material ("According to Beizer, xx means yy")
- After extensive discussions, AST switched resources from Open Cert to the BBST certification, b/c BBST courses foster higher-level knowledge
- Open Cert project contributes the question-creation/review server to the BBST project
- Satisfice 10K will go to AST BBST instead of open cert

# PROJECT ASSESSMENT

# Project assessment?

We need to assess

- progress of the project for NSF and
- success/failure of the subprojects
  - For NSF
  - For other instructors (lessons learned)
  - For ourselves (as BBST teachers)

# Project assessment

1. Project task tracking
2. SALG characterizations
3. Blind comparisons of final exams across courses (how do the answers rank, across courses)
4. Open comparisons of final exams across courses (how are the answers similar or different across courses)
5. Employer reactions X months later
6. Student reactions X months later
7. Instructor reactions
8. Adoption statistics?
9. Peer review / external evaluation
10. ??? What else ???

# Project task tracking

This is fairly straightforward:

- What did we expect to do
- What did we actually do
- How "done" is it and why do we think so
- What are we carrying over to the next review period

# SALG characterizations

Student assessment of learning gains:

- 1-hour course eval by students
- NSF funded instrument, much more informative than traditional smile sheets
- We have just moved to SurveyMonkey to simplify administration
- Fewer than 100% student feedback; we feel that we cannot demand responses from students who don't want to provide them (we think this goes beyond our IRB terms/conditions)
- We also have an instructor survey, just starting for next AST courses

# SALG -2

SALG asks students how well they feel they learned various topics:

- We list the learning objectives, and they assess the course against each
- We list the learning support materials and they assess each
- We ask questions about time spent, difficulty and value of the course

We think that student self-appraisal of how much they learned is a reasonable measure of how effective the course was for them.

We wish we could correlate this with their actual exam scores, but these responses are anonymous.

# Blind comparisons of exams

We've tried this twice:

- My course final exams include several questions. Some questions on any exam have appeared on other exams.
- Students typed handwritten exam answers, assigned code number identifiers to each answer. We then pooled the coded answers into a set of all answers to a given question, then regraded the answers, ranking them from best to worst.

First pass,

- several people regraded independently (all of them had taught testing courses, some at university). **Almost no correlation among rankings**.

Second pass, 18 months later (BBST course had matured),

- three graders ranked answers as a consensus activity (spent 3 days together in a conference room)
- Later classes statistically significantly better answers, but it was not a big effect.

# Blind comparison of exams

Value of this assessment:

- We can tell whether performance on exams is improving
- We can eventually compare this data across institutions (e.g. AST against Florida Tech)

Limitations

- My impression is that much of the improvement shows up in classroom discussion, in the activities and in the assignments
- The historical assignments are not consistent enough for blind regrading, but we might try something anyway

# Open comparisons of exams

Many exams will not be directly enough comparable for blind regrading, e.g. the Huston-Tillotson course will probably use a different sample application-under-test and that will show up in several exam questions or student examples in their answers.

Rather than trying to rank exams directly, the goal is to come up with a holistic grading structure that lets us compare across settings.

# Employer reactions X months later

Primary setting for this data:

- Employer E has its staff take the BBST course (e.g. AST BBST sections)
- X months later, the manager(s) of that staff assess the extent to which the course helped the staff improve

Some employers signed letters of support for the NSF project to do this.

Currently, this has not started and we are running into practical difficulties (management changes obsoleting old agreements)

# Student reactions X months later

- Second round of SALGs, X months after the course.
- We will start doing this with AST-BBST, probably in July
- The second round will ask more about long-term value of what they learned (and what they feel they're still missing)

# Instructor reactions

Instructor survey, comparable to SALG, for each course

In progress.

# Adoption statistics?

Because this is free online, it is very hard to find out who is actually teaching from it.

We have found some instances of plagiarism of the course materials (google searches) and we have been told that the materials have been translated into Chinese, Hebrew, Spanish, and Arabic, but we don't have detailed information.

We need to start begging the instructional community for case studies, etc.

# Peer review/ external evaluation

Create a questionnaire (somewhat like SALG) asking for overall impressions of each course or learning unit against its learning objectives.

Solicit free-form comments about the material and how to improve it.

We will start this in 2009

# What else???

Discussion at 2008 WTST meeting:

- Before/after testing. Test at start of course to show they don't know much about testing and then at the end to show they learned something.
  - Comment: We did this in the Padmanabhan thesis with predictable result. Students started with little knowledge and ended with more. We see obvious weakness in responses to the preparatory exercises. However, creating a before/after exam will be enormously time consuming for the students, and possibly demoralizing.
  - We might do this for academic fall 2008 to make the point, then drop it.

# WHERE WE'RE GOING (ADVISORY BOARD)

# Where we're going as a collective research group

- Opportunities for:
  - Broad collaboration across industry / academic and across academic institutional boundaries
  - Broad data pool
  - Grants to support collaboration and assessment
  - Commercial profit
  - A broader pool of people achieving technician-level or higher-level entry into the skilled workforce
- More general benefits
  - Activities pool more broadly useful in SE education
  - Instructional methods more broadly applicable

# Broad collaboration across industry / academic and among schools

The BBST material and teaching style continue to evolve with feedback from multiple quarters.

Example: March 2008

- BBST-Bug Advocacy is being completely rewritten from the academic course version to the AST version (and the AST version will be put into use in the academic course), based on AST (advanced practitioner) evaluations in 3 AST teachings of BBST Foundations.

# Broad data pool

Use of the course—and my instructional support—as part of the NSF project comes with a reciprocal obligation. The beneficiary organization must provide data and access to raw responses.

This makes it possible to compare exam answers (for example) across practitioner/academic divide

# Grants to support collaboration

NSF proposal to Florida Tech covers Florida Tech expenses for this project

If your institution wants to adapt the course, there is potential for funding for your work. My work (apart from travel expense, out of pocket expenses) to support you is covered by Florida Tech and current NSF project, so your application would be for funding for your institution.

# Commercial profit

AST model implements an open source economics vision:

- Open source base materials (the course)
- Intense services associated with the base materials (assessment, coaching, customization for individual employer)
  - No one is required to purchase these services
  - Some people can succeed without them
  - AST will offer courses free to members, but we figure a max capacity of 720 students-in-class per year (20 students per class, 3 classes open / month). If demand is greater, it will be met by for-profit instructors who offer their own version of the course. AST will certify instructors for individual courses (see AST policy /procedures for details.)
  - Some of the active volunteer instructors are active, helping improve materials, etc., specifically to prepare for their commercial offering of the same classes.
  - This is like the linux-related service businesses. We think it's entirely ethical

# Technician education

NSF funding available to community colleges to adapt material like this for technician education.

Have had inquiries from one community college, but they seem more focused on adult professional education.

We think BBST could easily be adapted to work well in a technician training program at a community college and would be glad to help

# Activities pool benefits SE education

The collection of activity patterns for software testing will probably generalize quickly to other courses in software engineering

- Specific activities, as examples (like the Activities in Psychology series)
- Patterns, applied to other courses
  - Opportunities for B.Sc. Thesis or M.Sc. level work to apply specific patterns to specific courses, publishing results at conferences like ACM Southeast Regional, CSEET, etc.

# WHAT THE PROJECT NEEDS TO WORK ON

# What the project needs to work on

1. Additional venues (including access to assessment data)
2. Figuring out what assessment data we should collect
3. Actually doing the analyses
4. New models for video content (and doing the videos)
5. Publishable pools of activities
6. Oxford English analogy for documenting testing vocabulary
7. Funding to support additional venues
8. Funding to support administration of the project
9. Active collaborators on the instructors' course
10. Writing up what we're learning
11. Alternative models of industrial certification
12. What else?