**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