04 · SystemVerilog OOP & Classes

Class skeleton

```
class packet;
  rand bit [7:0] addr;  byte payload[$];
  static int count;
  function new(bit [7:0] a = 0); addr = a;
count++; endfunction
  virtual function string
convert2string();
    return $sformatf("addr=%02h", addr);
  endfunction
endclass
```

Handles and objects

Code	Effect
`packet p;`	handle, `null`
`p = new();`	new object
`q = p;`	same object, two handles
`q = new p;`	shallow copy (new object)
`p = null;`	drop reference; GC frees
`p == q`	same object?

Calling a method through `null` → fatal error.

Copying

Field kind	`new p` copies
bits, ints, strings	value
queues, dynamic arrays	value
class handles	**handle only (shared)**

Deep copy = copy values **and** `clone()` every nested object.

Inheritance

```
class bad_pkt extends packet;
  function new(); super.new(); endfunction
  virtual function string
convert2string();
    return {"BAD ",
super.convert2string()};
  endfunction
endclass
```

virtual or not

Base method	Called via base handle on child object
`virtual`	child's override
non-virtual	base version

Once virtual, always virtual down the tree.

Casting

`base_h = child_h;` always legal (upcast). `if (!$cast(child_h, base_h)) $error(…);` downcast, checked at run time.

Abstract

`virtual class base; pure virtual function bit check(packet p); endclass` — cannot `new` it; children implement every pure method.

Other keywords

Keyword	Meaning
`this`	current object
`super`	parent's members
`static`	shared by all objects
`local`	visible in this class only
`protected`	this class and children
`const`	set once (in `new`)
`extern`	method body outside class
`typedef class c;`	forward declaration

Parameterised class

```
class sb #(type T = packet);
  T q[$];
endclass
sb #(bad_pkt) s = new();
```

Testbench objects map to UVM

Pre-UVM	UVM
transaction class	`uvm_sequence_item`
`copy` / `compare` / `convert2string`	`do_copy` / `do_compare` / `convert2string`
generator + blueprint	sequence + factory override
mailbox	TLM FIFO / port

Remember it like this

Idea	Picture
Class / object	blueprint / house
Handle	a TV remote; `null` = remote with no TV
`p2 = p1`	a second remote, same TV
Shallow copy	photocopied folder that still holds a shortcut to the same photo
Deep copy (`clone`)	copy the folder and the photo
Inheritance	a smartphone is still a phone
`virtual`	a universal remote's play button
`$cast`	check the parcel label before signing
`static`	the shared ticket dispenser
Blueprint generator	a cookie cutter: same cutter, fresh cookie