DAY-71
#100DAYSRTL

“System Verilog:-Inheritance”

“**Introduction**”:-

In System Verilog, inheritance serves as a pivotal feature enabling the creation of hierarchical relationships among classes. This concept allows derived classes to inherit properties and behaviours from base classes, promoting code reusability, modularity, and efficient design in hardware description and verification methodologies.

“**Inheritance**”:-

- This is the property of OOP by virtue of which a class can inherit properties and behaviour of another class called parent or base class.
- The child or derived class can add more properties or behaviour to the base class.

“**Child Class or Sub Class**”:-

- Subclasses, like children, inherit traits from parent classes.
- In verification, we often start with a basic reference class and gradually add more features to build a stronger base class.
- It's similar to how kids inherit characteristics from their parents and grow with added attributes over time

```
class <class name> extends <base_class_name>;
endclass
```

“Super Keyword”:-

- Super keyword is kind of like **this keyword** except it is used to access property or method present in the parent class.

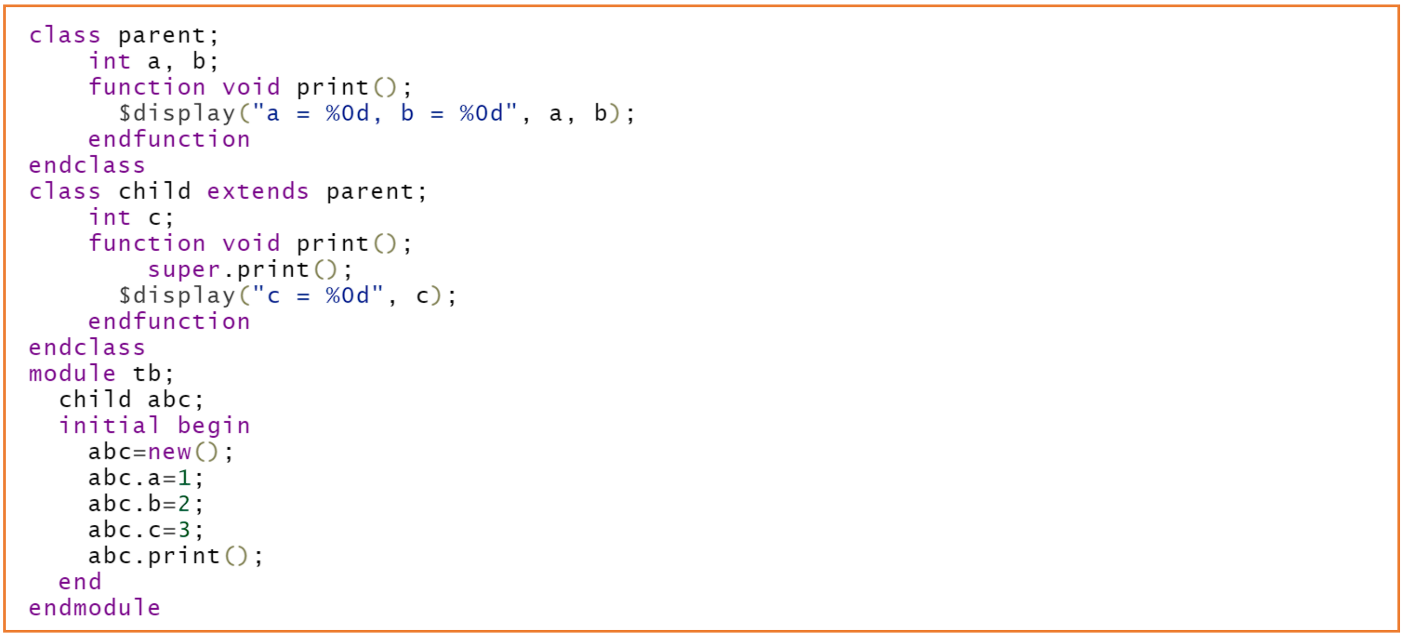
```
class abc extends xyz;
 function test();
  super.test1();
 endfunction
endclass
```

- **Note:-**It is not necessary to use the super keyword when the method name is unique b/w parent and child class. Super keyword is mainly used when method names are common in parent and child or when we do function overriding which we will learn later.
- When a constructor of the child class is called it automatically calls the new class of the parent class, i.e., **super.new()** is by default the first thing called inside the constructor of child class

“Single inheritance”:-

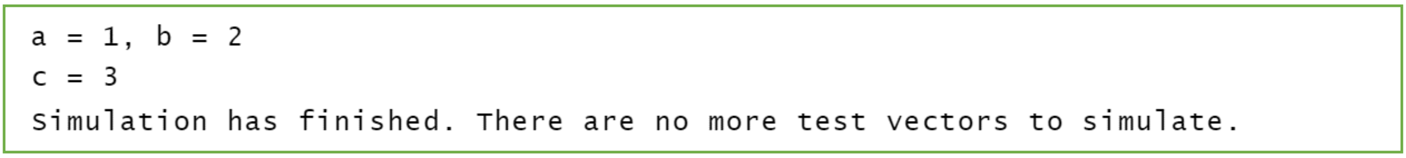
- In Single inheritance there was a child class and parent class.

“Code Practising”:-



```
class parent;
    int a, b;
    function void print();
      $display("a = %0d, b = %0d", a, b);
    endfunction
endclass
class child extends parent;
    int c;
    function void print();
        super.print();
      $display("c = %0d", c);
    endfunction
endclass
module tb;
  child abc;
  initial begin
    abc=new();
    abc.a=1;
    abc.b=2;
    abc.c=3;
    abc.print();
  end
endmodule
```

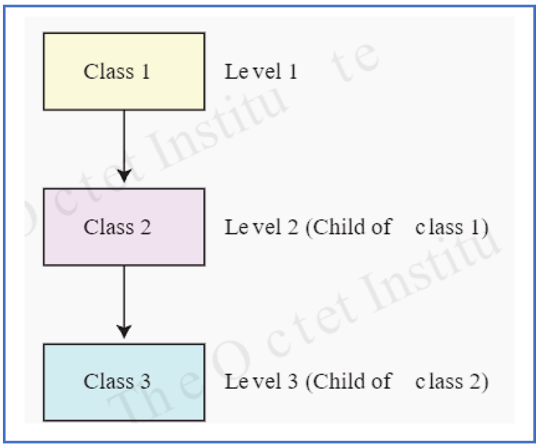
“Result”:-



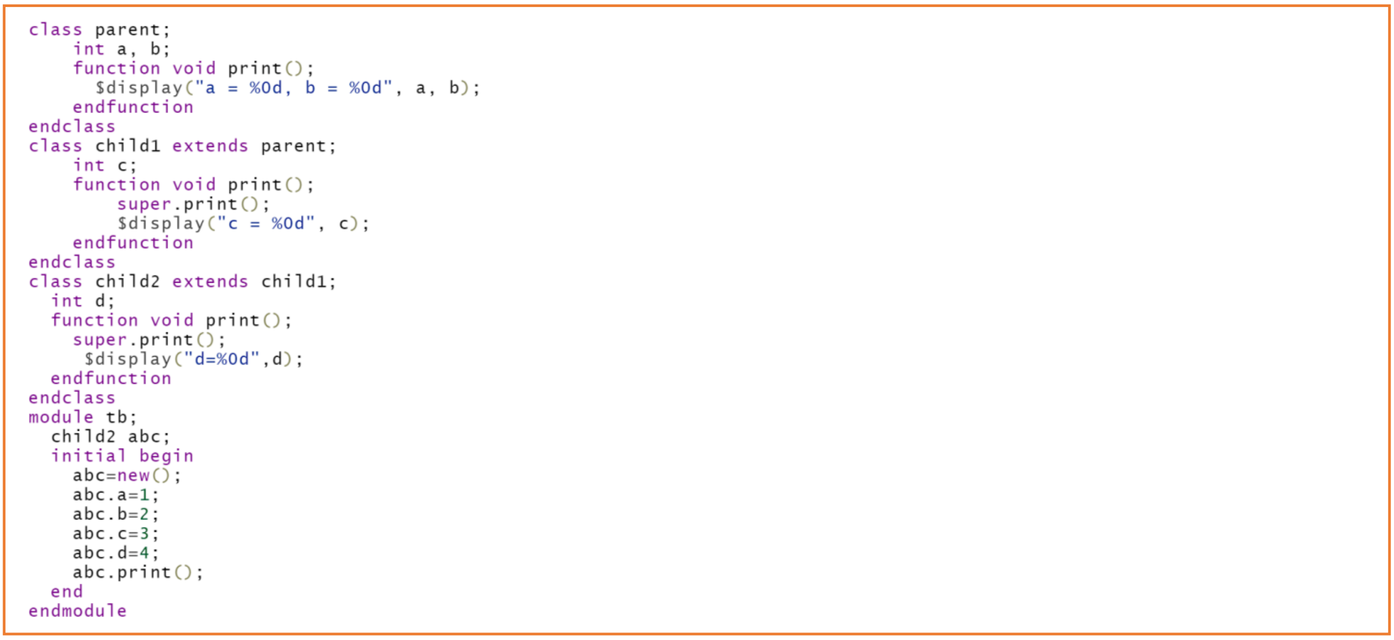
```
a = 1, b = 2
c = 3
Simulation has finished. There are no more test vectors to simulate.
```

“Multi-level Inheritance”:-

- In multi-level inheritance child class can also be the parent of another child
- In multi-level inheritance it is not allowed to reach a class higher than the immediate parent class. This means that **super.super.new()** or similar is not allowed.
- In multi-level inheritance it is not allowed to reach a class higher than the immediate parent class. This means that **super.super.new()** or similar are not allowed.

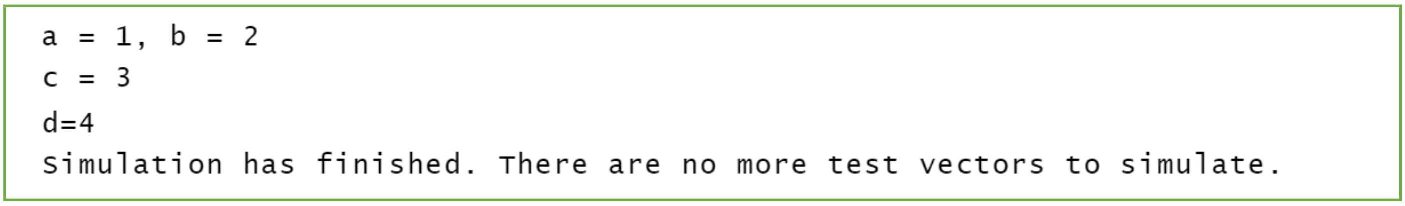


“Code practicing”:-



```
class parent;
    int a, b;
    function void print();
      $display("a = %0d, b = %0d", a, b);
    endfunction
endclass
class child1 extends parent;
    int c;
    function void print();
        super.print();
        $display("c = %0d", c);
    endfunction
endclass
class child2 extends child1;
  int d;
  function void print();
    super.print();
     $display("d=%0d",d);
  endfunction
endclass
module tb;
  child2 abc;
  initial begin
    abc=new();
    abc.a=1;
    abc.b=2;
    abc.c=3;
    abc.d=4;
    abc.print();
  end
endmodule
```

“Result”:-



```
a = 1, b = 2
c = 3
d=4
Simulation has finished. There are no more test vectors to simulate.
```