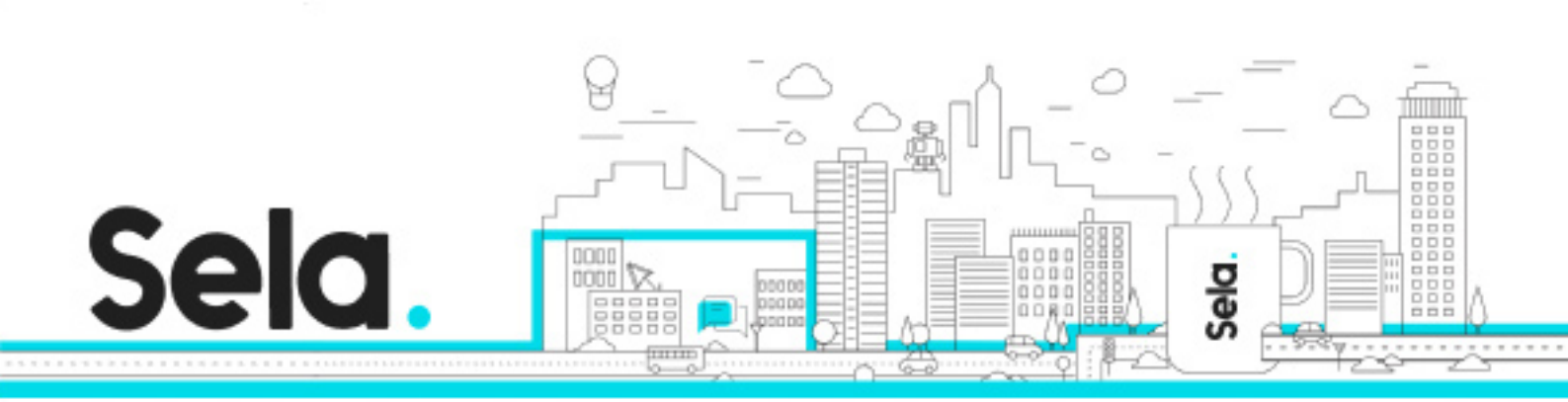
Sela.

TstDD

Test Driven Development

college@sela.co.il 03-6176666



Test Driven Development

TstDD - Version: 1

4 days Course

Description:

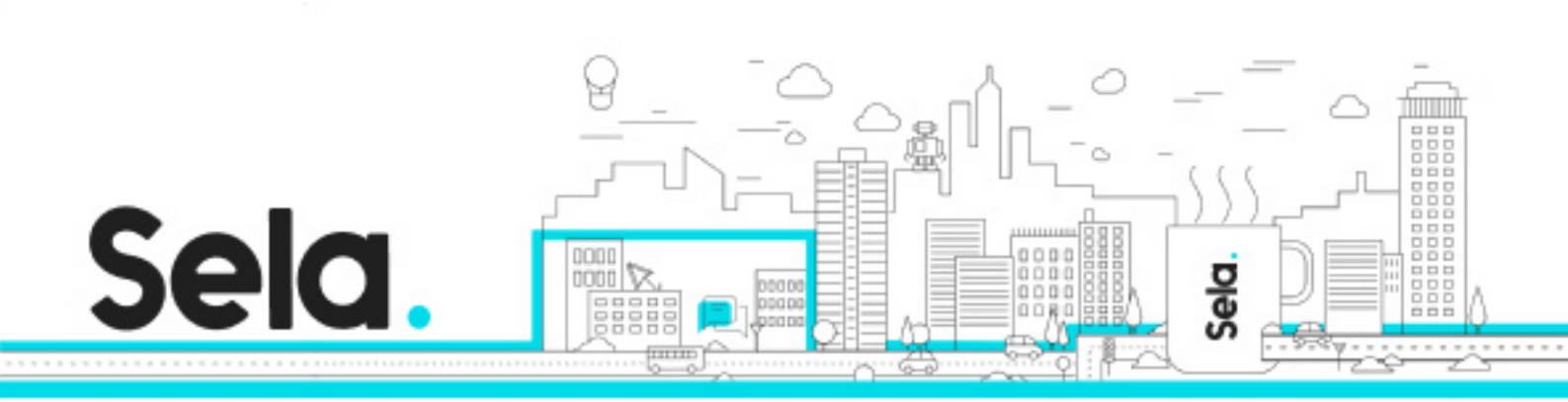
Testing forms an integral part of the modern software development flow. From customer-facing acceptance tests to code-focused unit tests, automated testing is part of the fabric of a modern build process and deployment pipeline. But it's not enough that there are some tests: to be a help, not a hindrance, tests need to communicate not just verify, and testing needs to be a development habit, not an afterthought.
In this online course, you will learn about good unit tests (GUTs) and test-driven development (TDD) by seeing what it takes, putting them into practice, and reviewing what you've learned. What practices support readable and maintainable tests? What test pitfalls hold developers and products back? How do you make testing fun and not a chore? Join us to answer these questions and more.
The examples and hands-on exercises will use Python and pytest, but you don't need to be a Python guru to take part. Discussion and examples will also take in other languages and frameworks.

Intended audience:

Developers, team leaders and architects who code and want to start unit testing or improve their existing practices

Prerequisites:

Participants should have either good working knowledge of Python or some knowledge of Python along with skills in other languages



Objectives:

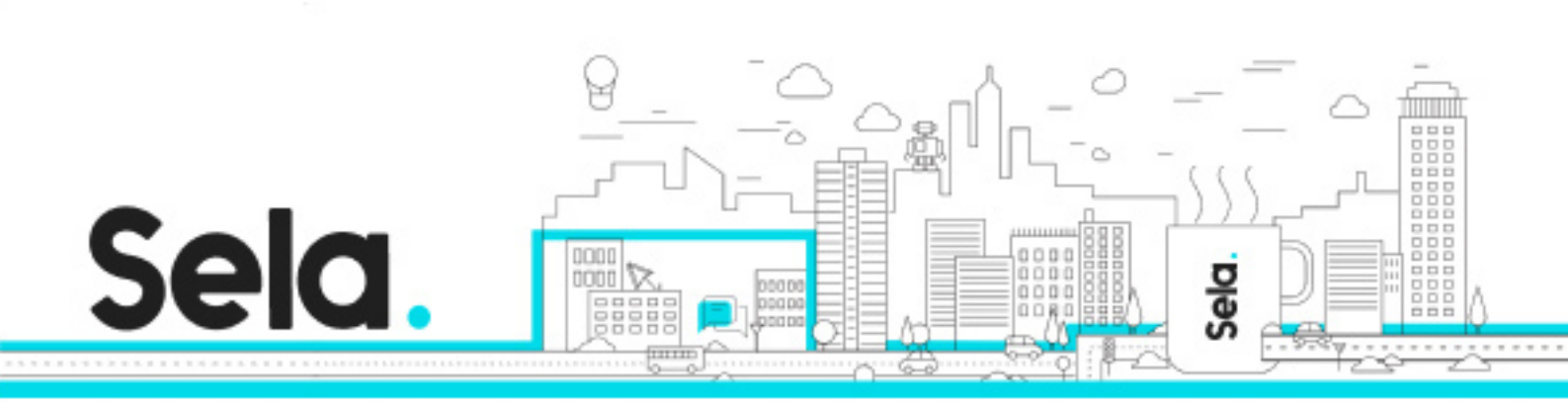
Appreciate the value of unit testing to developers
Understand how unit testing compares with other testing and quality-focused techniques and tools
Learn what good unit tests (GUTs) look like and how to write them
Reason about different testing workflows for developers, from defect-driven testing (DDT) to test-driven development (TDD)
Be able to put test-driven development into practice
Apply different techniques to legacy code

Topics:

º What do we mean by unit tests?

º How do unit tests fit into the testing and development landscape?

º What do GUTs look like?



º The elements of TDD

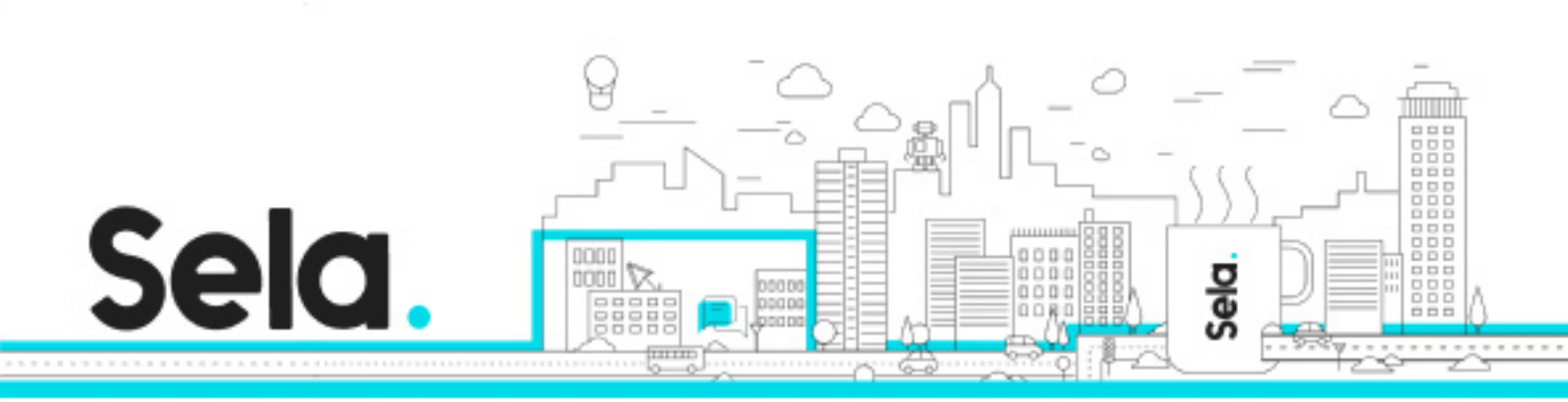
º Core pytest features

º Data-driven tests and how to choose test data

º Test naming and nesting

º The anatomy of test cases and test suites

º Plain ol' unit testing (POUT), defect-driven testing (DDT), and iterative test-last (ITL)



º Identifying and adjusting overfitting and underfitting tests

º Reasons writing tests can be hard, and how to respond

º Mocks, stubs and other test doubles

º When to use values instead of test doubles

º Testing and legacy code