BBST® Foundations
Course Syllabus
Introduction

This course is a prerequisite for all other courses in the Black Box Software Testing series. You will learn the essential concepts in black box testing and solve common challenges faced by beginners and by experienced testers.

Don’t let the name “Foundations” fool you. The course is not designed exclusively for the beginning tester. It provides deep insight into core concepts and common challenges in testing. Even if you are an experienced tester, you will discover there are valuable things to take away from this course.

This course is for those of you who want to be great testers. Perhaps you feel that you haven’t approached testing with the right mindset, or your team of testers needs a common ground for testing approaches and techniques, not just terms and tools.

This syllabus will help you get a clearer idea of what the course entails. You will find here a detailed outline of the BBST Foundations course and information about the structure, format and workings of the course.

The lesson introductions presented here are excerpts from the course workbook: *Foundations of Software Testing*, by Cem Kaner and Rebecca L. Fiedler.
How It Works

BBST Foundations is a 4-week course organized in online classes of up to 21 students and 3 instructors.

Before each class starts, students receive the workbook and login access to the course platform. You will use the platform to watch video lectures, participate in discussions with your peers and with the instructors, download readings and submit assignments.

Aside from these resources, you will have the book to help you follow along. All required readings and part of the recommended readings are available as pdf files on the course platform.

The course material is structured in 6 lessons. Each lesson is comprised of a video lecture, a set of slides and a chapter in the course workbook “Foundations of Software Testing.” These are accompanied by a list of required and recommended readings, a quiz (for self-assessment) and one or two assignments. Assignment deadlines are on Wednesday and Sunday each week. You will receive feedback on it from your instructor, to which you can reply back and so on. You will also do peer reviews for several assignments.

After the sixth lesson, there is a live Q&A session with Cem Kaner and the class instructors. In this session, students can discuss any topics from the class in more details and ask Cem or one of the instructors any questions about the material or more general questions about testing and career experiences.

The fourth week is dedicated to preparing for the exam and taking the exam. After the course, you can participate in the grading of your exam by opting for the 1:1 interactive grading session with your instructor.

The final course assessment—“successfully completed” or “has not successfully complete yet”—is based on several criteria: participation in the course activities, incorporating feedback received from instructors, performance on the main course assignments and the exam performance.
## Learning Objectives

As this is the first course in the series, its objectives cover both foundational content and skills for succeeding in courses of this type.

<table>
<thead>
<tr>
<th>Objective</th>
<th>Anderson/Krathwohl level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  Familiar with basic terminology and how it will be used in the BBST courses.</td>
<td>Understand</td>
</tr>
<tr>
<td>2  Aware of honest and rational controversy over definitions of common concepts and terms in the field.</td>
<td>Understand</td>
</tr>
<tr>
<td>3  Understand there are legitimately different missions for a testing effort. Understand the argument that selection of mission depends on contextual factors. Able to evaluate relatively simple situations that exhibit strongly different contexts in terms of their implication for testing strategies.</td>
<td>Understand, Simple evaluation</td>
</tr>
<tr>
<td>4  Understand the concept of oracles well enough to apply multiple oracle heuristics to their own work and explain what they are doing and why.</td>
<td>Understand and apply</td>
</tr>
<tr>
<td>5  Understand that complete testing is impossible. Improve ability to estimate and explain the size of a testing problem.</td>
<td>Understand, rudimentary application</td>
</tr>
<tr>
<td>6  Familiarize students with the concept of measurement dysfunction.</td>
<td>Understand</td>
</tr>
<tr>
<td>7  Improve students’ ability to adjust their focus from narrow technical problems (such as analysis of a single function or parameter) through broader, context-rich problems.</td>
<td>Analyze</td>
</tr>
<tr>
<td>8  Improve online study skills, such as learning more from video lectures and associated readings.</td>
<td>Apply</td>
</tr>
<tr>
<td>9  Improve online course participation skills, including online discussion and working together online in groups.</td>
<td>Apply</td>
</tr>
<tr>
<td>10 Increase student comfort with formative assessment (assessment done to help students take their own inventory, think and learn rather than to pass or fail the students).</td>
<td>Apply</td>
</tr>
</tbody>
</table>
Week 1 - Course Policies, Lesson 1 and Lesson 2

You will start the course by familiarizing yourself with the canvas platform and the course structure, watch the course overview lecture, meet the instructors and meet other students.

You will need to sign the Privacy Policy, Terms of Use and Intellectual Property Policy before Tuesday evening in the first week of the course.

Lesson 1: Overview and Basic Definitions

In this lecture, Cem provides an overview of the online Black Box Software Testing courses and introduces some definitions commonly used in the testing field.

Lesson Introduction

Testing terminology is not uniform. One reason for that is that the underlying philosophy of testing, that is reflected in the language, is not uniform. For example, some people think that a test is not a test unless there is an oracle (an expected result that will tell you whether the test passed or failed). Testing without an expected result is, to them, incompetent. For them, it is natural to include expected results as part of the definition of “test.” However, other testers consider explorations of the product (which includes testing without knowledge of what will happen) to be quite useful. And others think that there are many possible oracles for the same test and that the skilled tester will focus on different potential errors (consider different oracles) at different times. To these people, the choice of oracle is a separate thing from the basic idea of a test. No definition of “test” will satisfy all three groups.

You will also be introduced to the way quizzes are used in this course and check your understanding of the lecture with Quiz 1. There is a quizz for every lesson.
Lesson 2: Strategy

This lecture introduces you to context-driven testing. Before watching the lecture, you should do the orientation exercise—Describe the role of testing groups. You will be able to see the answers of other students in your class and discuss them.

Lesson introduction
The goal underlying the orientation exercise and the required reading is to open your mind to the great variety of testers’ roles. If you have no preconception about testers and their roles, you will learn something new that will help you when you interview for your first testing job. If you come to the course with a narrow view of the roles of testers and test groups, this lesson will challenge you to unlearn your narrow view.

The lecture defines testing as a service to stakeholders. In particular, it is an empirical search for quality-related information about the product, on behalf of those stakeholders. The key stakeholders often have different informational needs. To meet those needs, testers have to adjust how they test (what they do, what tools they use, how they prioritize their time).

Testers must also adjust to the practical realities of the project, such as the budget, the schedule, the skills of the staff, and the availability of suitable tools.

At its core, context-driven testing reflects this dual adjustment to the needs of the key stakeholders and to the practical realities of the project.

Application assignment: Mission of testing
Apply what you’ve learned about the mission of testing. Due on Sunday evening (7 days into the course).
Week 2 - Lesson 3 and Lesson 4

In week 2 you will learn about oracles and test coverage. You will complete two phases of an application assignment, by first submitting your answer and then responding to work from a couple of your peers.

Lesson 3: Oracles

This lesson introduces you to oracles. You will complete an orientation assignment—Testing a Word Processor—before watching the lecture. After submitting your answer, you can review other answers, posting comments as you wish.

Lesson introduction

The classical view of oracles is that they are mechanisms for determining whether a program has passed or failed a test. Along with this idea of the oracle, there is a classical idea that testers will (or even must) have an oracle for every test. Thus, for example, Glen Myers (1979) tells us that we must have an expected result for every test. This lesson presents a different view. Ideally, you will take three things away:

1. No test has one true oracle. The best we can achieve are useful approximations.

2. We can describe the process of deciding that a program probably passed or failed a test as the outcome of a comparison and identify several widely-used comparators.

3. It is useful to have a collection of many very specific oracle heuristics to support test automation.

Application assignment: Using the Consistency Heuristics

You will have a few different contexts and you will work with consistency heuristics to learn how to better argue fixing specific bugs.
Lesson 4: Programming Fundamentals and Coverage

This lesson presents information about basic data handling and storage to help testers think about the multi-dimensional problem of test coverage in more sophisticated ways. Its goal is to improve testers’ computing-related literacy and to build a bridge from the material covered so far to the more technical issues in Lessons 5 and 6.

Lesson introduction
Lesson 4 introduces several topics from the fundamentals of computing:
- How computers store numbers and text. What it means to overflow the storage reserved for a piece of data.
- How computers do arithmetic. The basics of binary storage and binary calculations.
- The nature of floating-point arithmetic. Why floating-point calculations introduce errors as a matter of design. Why testers need to be aware of this.
- The main data types.
- The main control structures in programs. We include interrupts and exceptions in this set, and a few examples of how programs get into trouble with each of these.

Assignment: Phase 2 of Using the Consistency Heuristics (due on Sunday evening)
Here, you will provide and receive feedback from your classmates on the Consistency Heuristics assignment.
Week 3 - Lesson 5 and Lesson 6

This is the last week before the exam. You will learn about some technical issues in testing and about measurement and metrics.

Lesson 5: The Impossibility of Complete Testing

This lesson explores the complexity of determining when testing is finished and how the goal of complete testing is unattainable. Before watching the lecture, you should complete the orienting activity: Testing a Square Root Function.

Lesson introduction

We expect you to remember six things from this lesson:

- **Two key definitions:**
  - Two tests are **distinct** if one test would expose a bug that the other test would miss.
  - To achieve **complete testing**, you have to run every distinct test.

- **Two key examples:**
  - In the **MASPAR square root example**, testers had to test all possible inputs to find the function’s two bugs.
  - In the **Telenova stack overflow example**, covering all branches and statements and independent sub-paths was not enough. To replicate a system-killing bug in the field, testers had to create sequences that were so long and so complex that it would be impossible to find all bugs like this in the lab.

- **One key formula:**
  - If $V_1$ through $V_k$ are $k$ independent variables, and if $N_i$ is the number of possible values of variable $V_i$, then the number of combination tests of all the variables together is $N_1 \times N_2 \times \ldots \times N_k$.

- **One conclusion:**
  - **Complete testing is impossible and therefore all of testing involves tradeoffs.** Testing involves many tasks, such as designing and running tests, writing effective bug reports, documenting test ideas, creating test
tools, etc. The amount of time needed to do all of these is infinitely greater than the time testers have. You only have time to do a small sample of this work and the time you spend on one task will no longer be available for the others. Inflexible directives, like “You must write down an expected result for every test”, are unreasonable because they demand a huge amount of work on one testing task without considering what other testing tasks will be left undone as a result. The optimal tradeoffs will vary on a project-by-project basis.

Assignment
You will participate in the Exam Coaching Lab by answering a sample exam question. Your instructor will assign peers for you to review during Lesson 6.

Lesson 6: Introduction to Measurement
This lesson addresses the challenges in software testing and introduces you to software metrics.

Orienting activity: What Makes a Measurement?

Lesson introduction
You will learn about four key concepts:

- **Measurement**: the empirical, objective assignment of numbers to attributes of objects or events according to a rule derived from a model or theory with the intent of describing them.

- **Construct validity**: the basis for believing that a measure actually describes the attribute.

- **Surrogate measures**: a surrogate measure ascribes numbers to attributes but without the benefit of an underlying model or theory.

- **Measurement dysfunction**: people will optimize their behavior to improve the scores they get when they are measured.

The lesson presents two examples involving bug counts. One is the risk of using bug counts to measure the skill of testers. The other is the risk of using bugs-per-week as a measure of project progress, and in particular of using this in conjunction with a statistical model that is easily proved to be completely invalid.

Assignment
Review work to the Exam Coaching Lab submitted by your peers in the previous lesson.
Week 4 - Exam Week

Your only focus in the first part of this week is to prepare for the exam. You will have access to the full list of possible exam questions from the beginning of the course so that you have time to work on them and ask questions if anything seems unclear. You will also have access to a guide on how to prepare for the exam and several examples of how we grade exam answers.

You can draft sample exam answers and review those of your peers by Wednesday evening.

On Thursday, we will publish the exam, comprised of 3 short questions and 3 long questions. The exam is closed-book and you will have 3 days to work on it.
Interactive Grading

Interactive Grading is a technique that requires the student to participate in the grading of their work. The instructor has a chance to ask the student questions about the submitted work. The student has the opportunity to better demonstrate what they understood from the course material and to get helpful feedback directly from the instructor on what to improve and how.

How it works in this course
You will have one interactive grading session with your instructor after exam week. This is basically a 1:1 Skype call that will be scheduled ahead to find a time that’s suitable for both of you. During the call you will go together through your exam answers. This will be the first time your instructor will read your answers. They typically ask open-ended questions to help them understand what the student suggested in the work, like:

- Can you give me a real-life example of what you are describing?
- How would you answer this if I changed the question’s wording this way?
- Is there a simpler way to do the same thing?

This session is not a lecture. Instructors do their best to keep the student engaged and comfortable throughout the call. You will receive actionable feedback that can help you:

- understand the material better.
- communicate what you’ve learned more clearly.
- improve your “student-skills” such as studying and looking things up online.

This session is not compulsory. You can choose to have your exam graded privately, but know that you will be missing a key learning experience. Many BBST students declared that they found the interactive grading sessions to be a good use of their time.
Expectations and FAQ

This is an intensive course. We expect you to commit to a studying schedule, maintain activity on the course platform and meet the assignment deadlines.

**Time commitment.** Most students need to spend a minimum of 15 hours per week in this course. Some spend much more time. Few can expect to spend less time and still succeed in the course.

**Communication.** Check for course-related emails and announcements every day. If you have questions, ask them in the discussion board areas.

**Plan your schedule to meet the assignment deadlines.** This class is self-scheduled in that you schedule your own time to work on assignments. This course is not self-paced in that your instructors set the deadlines and thus, the pace for the class. You will have one or more assignments due each half-week. Plan to turn them in before the deadlines. You are free to work ahead as you have time available although, as a general rule, you won't want to get more than one or two weeks ahead because the interactions with your classmates become difficult to manage.

**Frequently asked questions**

*Do I need to login every day?*
No. As long as you submit your assignments in time, you have every chance to pass the course. You also get points for active participation in the discussions. If you work full-time, our advice is to plan your study schedule carefully and try to spend 2 hours of studying each day rather than cramming them in the weekends.

*Do I need to be online at specific times?*
No, except for the interactive grading session and the Q&A session (both optional and scheduled ahead). You can work on assignments and participate in discussions at any hour of the day. You will be able to schedule the interactive grading session with your instructor at a time that is suitable for both of you. We take into account the time zone difference of every student when scheduling the live Q&A session, so that everyone is able to participate.
*Do you offer certification?*
You will receive a certificate of completion signed by all of the instructors in your class. If you complete all three courses in the BBST series—Foundations, Bug Advocacy and Test Design—you will receive an additional, printed certificate.

*How do you grade?*
The assignments and the exam are graded from 0 to 2, conforming to our summative feedback system: 0 - no assignment; 1 - does not meet expectations; 2 - meets expectations. Quizzes are not graded.

"Final" grades in BBST classes are "successfully completed" or "has not successfully complete yet." We know that busy working professionals sometimes have to shift their priorities away from class. When this happens, "has not successfully complete yet" seems far more accurate than "fail." We don't use that in BBST classes.

*What if I cannot complete the course?*
In exceptional circumstances, we may be able to offer you a place in a future session of the course, based on availability.

*Are there any technical requirements to be met during class?*
We do have a list of course requirements such as having Skype and a good internet connection, nothing unusual or difficult to install.

*Do I need to purchase the course workbook separately?*
No. Altom will purchase the workbooks and send them out at no additional cost.

*Is there an additional cost for taking the exam?*
No.

*Will I have access to the course material after finishing the course?*
You will have access to the course platform for at least another month after the class ends.

If you have more questions, please email us at training@altom.ro
Preparing for Bug Advocacy

After completing BBST Foundations you will become eligible for the next course module—Bug Advocacy. Here, you will work on live bug reports of open-source applications. The skills you will practice are:

- investigating a behavior based on the available information.
- communicating your findings through arguments that motivate others to take the appropriate action – either fix it or close it/focus on something more important.

Bug Advocacy has a slightly more demanding workload than Foundations, but the nature of the assignments is more practical.

Interested? Register here for the next class

or email us at training@altom.ro with any questions.