Behavior Focused Test-Driven Development

Built-in quality is a key practice in agility. Testability is a crucial aspect of quality. This course teaches developers how to write behavior-focused unit tests and to develop incrementally using those tests. It emphasizes how these tests reflect the system requirements. Refactoring legacy code and adding unit tests to that legacy code is also explored. The course emphasizes hands-on practice in writing unit tests, creating mocks, and refactoring.

TARGET AUDIENCE:

Software developers in Java, C#, or C++ who want to learn to create quality code.

LEARNING OBJECTIVES:

- How to be "test-driven"
- How to write behavior focused unit tests
- When and how to use mock objects (test doubles)
- How to refactor legacy code

INSTRUCTOR

Ken Pugh, author of *Lean-Agile Acceptance Test-Driven Development: Better Software Through Collaboration*.

OUTLINE:

- Introduction
    - How TDD helps with quality
    - Creating tests first
    - Running the framework
- Unit test introduction
    - Naming tests
    - Trying out the framework
- TDD Exercise
    - Organizing unit tests
    - A TDD exercise
- Mocking (Test doubles)
    - When to use
    - How to create
    - A mocking exercise
    - Ways to minimize mocking
- Refactoring
    - Common refactorings
    - A legacy code refactoring exercise
- Your TDD
    - Large exercise in TDD

ATTENDEE MATERIALS

Workshop materials are provided at the start of the class

ROOM SETUP AND EQUIPMENT

One computer for every two students, loaded with necessary software.
Flip chart and whiteboard for the instructor.
A projector with screen.

PREREQUISITES

Experience in C#, Java, or C++

COURSE LENGTH

3 days

MAXIMUM

24 students

PROVIDER

Ken Pugh, Inc. 732 Ninth Street #695, Durham, NC 27705.

CONTACT

Leslie Killeen 919-490-6335

leslie@kenpugh.com