1. What is the problem with below code snippet?

task generator\_example(int a);

BustTran b;

b=new();

repeat(n) begin

b.addr = $random();

$display(“Sending addr=%h”,b.addr);

transmit(p);

end

endtask

* The repeat(n) loop is using n as the number of iterations, but n is not defined anywhere in the task. You either need to pass n as an argument to the task or define it within the task.
* The transmit(p); statement references a task called transmit with an argument p, but transmit is not defined in the provided snippet.
* The BustTran class is used, but it isn't defined in the provided code snippet. If BustTran is a custom class, you need to define it somewhere in your code.
* The transmit(p); line uses a variable p, but this variable is not defined anywhere in the code.

1. Output of below code:

class constraint example:

rand bit[15:0] a,b,c;

constraint c1 {0<a<b<c;}

endclass

* a will be a random value greater than 0.
* b will be a random value greater than a.
* c will be a random value greater than b.

1. Output of below code:

initial begin

$display(“@%0d: start fork…join example”,$time);

#10 $display(“@%0d:sequential after #10”,$time);

fork

$display(“@%0d:parallel start”,$time);

#50$display(“@%0d:parallel after #50”,$time);

#10 $display(“@%0d:parallel after #10”,$time);

begin

#30 $display(“@%0d: sequential after #20”,$time);

#10 $display(“@%0d: sequential after #10:,$time);

end

join

$display(“@%0d: after join”,$time);

#80 $display(“@%0d: final after #80”,$time);

end

Output

@0: start fork...join example

@10: sequential after #10

@0: parallel start

@50: parallel after #50

@10: parallel after #10

@30: sequential after #20

@40: sequential after #10

@50: after join

@130: final after #80

* The fork...join structure allows the execution of parallel tasks. The sequential block (begin...end) inside the fork will execute in parallel with the other fork tasks.
* The delays (#10, #50, #30, etc.) will determine when the respective $display statements are printed.