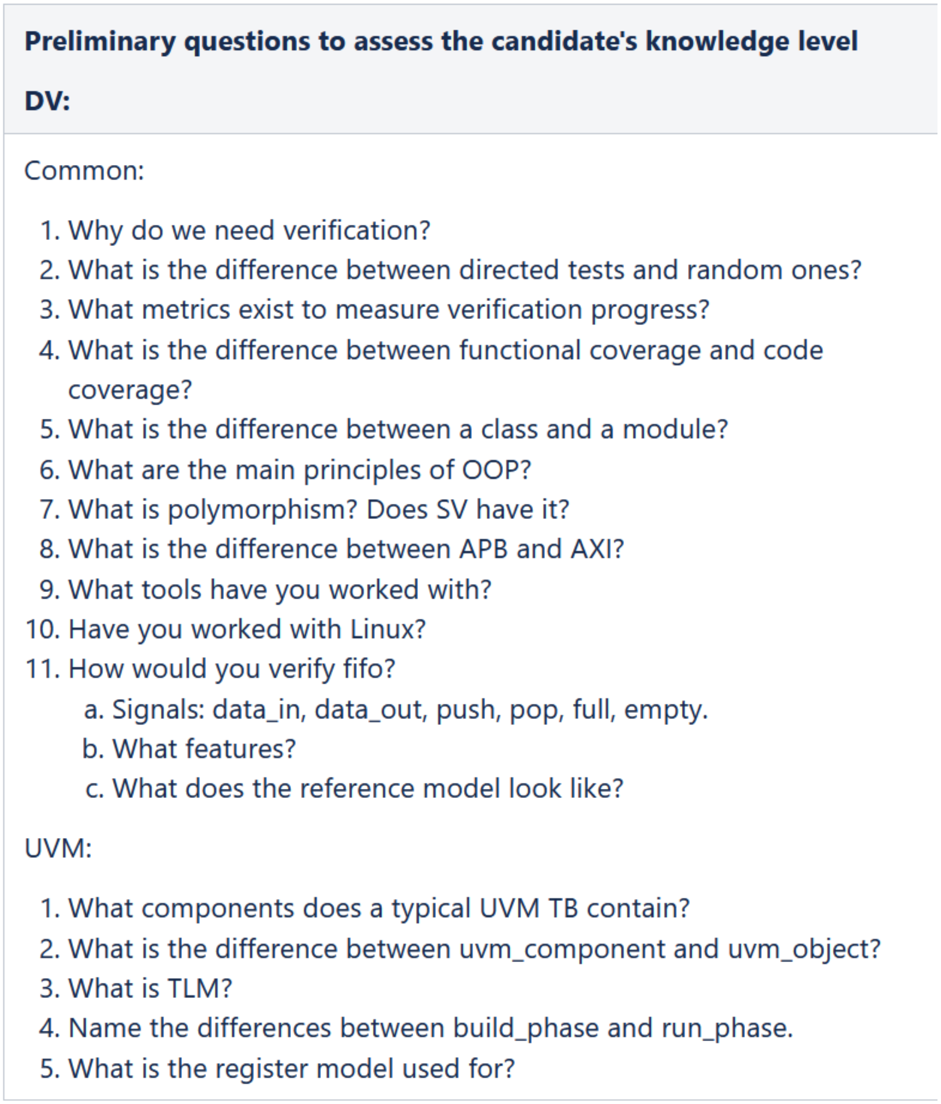


Preliminary questions to assess the candidate's knowledge level

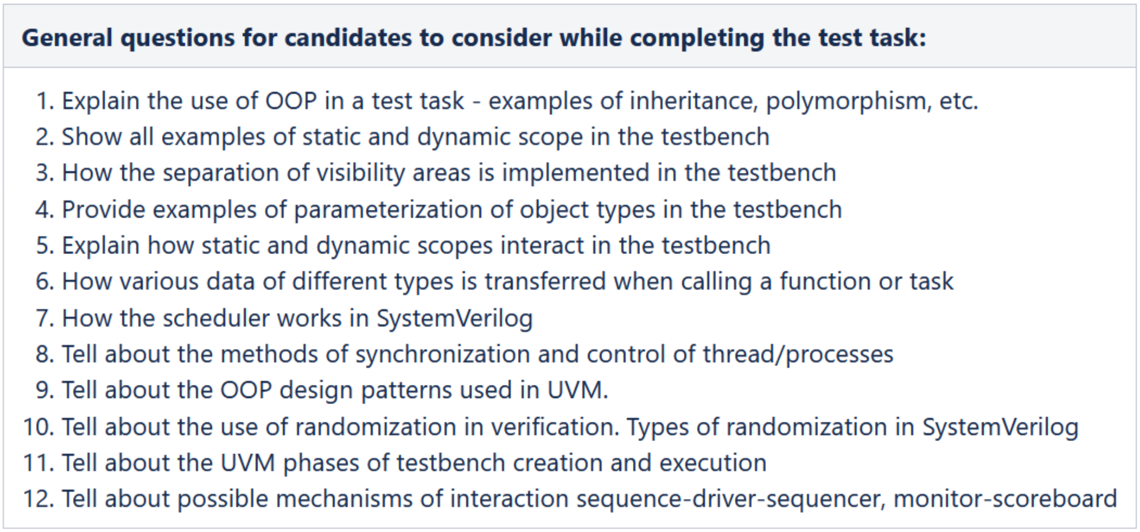
DV:

Common:

1. Why do we need verification?
2. What is the difference between directed tests and random ones?
3. What metrics exist to measure verification progress?
4. What is the difference between functional coverage and code coverage?
5. What is the difference between a class and a module?
6. What are the main principles of OOP?
7. What is polymorphism? Does SV have it?
8. What is the difference between APB and AXI?
9. What tools have you worked with?
10. Have you worked with Linux?
11. How would you verify fifo?
    a. Signals: data_in, data_out, push, pop, full, empty.
    b. What features?
    c. What does the reference model look like?

UVM:

1. What components does a typical UVM TB contain?
2. What is the difference between uvm_component and uvm_object?
3. What is TLM?
4. Name the differences between build_phase and run_phase.
5. What is the register model used for?



General questions for candidates to consider while completing the test task:

1. Explain the use of OOP in a test task - examples of inheritance, polymorphism, etc.
2. Show all examples of static and dynamic scope in the testbench
3. How the separation of visibility areas is implemented in the testbench
4. Provide examples of parameterization of object types in the testbench
5. Explain how static and dynamic scopes interact in the testbench
6. How various data of different types is transferred when calling a function or task
7. How the scheduler works in SystemVerilog
8. Tell about the methods of synchronization and control of thread/processes
9. Tell about the OOP design patterns used in UVM.
10. Tell about the use of randomization in verification. Types of randomization in SystemVerilog
11. Tell about the UVM phases of testbench creation and execution
12. Tell about possible mechanisms of interaction sequence-driver-sequencer, monitor-scoreboard

Resources to study SV, UVM

On SV: standard and examples from the Internet. There is a lot of information on the Internet.
http://ecee.colorado.edu/~mathys/ecen2350/IntelSoftware/pdf/IEEE_Std1800-2017_8299595.pdf
On UVM: Take as a basis examples from the "UVM cookbook" uvm-cookbook.pdf, which can be downloaded here:
https://verificationacademy.com/resource/68003?download_refer=/cookbook
You can also refer to http://simhard.com Special course (Verification of digital circuits)
The book https://www.springer.com/gp/book/9781461407140 can be "purchased" on gen lib rusek (I hope the candidate with a hint will do the right thing). "Buy" the 3rd edition (2012)
And here is the standard for UVM:
https://www.accellera.org/images/downloads/standards/uvm/uvm_users_guide_1.1.pdf
The UVM Primer - Salemy Ray
A Practical Guide to Adopting the Universal Verification Methodology - Sharon Rosenberg
Coverage Cookbook - Verification Academy
Writing Testbenches using SystemVerilog - Janick Bergeron