



VERIFICATION ACADEMY

Basic OVM

OVM "Hello World"

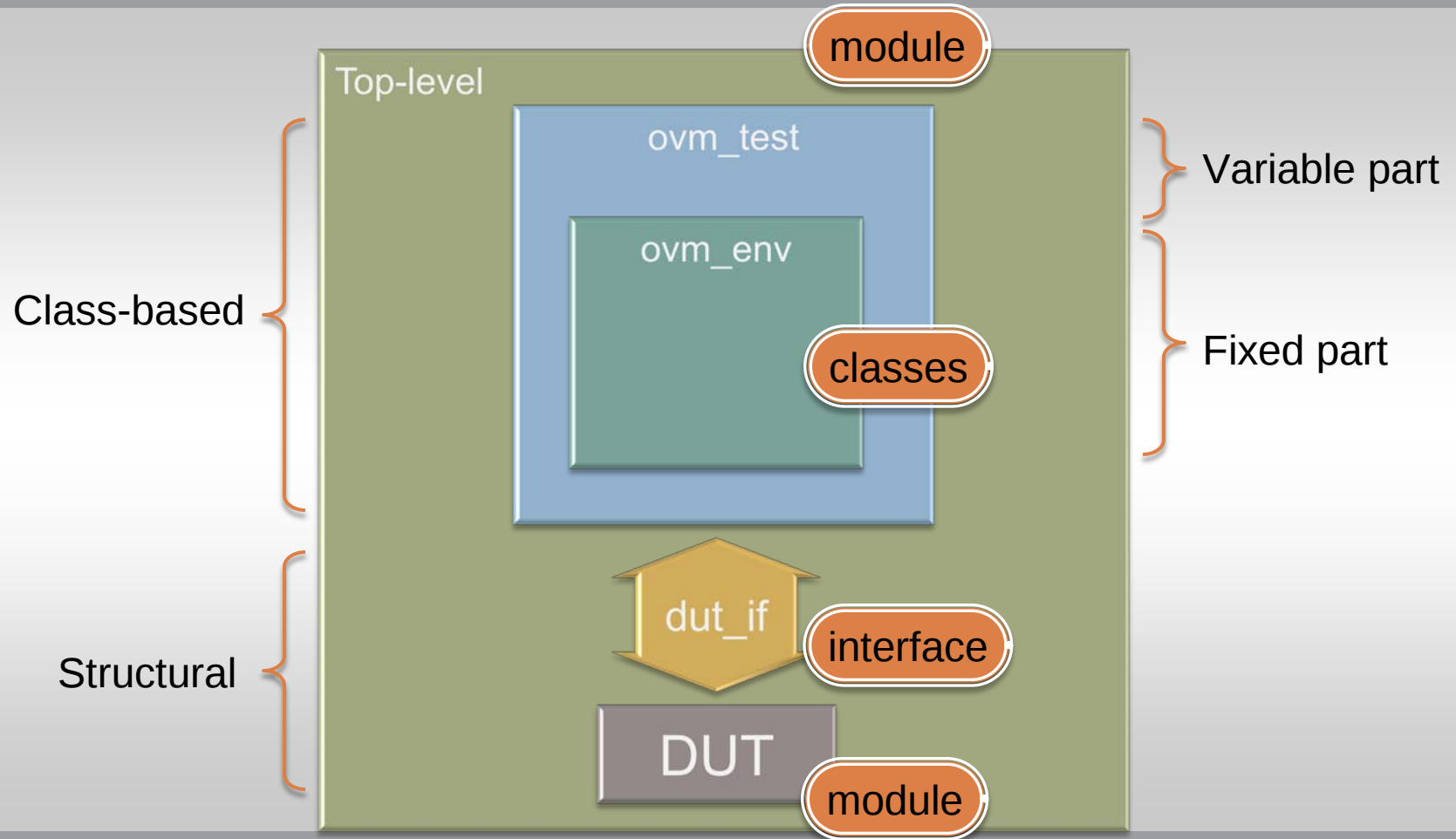
John Aynsley
CTO, Doulos

academy@mentor.com
www.verifacationacademy.com





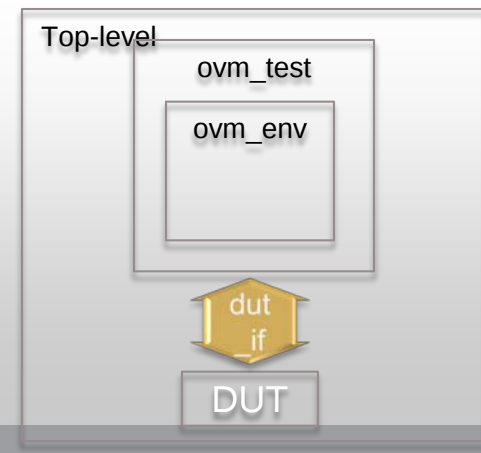
DUT and Verification Environment





Interface

```
interface dut_if();  
  
    ...  
  
endinterface: dut_if
```



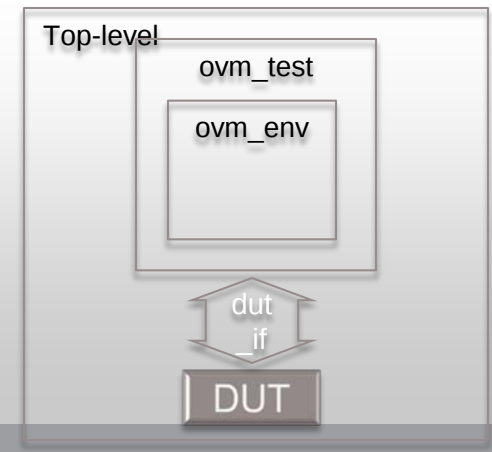


DUT 

```
module dut(dut_if _if);
```

```
...
```

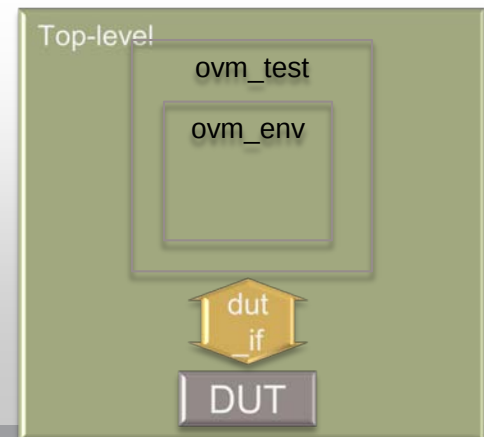
```
endmodule: dut
```





DUT Instantiation

```
module top;  
    ...  
  
    dut_if dut_if1 ();  
  
    dut    dut1 ( ._if(dut_if1) );  
  
    ...  
endmodule: top
```

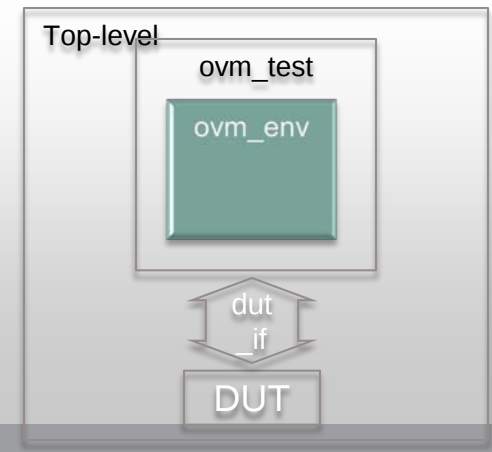




Env 

```
class my_env extends ovm_env;
```

```
endclass: my_env
```

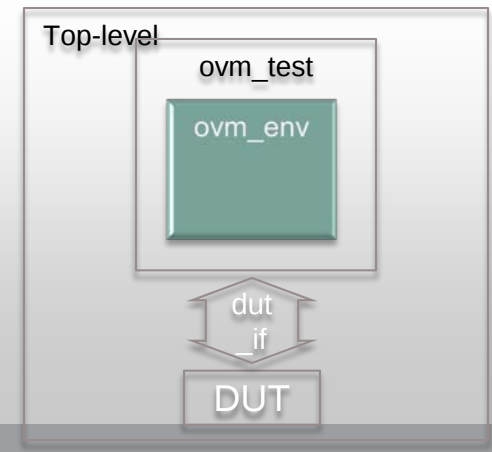




Env

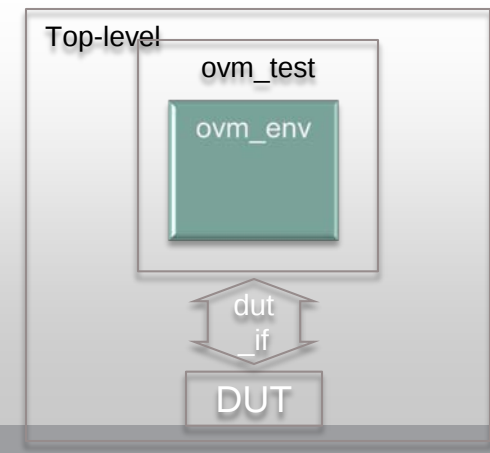
```
class my_env extends ovm_env;  
  `ovm_component_utils(my_env)
```

```
endclass: my_env
```



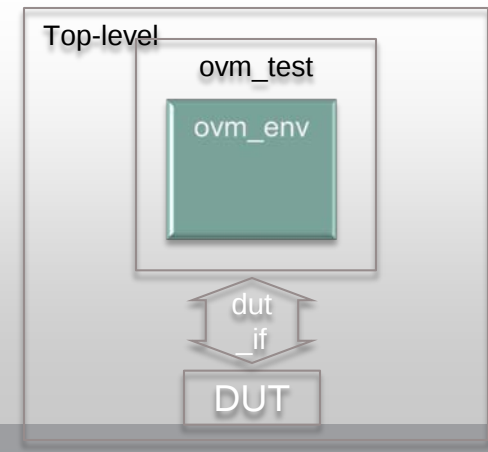


```
class my_env extends ovm_env;  
  `ovm_component_utils(my_env)  
  
  function new(string name, ovm_component parent);  
    super.new(name, parent);  
  endfunction: new  
  
endclass: my_env
```



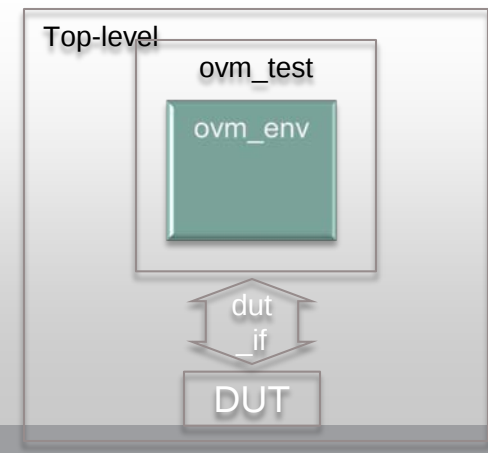


```
class my_env extends ovm_env;  
  `ovm_component_utils(my_env)  
  
  function new(string name, ovm_component parent);  
    super.new(name, parent);  
  endfunction: new  
  
  function void build;  
    super.build();  
  endfunction: build  
  
endclass: my_env
```





```
class my_env extends ovm_env;  
  `ovm_component_utils(my_env)  
  
  function new(string name, ovm_component parent);  
    super.new(name, parent);  
  endfunction: new  
  
  function void build;  
    super.build();  
  endfunction: build  
  
  task run;  
  
  endtask: run  
endclass: my_env
```



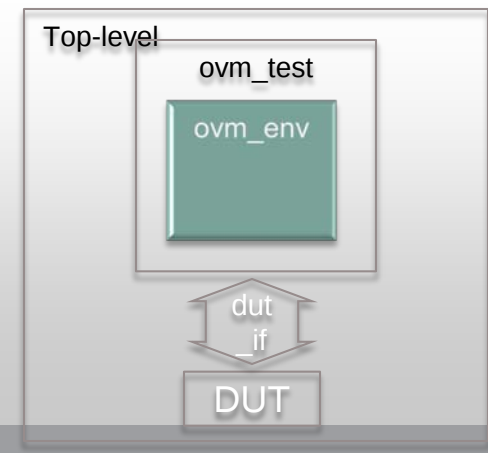


```
class my_env extends ovm_env;
  `ovm_component_utils(my_env)

  function new(string name, ovm_component parent);
    super.new(name, parent);
  endfunction: new

  function void build;
    super.build();
  endfunction: build

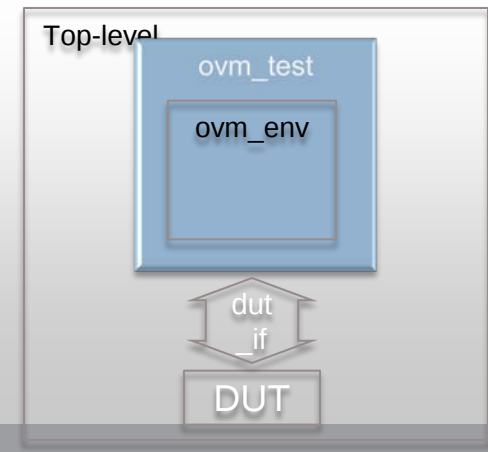
  task run;
    #10 ovm_top.stop_request();
  endtask: run
endclass: my_env
```





```
class my_test extends ovm_test;  
  `ovm_component_utils(my_test)
```

```
endclass: my_test
```



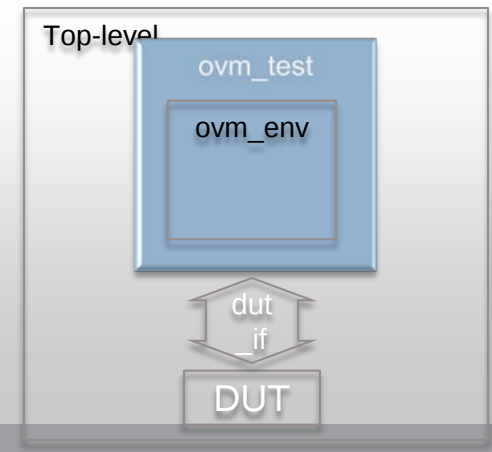


```
class my_test extends ovm_test;  
  `ovm_component_utils(my_test)
```

```
my_env my_env_h;
```

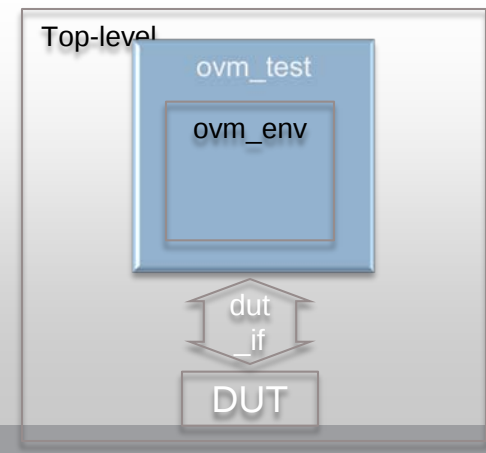
_h = handle

```
endclass: my_test
```





```
class my_test extends ovm_test;  
  `ovm_component_utils(my_test)  
  
  my_env my_env_h;  
  
  function new(string name, ovm_component parent);  
    super.new(name, parent);  
  endfunction: new  
  
  function void build;  
    super.build();  
  
  endfunction: build  
  
endclass: my_test
```





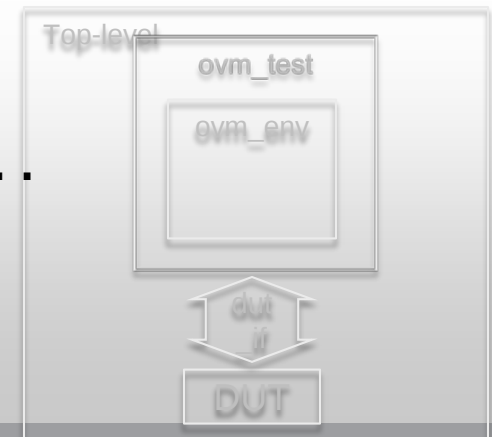
```
class my_test extends ovm_test;
  `ovm_component_utils(my_test)

  my_env my_env_h;

  function new(string name, ovm_component parent);
    super.new(name, parent);
  endfunction: new

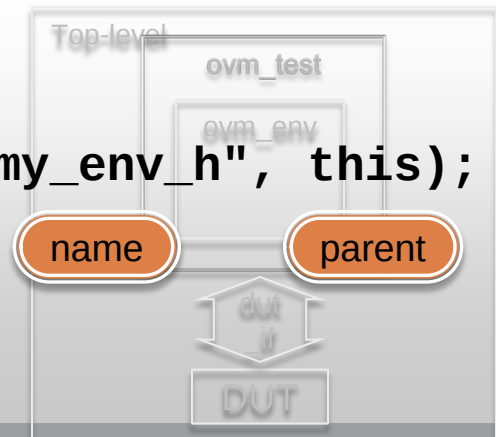
  function void build;
    super.build();
    my_env_h = my_env::type_id::create(...
  endfunction: build

endclass: my_test
```





```
class my_test extends ovm_test;  
  `ovm_component_utils(my_test)  
  
  my_env my_env_h;  
  
  function new(string name, ovm_component parent);  
    super.new(name, parent);  
  endfunction: new  
  
  function void build;  
    super.build();  
    my_env_h = my_env::type_id::create("my_env_h", this);  
  endfunction: build  
  
endclass: my_test
```





```
`include "ovm_macros.svh"

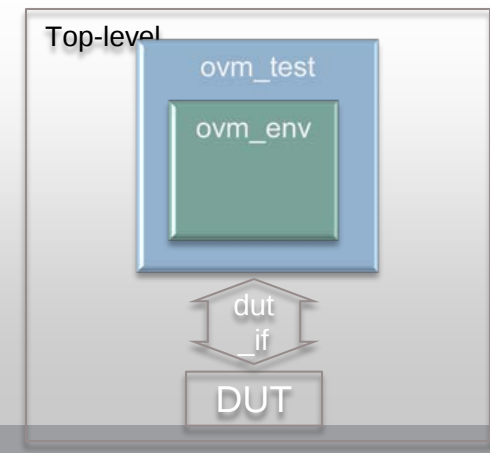
package my_pkg;

    import ovm_pkg::*;

    class my_env extends ovm_env;
        ...
    endclass: my_env

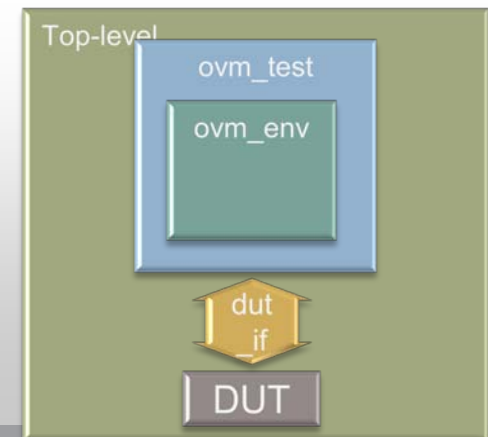
    class my_test extends ovm_test;
        ...
    endclass: my_test

endpackage: my_pkg
```



Test Instantiation

```
module top;  
  
    import ovm_pkg::*;  
    import my_pkg::*;  
  
    dut_if dut_if1 ();  
  
    dut    dut1 ( ._if(dut_if1) );  
  
endmodule: top
```





Test Instantiation

```
module top;

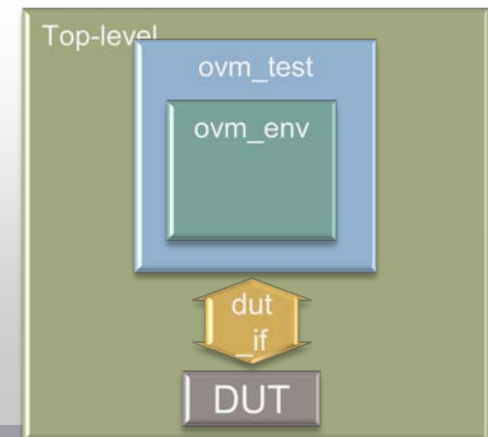
    import ovm_pkg::*;
    import my_pkg::*;

    dut_if dut_if1 ();

    dut    dut1 ( ._if(dut_if1) );

    initial
    begin
        run_test("my_test");
    end

endmodule: top
```





Running the Simulation

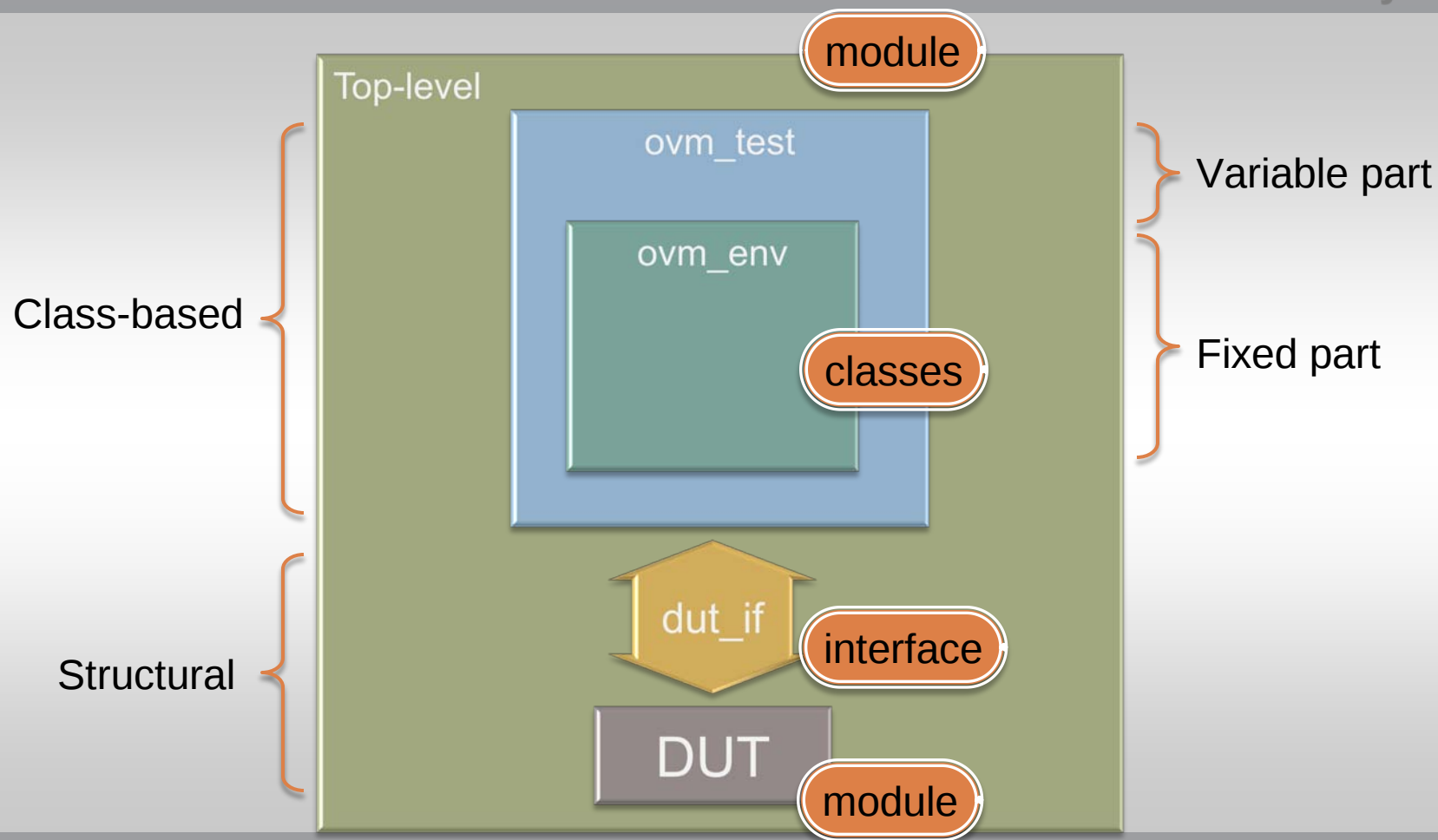
```
qverilog file.sv
# Loading sv_std.std
# Loading mtiOvm.ovm_pkg(fast)
# Loading work.my_pkg(fast)
# Loading work.top(fast)
# Loading work.dut_if(fast)
# run -all
# -----
# OVM-2.1
# (C) 2007-2009 Mentor Graphics Corporation
# (C) 2007-2009 Cadence Design Systems, Inc.
# -----
# OVM_INFO @ 0: reporter [RNTST] Running test my_test...
#
# --- OVM Report Summary ---
```



Running the Simulation

```
#
# --- OVM Report Summary ---
#
# ** Report counts by severity
# OVM_INFO : 1
# OVM_WARNING : 0
# OVM_ERROR : 0
# OVM_FATAL : 0
# ** Report counts by id
# [RNTST] 1
# ** Note: $finish : C:/roam/ovm/ovm-2.1/src/base/ovm_root.svh(507)
# Time: 10 ns Iteration: 8 Instance: /ovm_pkg::ovm_root::run_test
```

Summary





VERIFICATION ACADEMY

Basic OVM

OVM "Hello World"

John Aynsley
CTO, Doulos

academy@mentor.com
www.verifacationacademy.com

