



# July Batches

1: What will be the output of the below code

---

```
class parent;
  int i;
  string name;
  function new();
    this.i=6;
    this.name="sumedha";
  endfunction

  virtual function print();
    $display("the parent values of i:%0d and string=%s",i,name);
  endfunction
endclass

class child extends parent;
|

  function new(int i, string name);
    super.i=i;
    super.name=name;
  endfunction

  function print();
    super.print();
    $display("the child values of i :%0d and string=%s",i,name);
  endfunction
endclass

module tb;
  parent p;
  child ch;
  initial begin
    p=new();
    ch=new(5,"sumedhaIT");
    ch.print();
    $display("*****");
    p.print();
    p=ch;
    $display("*****");
    p.print();
    $display("*****");
  end
endmodule
```

2: CHANGE THE BELOW GIVEN COVERAGE TO 62.5%, THE GIVEN COVERAGE IS 25%.

```
program main;
    bit[0:3]y;
    bit [0:3]values[$]='{1,3};
covergroup cg;
cover_point_y : coverpoint y
    {option.auto_bin_max=8;}//do not change the auto bin max
endgroup
cg cg_inst = new();
initial begin
foreach(values[i])
begin
y=values[i];
cg_inst.sample();
end
$display("the coverage is : %.1f", $get_coverage());
end
endprogram
```

3. Explain the concept of fork-join, fork-join none, fork join any in SystemVerilog and provide an example code.

a) make fork join to behave as fork join any.

4. Declare a parent class with 3 properties inside parent, now make two derived class from it, both should be extended from the parent. Don't declare the parent properties inside derived classes.

Perform any arithmetic operation on this properties inside one of the derived class only, such that the change should be updated in both parent and other derived class.

