1. Write a snippet constituting a class with three data members of integral type, and take a subroutine to display the value of the properties.
2. Write a snippet constituting a class having two data members, take multiple objects assign different values to the different objects and delete one of the objects before finish.
3. Write a subroutine to add two properties and assign it to third property.
4. Write a snippet to track the number of times class object has been created.
5. Write a piece of code with class having some properties and take values from user through method.
6. Write a code to access the local property of a class from outside the class.
7. Can we access local property of parent class inside child? If yes how can we restrict to access the parent class property from outside the class,
8. Write a class with two objects of same class and perform the object assignment of one class object handle to another handle.
   * Update the property using both handles and observe the changes accordingly.
   * Deallocate the assigned object.
9. Write a class with different properties take an object, copy one object to another object.
10. Write a snippet constituting a parent class, a child class with set of different properties with a scenario of having one property as in common to both the classes.
    * Following above scenario, now write a method inside child class, to access the parent property i.e. the property which is common to both the classes.
    * Can we give value to parent property via child handle????...If yes then how?
11. Write a snippet constituting a super class with three properties and take a subroutine to perform the addition of two properties and assign it to the third one.
12. Write a piece of code with the scenario having multiple child classes and try to override the existing parent method. (Note: Method Overriding means to override a method with same name same argument but different class, overloading is not in practise in sv ).
13. Write a generic class with some properties. (Note: Intention behind the generic class, can change the property/can be configured during instantiation).
14. Write a super class having a method where user will ask the testcase writer to mandatorily indulge the behaviour of the method from the user itself.
15. Write a piece of code with class having some properties of rand type (Note: some of the properties to be randc type and some properties to be of non-rand type). Display these properties using some built in methods (what are those built-in methods).
16. Randomize a class property without using rand or randc. (try std::randomize() and argument to randomize method)
17. Perform the randomisation of a variable in such a way that it always generates odd members with a condition having difference between current value and previous value is equal to 4.
18. Write a class with two variables ‘x’ and ‘y’, constraint the randomisation of these variables as follows:
    * If the value of x is ranging from 5 to 10 then the value of y should be less than 20.
    * If the value of x is ranging in between 20 to 40 then the value of y should be greater than 30 and less than 70 otherwise the value of y should be greater than 70.
19. Write a piece of code to constraint a variable in such a way that the value of the variable must not contain 10,20 and it should not fall in the range of 50 and 65.
20. Write a piece of code to randomize a 2-dimensional array to generate below pattern:

1 2 3 4 5

2 3 4 5 6

3 4 5 6 7

4 5 6 7 8

5 6 7 8 9

1. Write a code to randomise a queue array in such a way that the array contains only even numbers and must bounded in a range of 50 to 100.
2. Assume that implementer has written a class to randomize the variable with restriction that it always generates the value greater than 50, Now as a testcase writer got a task to override the constraint and generate the value less than 50, Write the code for the above scenario.
3. Assume that there is a property in super class which is of rand type, Now constraint the randomisation of the particular property in the range of 100 to 150, from child.

Now take the same property in both parent class and child class and repeat the same scenario as above.

1. Write five time consuming tasks with a condition that all tasks should end at different instances
   * Execute all the above tasks concurrently.
   * Execute any three tasks concurrently and remaining two tasks should execute after the execution of first three tasks (note: these tasks run concurrently).
   * Execute any three tasks concurrently and remaining two tasks should start after completion of any task that consumes least time.
   * Execute any three tasks concurrently and the remaining two tasks also should run concurrently irrespective of the status of those three tasks.
   * Execute all 5 tasks sequentially but task 3 should run in background.
   * Execute task 1,2,3 concurrently, Now the task 4 should start to execute if any one of the tasks above started completed its execution, And task 5 should execute only after the completion of task 1,2, and 3.
2. What is Interface class?
3. What is actual Class?