Assigned: *02/28/2019* **Due:** *03/08/2019* **Total Points:** *100*

1. (**10 Points**) What is "phasing" in UVM? How do objections help in UVM phasing?
2. (**10 Points**) What are virtual interfaces in SystemVerilog? Where are virtual interfaces useful in a SystemVerilog UVM testbench?
3. (**10 Points**) When writing the SystemVerilog code for a UVM class, which phases can be defined as functions and which as tasks? Justify your answer.
4. (**15 Points**) What does it mean for a phase to be top-down or bottom-up? Which UVM phases are top-down / bottom-up? Explain why they apply the corresponding strategy.
5. (**15 Points**) Consider an agent with three components - a sequencer, a driver and a monitor. What are the roles of each of these components? Can the monitor be removed and the driver be used to provide similar functionality? If yes, why would that not be a good idea?
6. (**10 Points**) What is the advantages and differences of ‘uvm_component_utils() And ‘uvm_object_utils()?
7. (**30 Points**) Consider a 32-bit LU (logical unit) as the DUT. The design specs of the DUT are as follows:

   (a) Inputs - A[31:0], B[31:0], clk, opcode[1:0]

   (b) Outputs - OUT[31:0]

   (c) Operation - The opcode decides what operations would be performed.

   - 2’b00 - A NAND B
   - 2’b01 - A NOR B
   - 2’b10 - A XOR B
   - 2’b11 - Invalid input

   (d) It is a non-pipelined design with a latency of 1 cycle.

   An RTL design, developed on the above specs, is black-boxed and given to you for testing. How would you go about creating a UVM testbench to test this design for bugs? In your solution, do mention about the structure/hierarchy of the testbench, constraints required, suggested coverage measurement, types of test sequences that can be used and approach for checking the correctness of the DUT.

   If the design is now replaced with a pipelined version of the DUT which has a latency of 3 cycles with a throughput of 1 ins/cycle, what changes will you have to do in the testbench?