

Activity 4-3 Introduction to Test-Driven-Development

Interjecting segments of the TDD video, part 2 [Link¹](#).

Intro to TDD

Video segment: intro 00:50-3:00

Video segment: The bowling game: 3:00-6:00

- How many frames are there in a bowling game?
- How many rolls are there on each frame?
- What is a spare? What is a strike? How are they scored?
- What is the maximum score in a bowling game?

Video segment: The bowling game implementation plan: 6:00-8:30

- What are the classes involved in the planned design, and what are their relationships?

Video segment: TDD approach to the bowling game: 8:30-42:30

- “What test do I need to write, that will force me to write the code I know I want to write?”
- “When you must write real code, write the simplest code that you possibly can.”
- “You must see your tests fail first, before you make them pass.”

RED PHASE Focus on writing a failing test.

GREEN PHASE Write the code that makes the failing test pass.

REFACTOR PHASE Refactor your code before moving to a new test.

- “Misplaced responsibility”
- “Tests are as important as production code and you keep your tests clean!”

Video segment: objections to TDD: 42:30-60:00

- “Tests are more code you have to write”
- “Managers might object”

¹https://learning.oreilly.com/videos/clean-code/9780134661742/9780134661742-CODE_01_05_02

- “Refactoring is rework, it would be better to write the code right the first time”
- “Maintaining the tests could become time-consuming”
- “No tests can prove the program fully correct”
- “It doesn’t matter when you write the tests”
- “How can we work with legacy code?”
- “How do you test the GUI?”
- “How do you test databases?”
- “Programmers are not testers”
- “Test-driven development is too hard”

Video segment: Professionalism: 60:00-69:00

- Double-entry bookkeeping and sensitive industries
- QA should find nothing
- 100% code coverage is the goal

Conclusions: 69:00-75:00