



Course

Progress

Course

>


Readings/Videos


>


Reading 3: Testing


>


Questions


Previous






































Next

Questions

Bookmark this page

regression testing

1/1 point (graded)

Which of the following best defines regression testing?

- ☒ Changes should be tested against all inputs that elicited bugs in earlier versions of the code.

☐ Every component in your code should have an associated set of tests that exercises all the corner cases in its specification.

☐ Tests should be written before you write the code as a way of checking your understanding of the specification.

☐ When a new test exposes a bug, you should run it on all previous versions of the code until you find the version where the bug was introduced.



Explanation


The first choice is correct because regression test cases are inputs that once triggered bugs, and the idea of regression testing is to test new changes to the code against these test cases.

The second choice is a good practice -- black-box unit testing -- but it isn't regression testing.

The third choice is also good practice -- test-first programming -- but again, not regression testing.

The fourth choice may be useful for tracking down a difficult bug. It is a variant of delta debugging, and is never done as exhaustively as *all* previous versions, but more typically with a binary search. Still, it is not regression testing.

Submit

Show Answer

Answers are displayed within the problem

running automated tests

1/1 point (graded)

Which of the following are good times to rerun all your JUnit tests? Check all that apply.

- ☒ Before doing git add/commit/push

☒ After rewriting a function to make it faster

☒ When using a code coverage tool

☒ After you think you fixed a bug



Explanation


Pushing your code to git sends it to the rest of your team, so rerun the tests first to make sure you're not pushing broken code.

Rewriting a function may introduce bugs, so rerun your tests to find them.

Rerunning tests is an essential part of using a code coverage tool, because you want to see the code lines that your tests don't reach.

Fixing a bug is a change to your program, and you should rerun your tests after every change.

Submit

Show Answer

Answers are displayed within the problem

testing techniques

1/1 point (graded)

Which of these techniques are useful for choosing test cases in test-first programming, before any code is written? Check all that apply.

☒ black box

☐ regression

☐ static typing

☒ partitioning

☒ boundaries

☐ white box

☐ coverage



Submit

Show Answer

Correct (1/1 point)

Open Learning Library

About

Accessibility

All Courses

Why Support MIT Open Learning?

Help

Connect

Contact

Twitter

Facebook