

UVM PROGRAM WRITING CHECKLIST

//-----//

Understand DUT First

- ✓ Interface signals
- ✓ Clock & reset behavior
- ✓ Input/output timing
- ✓ What to verify (functional intent)

Never start UVM without DUT clarity.

//-----//

1. Create Interface

- ✓ Define all DUT signals
- ✓ Add clocking block (optional but recommended)
- ✓ Use modports (driver / monitor)

```
interface dut_if(input logic clk);
```

//-----//

2. Transaction (uvm_sequence_item)

- ✓ Extend uvm_sequence_item
- ✓ Declare stimulus fields as rand
- ✓ Add constraints if required
- ✓ Register with factory

Checklist:

- ✓ rand variables
- ✓ new() constructor
- ✓ uvm_object_utils

//-----//

3. Sequence (uvm_sequence)

- ✓ Extend uvm_sequence
- ✓ Create transaction
- ✓ Randomize
- ✓ Start & finish item()

Checklist:

- ✓ start_item()
- ✓ randomize()

- ✓ finish_item()

//-----//

4. Sequencer

- ✓ Declare as transaction_sequencer type
- ✓ No custom logic needed initially

Checklist:

- ✓ Correct transaction type
- ✓ Factory registration

//-----//

5. Driver

- ✓ Extend uvm_driver
- ✓ Get transaction from sequencer
- ✓ Drive DUT signals
- ✓ Use clock synchronization

Checklist:

- ✓ seq_item_port.get_next_item()
- ✓ Drive signals
- ✓ item_done()
- ✓ Reset handling

//-----//

6. Monitor

- ✓ Extend uvm monitor
- ✓ Sample DUT signals
- ✓ Create transaction from observed data
- ✓ Send to scoreboard using analysis port

Checklist:

- ✓ uvm_analysis_port
- ✓ Sample at correct clock edge
- ✓ No driving here

//-----//

7. Scoreboard

- ✓ Extend uvm scoreboard

- ✓ Receive transactions
- ✓ Implement reference model
- ✓ Compare expected vs actual
- ✓ Report PASS/FAIL

Checklist:

- ✓ uvm_analysis_imp
- ✓ Clear comparison logic
- ✓ uvm_error on mismatch

//-----//

8. Agent

Instantiate:

- ✓ Driver
- ✓ Sequencer
- ✓ Monitor

Configure:

- ✓ Active / Passive

Checklist:

- ✓ Correct connections
- ✓ Driver ↔ Sequencer connected
- ✓ Factory registration for agents

//-----//

9. Environment

- ✓ Instantiate agents
- ✓ Instantiate scoreboard
- ✓ Connect monitor → scoreboard

Checklist:

- ✓ All components created
- ✓ Analysis connections done
- ✓ Optional: coverage collector

//-----//

10. Test

- ✓ Extend uvm_test
- ✓ Create environment
- ✓ Start sequence
- ✓ Control simulation end

Checklist:

- ✓ raise_objection
- ✓ Start sequence
- ✓ drop_objection

//-----//

11. Top Module

- ✓ Instantiate DUT
- ✓ Instantiate interface
- ✓ Set interface via uvm_config_db
- ✓ Call run_test()

Checklist:

- ✓ Virtual interface set
- ✓ run_test() called

//-----//

12. Reporting & Debug

Use:

- ✓ uvm_info
- ✓ uvm_warning
- ✓ uvm_error
- ✓ uvm_fatal

Checklist:

- ✓ Meaningful TAG names
- ✓ No excessive prints

//-----//

11. Final Sanity Checks

- ✓ Simulation runs without fatal error
- ✓ Sequence executed
- ✓ Monitor captured data
- ✓ Scoreboard reported result

- ✓ Coverage goals met
- ✓ Factory overrides work

//-----//

QUICK FLOW MEMORY LINE

Transaction → Sequence → Sequencer → Driver → DUT → Monitor → Scoreboard