Module: UVM-1
Section: UVM Concepts **Task:** UVM Agent

UVM Agent

Task 1

- **Driver Methods:**

  In UVM, besides **get_next_item**, another primary method a driver can use to get a sequence item from the sequencer is **try_next_item**. Here's a quick comparison of both:

  1. **get_next_item()**: This method blocks the driver until the sequencer has an item available. It’s ideal when you want the driver to wait for the next item and ensure that it is processed in sequence.
  2. **try_next_item()**: This is a non-blocking method that checks if there’s an item available without waiting. If an item is available, it fetches it; otherwise, it moves on. This method is useful in situations where the driver needs to perform other tasks if no item is available immediately.

- **Example Code:**

```
class my_driver extends uvm_driver #(my_sequence_item);

  `uvm_component_utils(my_driver)

  function new(string name, uvm_component parent);
    super.new(name, parent);
  endfunction

  virtual task run_phase(uvm_phase phase);
    my_sequence_item req;

    phase.raise_objection(this);

    // Using try_next_item() to check for a sequence item without blocking
    forever begin
      if (seq_item_port.try_next_item(req)) begin
        `uvm_info(get_type_name(), $sformatf("Processing item with data: %0d", req.data),
UVM_LOW)

        seq_item_port.item_done(); // Indicate item processing is complete
      end
      else begin
```

```
        `uvm_info(get_type_name(), "No item available, performing other tasks...", UVM_LOW)
        // Perform other tasks or wait
        #10ns; // Example of other tasks
      end
    end

    phase.drop_objection(this);
  endtask

endclass
```