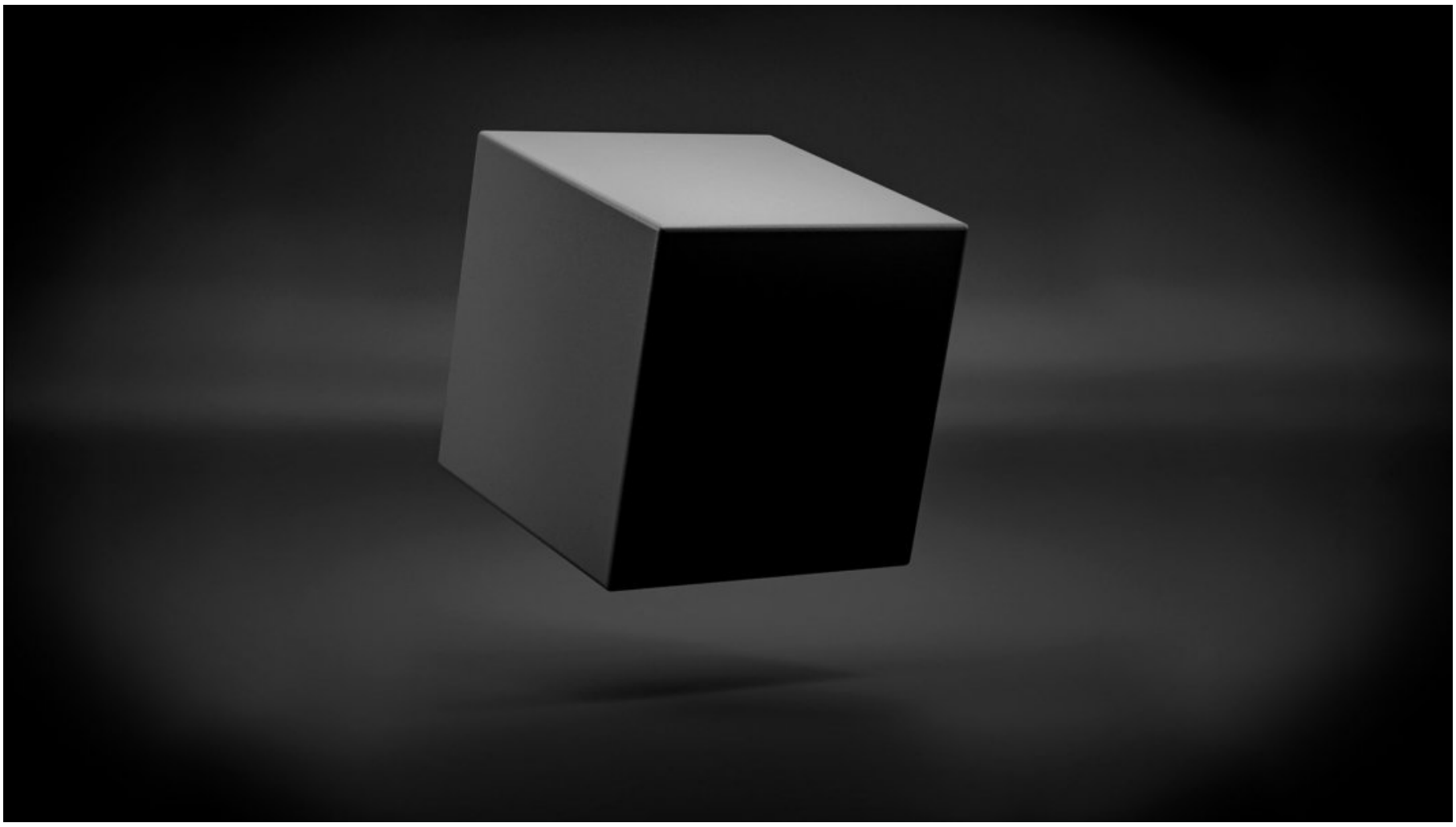
Intro

- Software Testing is the most important step of the development cycle and is considered essential knowledge for any software position, but is often overlooked when preparing for interviews.
- Testing questions are always fair game as a quick interview trivia question, so here's a quick cheat-sheet of the most popular terms for you to be prepared the next time you're asked about testing.

White Box vs. Black Box Testing



> Main Difference: The visibility of the internal design

White Box Testing

Quick Summary: The internals of the item tested is known to the tester

Also known as: Internal Testing, Clear Box Testing

Important Facts

- The tester knows how the code works and what it does
- Lower level (more rigorous) testing
- Time consuming

Examples

- Testing your own code after your write it (unit testing)
- Testing every if-else case in your code (path testing)

Black Box Testing

Quick Summary: The internals of the item tested is unknown to the tester

Also known as: External Testing, Closed Testing

Important Facts

- The tester does not know how the code works but may know what it does
- Higher level (less rigorous) testing
- Time efficient

Examples

- Testing function outputs to a series of inputs
- General usability (speed, security, reliability, non-functional testing)

Grey Box Testing

Quick Summary: Black Box Testing but with some White Box information

Also known as: Gray Box Testing

Important Facts

- The tester knows how the code works through its design specification and documentation such as UML and database diagrams
- Preserves the boundary between testers and developers
- Compared to White Box testing, the tester does not have access to the source code
- Compared to Black Box testing, the tester has more information about the item itself

Examples

- Testing an API using its documentation
- Ensuring software meets its design requirements

Functional Testing

Does the code run as expected without any bugs?

Unit Testing

Quick Summary: A test written by developers to ensure that a "unit" of code (usually a module or a function) behaves as intended

Also known as: Component Testing

Important Facts

- Detect bugs early (don't ship your code if it's already broken!)
- Isolate a unit to its own independent environment to reveal and remove extra dependencies
- Can be performed both manually/automatically with a test framework
- White/Black/Grey box unit testing

Examples

- Using basic assertions to ensure that the program state is behaving as expected
- Using a test framework such as JUnit to write code to simulate automated test inputs for a function

- Be informed of the testing frameworks available to the language you will be working in to score some extra points with the interviewer

Integration Testing

Quick Summary: Testing the compatibility of units

Also known as: Component/System Integration Testing

Important Facts

- Combine units and test them together
- Usually performed after unit testing

1. Component (unit) Integration Testing
    - Focus on interactions between components
2. System Integration Testing
    - Focus on interactions between systems of components

Examples

- Ensuring that nested React components in a website behave correctly (component integration testing)
- Interacting with an API using your program without errors (system integration testing)

Regression Testing

Quick Summary: New code changes do not break old code

Important Facts

- Retesting code requires prioritization of test cases
    - Testing is expensive! Choose the most important tests to maximize test coverage and efficiency
- Best if automated using an automated testing tool such as Selenium
- Update the test cases frequently for new code changes

Examples

- Retesting after writing new code
- Pulling new code from the development branch

Smoke Testing

Quick Summary: Ensure that the software has basic functionality before testing

Important Facts

- The most important functions should be working
    - If the device does not catch fire and start smoking, proceed with further testing
- Usually black box testing (quick and efficient)

Examples

- The first test before running other, more intensive tests on new code (Integration, Regression, System)

System Testing

Quick Summary: Testing if the system meets design requirements

Important Facts

- Usually performed after Integration Testing and evaluates the end software product
- The most important step of testing, many different types of functional and non-functional tests are performed at this stage
- Guarantee that all the components work together and the system functions as expected

Examples

- Smoke Testing
- Regression Testing
- Usability Testing
- Performance Testing
- Security Testing

Let's test your knowledge. Is this statement true or false?

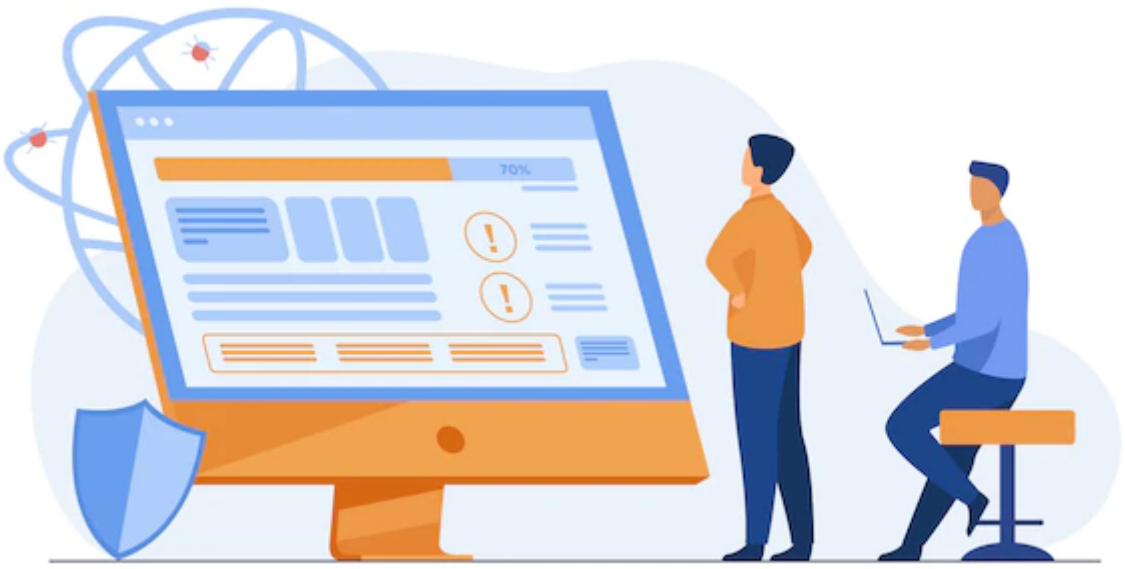
Unit testing can be both white box and black box testing, depending on how you implement the test cases.

Press true if you believe the statement is correct, or false otherwise.

TRUE

FALSE

Non-Functional Testing



Improving the code's performance and usability

Acceptance Testing

Quick Summary: Ensure that the software will be received well by users/consumers

Also known as: User Acceptance Testing

Important Facts

- Conducted at the end of the software lifecycle for sequential models such as Waterfall
- Conducted at the end of each iteration for iterative models such as Agile
- Black Box Testing

Examples

- Alpha testing an unreleased game within the company (Internal Acceptance Testing)
- Beta testing an unreleased game by sending copies to popular streamers (External Acceptance Testing)

Usability Testing

Quick Summary: Ensure that the software is intuitive and easy to use

Also known as: User Experience/UX Testing

Important Facts

- Find frequent use cases and make them more efficient
    - Condense long processes into simple steps
- Detect user misunderstandings and operation errors
- Black Box Testing

Examples

- Observing a user complete a set of tasks, noting their thoughts as they use the software

Performance Testing

Quick Summary: Stress testing the speed and stability of a system

Also known as: Responsiveness/Stability Testing

Important Facts

- Find the system bottlenecks to optimize them
- Monitor system performance and endurance under different workloads
    - The system should perform as expected under heavy loads

Examples

- Flooding an e-commerce website with online orders until it crashes (Stress Testing)
    - Pushing the system beyond its limits
- Having many users visit and browse your website frequently (Endurance Testing)
    - Continuous workload
- Handling a massive influx of online shoppers during a sneaker drop (Spike Testing)
    - The workload is abruptly increased

Security Testing

Quick Summary: Expose system vulnerabilities

Also known as: Vulnerability Testing

Important Facts

- Build user trust with robust security
- Analyze network, system, client-side, and server-side security

Examples

- Examining network policies and external data transfers (Network Security)
- Assess the software that the program depends on (System Security)
- Ensuring that local manipulation in the client or browser cannot occur (Client-Side Security)
- Confirming that the technology used on the back-end server can withstand intrusion (Server-Side Security)

Try this exercise. Click the correct answer from the options.

Which type of testing should be used in case of an interface error?

Click the option that best answers the question.

White box testing

Unit testing

Integration testing

Performance testing

Test Driven Development

Agile Development

- An iterative software development lifecycle
- Very flexible, client-oriented process
- Performs testing very frequently after every iteration of the product
    - Continuous feedback to improve software

Agile Testing

- Tests planned on whim and test processes are less structured
- Test cases are continually changing to meet changing demands
- Has a focus on software that works

Build your intuition. Fill in the missing part by typing it in.

Testing if a landing page of a website loads within few seconds is an example of ___ testing.

Write the missing line below.

Write missing line

Submit

Reveal answer

Testing Questions

Some sample interview questions with example answers. Many of these questions are open and have many answers; try to come up with your own!

Name some testing libraries that you have used and your experience with them

Some popular ones include:

Python

- pytest
- unittest

Java

- JUnit
- TestNG

JavaScript

- MochaJS
- Jest
- Jasmine

Other

- Selenium
- Cucumber

- Capybara

> I designed and wrote scripts to automate testing for an e-commerce web application using the Selenium WebDriver library in Java

What are the 4 main levels of testing?

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

- Regression Testing can be performed in-between any of these levels

What are the key elements of good test cases?

Simplicity

- Be clear and concise. Write not for yourself, but for the person after you.

Maximum coverage

- We want to minimize our test cases while maximizing the chance to find a defect

Repeatability

- The test case should always generate the same results, no matter the environment

What is the difference between software testing and software verification?

- Software Testing is validating whether software meets the consumer's needs through the design of test cases
  - This cheat sheet focuses on Software Testing and its applications
- Software Verification is more rigorous; formally proving using a proof system whether software is "correct" to its specification
  - Not all software can be formally verified

What kind of testing would you perform on a retractable ballpoint pen during the design process to verify its quality?

Try to be specific and creative! Think about how all the individual components interact with each other.

Unit Testing

- Does the ink tube leak?
- Is the pen's body durable?
- Does the ink tube retract properly?

Smoke Testing

- Can the pen write?

Integration Testing

- Do all the pen parts fit inside the body?
- Does the retraction mechanism interfere with the writing mechanism?

Regression Testing

- A new click-spring mechanism is introduced to the pen, does this affect pen retraction?
- After assembling the pen, does the pen leak?

Acceptance Testing

- Is the pen comfortable?
- Does the pen write smoothly?

Usability Testing

- Can users easily figure out how to retract the pen?
- Will users recognize that it is, in fact a pen?

How many testers does it take to change a light bulb?

- None. Testers do not fix problems - they just find them.

Build your intuition. Click the correct answer from the options.

Which kind of testing does not require you to design test cases?

Click the option that best answers the question.

- Black box testing
- Usability testing
- Unit testing
- Smoke testing

Last Tutorial
Navigate to previous step

Next Tutorial
Navigate to next step

Programming Categories

- Systems Design Interview Questions
- Trees Interview Questions

Popular Lessons

- All Courses, Lessons, and Challenges
- Data Structures Cheat Sheet
- Free Coding Videos
- Bit Manipulation Interview Questions
- Javascript Interview Questions
- Python Interview Questions
- Java Interview Questions
- SQL Interview Questions
- QA and Testing Interview Questions
- Data Engineering Interview Questions
- Data Science Interview Questions
- Blockchain Interview Questions
- Boolean Algebra and Logic Gates
- Data Modeling Cheat Sheet
- Pandas Architecture (CRM)
- BFS vs. DFS: Understanding Breadth First Search & Depth First Search
- Arrow Functions in Javascript/Typescript

Jump To

Interactive Mode