1. What is the problem with the following code?

class pkt\_len;

rand byte pkt1\_len,pkt2\_len;

constraint total\_len{

pkt1\_len+pkt2\_len==64;

}

endclass

Depending on the randomization process, there might be issues in achieving this exact sum,

depending on the randomizer and the constraints.

1. What constraints will be generated by this code?

class data\_size;

rand reg[31:0] data[];

Constraint data\_size{data.size inside{[1:10]};}

endclass

* data.size inside { [1:10] }: This constraint ensures that the size of the array data (i.e., the number of elements in the array) must be between 1 and 10, inclusive.
* The constraint will force the randomization process to generate an array data[] where the number of elements (i.e., data.size) is between 1 and 10.

1. What is the output?

class trans;

bit[31:0] data;

constraint c\_rand\_data {data>0; data<11;}

endclass

* t.data will be randomly assigned a value between 1 and 10, inclusive (since the constraints are data > 0 and data < 11).
* The output will be a random integer in the range of 1 to 10.