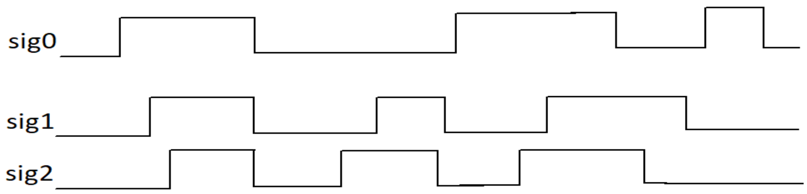
SV Assignments

- ❖ Practice with program block for DFF code.
- ❖ Operator : Practice on Streaming and wildcard operator.
- ❖ Array : Practice on all unpacked array.

1. Declare an array of dynamic type and write values 10, 45, 99 to some random locations (i.e. 25, 34, 56). (Try with Dynamic and queue array)
2. Write a logic to find unique values out of 20 values where some entries are duplicate.
3. Write a logic to find addition of values greater than 30 out of 200 random values.
4. Write a logic to find maximum value out of 200 random values.
5. Write a logic to insert a queue array into another queue array.
6. Declare an associative array and write 10 random values on 10 random locations (non-contiguous). Write a logic to print all entries with their associative index.
   - Try foreach loop
   - Try built-in method of associative array
   - Delete 2-3 entries
   - Read non-existing array index

❖ **Task & Function**

- Practice on all task & function features as directed.
- Write a logic to sample below 3 signals sig0, sig1, sig2 in sig0_mon, sig1_mon, sig2_mon respectively.



(Think what if no of signals are huge, write generic logic)

❖ **SV Interface**

- ➢ Interface and modport example code as given in notes.
- ➢ Clocking block example code (DFF) as given in notes.

❖ **OOPs**

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, take multiple objects (more than 5) and do experiments.
4. Write a piece of code with class having some properties and take values from user through method.
5. Write a class having a method to print “hello world”. Call this method with and without creating object of class.
6. Write a snippet to track the number of times class object has been created.
7. Write a code to access the local property of a class 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 one class having sub object of another class. Perform object assignment & shallow copy and observe the results.

10. Optimize below code:

```
class A;
  int a, b;

  function void xyz();
    :
  endfunction
endlass

class B;
  int a;
  byte c;

  function void xyz();
    :
  endfunction

  task write(…);
     :
  endtask

endlass

class C;
  int a;

  function void xyz();
    :
  endfunction

  function read(…);
     :
  endfunction
endlass
```

11. 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.

12. 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?

13. Analyse below code, write your expectation. Run the code and observe the actuals.

```
class parent;
  function void xyz();
     $display(“Parent Method”);
  endfunction
endlass

class child extends parent;
  function void xyz();
    $display(“Child Method”);
  endfunction
endclass


parent p_h;
child c_h;
```

```
initial begin
  p_h = new();
  c_h = new();
  p_h.xyz();
  c_h.xyz();
  p_h = c_h;
  p_h.xyz();
end
```

14. 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)(polymorphism).
15. Write a code to add some extra behaviour/line_of_code in the exiting parent method through child class.
16. Write a class with one variable having default size 8. Take four objects of this class with below conditions:
    1. **Two objects with size remain 8**
    2. **Third object with size 10**
    3. **Fourth object with size 16**
17. Write a generic class with some properties. (Note: Intention behind the generic class, can change the property/can be configured during instantiation).
18. Write a super class having a method where user will ask the testcase writer to mandatorily include/write the behaviour of the method from the user itself.
19. Write a class to have only single object of it (singleton class).

- **Randomization**
  - 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).
  - Randomize a class property without using rand or randc. (try std::randomize() and argument to randomize method)
  - 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.

20. 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.
21. 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.
22. Write a piece of code to randomize a 2-diamensional 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
```

- 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.

- 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.
- 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.

- **Threads**
    - Write a checker logic for below condition:
        - If value of one bit variable ‘a’ is change to one. Then within 15ns value of one-bit variable ‘b’ should be one.
    - Do as directed.

```
class exp;
  task write();
     $display(“Entering into the method”);
     #10;
     $display(“Exiting from the method”);
  endtask
endlass
```

  - Take four object of above class (array of abject) and call ‘write’ method for all parallelly, but make sure this should block the execution of further code.
- 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.

- Do discussed experiments in shared semaphore, event code.

- **Verification of a Design**
    - Verification Environment Developed from scratch.
    - Multiple Testcase Handling
    - Function Coverage
    - Procedure/testcase and checkers implementation
    - Regression and coverage report analysis

- What is Interface class?
- What is actual Class?