

✔ **Congratulations! You passed!**

Grade received **100%** To pass 80% or higher

Go to next item

1. What does the Python debugger allow debugging failing tests?

1 / 1 point

- ☒ Stepping through the code execution flow
- ☐ Enabling exception handling
- ☐ Reporting the root cause analysis
- ☐ Parallelizing test execution

✔ **Correct**
Correct. PDB lets you step through code execution to diagnose issues.

2. How can fixtures improve managing test resources?

1 / 1 point

- ☐ By enhancing assertion output
- ☐ Through exception handling
- ☒ Automatic setup/teardown
- ☐ Command line integration

✔ **Correct**
Correct. Fixtures handle automatically setting up and tearing down test resources.

3. What does the pytest "last failed" option provide?

1 / 1 point

- ☐ Detailed failure output
- ☒ Re-running only failed tests
- ☐ PDB integration
- ☐ Parallel execution

✔ **Correct**
Correct. Last failed reruns only tests that failed in the previous run.

4. What is a benefit of Pytest's test failure output?

1 / 1 point

- ☒ Precise failure location reporting
- ☐ Automated root cause analysis
- ☐ Standardized assertions
- ☐ Detailed resource leaks

✔ **Correct**
Correct. Pytest failure output pinpoints the exact line causing the test failure.

5. How could the Python debugger help diagnose a failing test?

1 / 1 point

- ☐ By mocking test data
- ☐ Through code coverage reporting
- ☒ Inspecting variable values
- ☐ By parametrizing tests

✔ **Correct**
Correct. PDB allows inspecting variables' values during test execution to understand the failure cause.