Thank You

- For more videos, please visit us at -

https://aspiringdirections.wordpress.com

Abstract class & pure virtual method

virtual class;

endclass

-- can't take its instance ; only can extend

in virtual class – can't define method ; so can declare pure virtual method without body

virtual class;

  pure virtual function void get_properties; // no – implementation

endclass

Upcasting (Polymorphism)

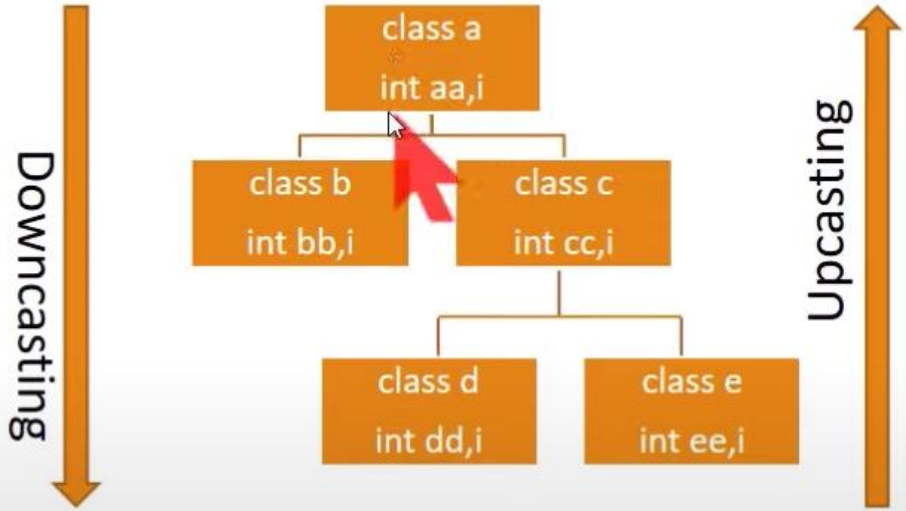
Same code – works differently – for different types.

Static – parameterized types … *Dynamic – virtual method*

virtual function

code reuse ---------- HOW ?

Overridden members – Upcasting & Downcasting



Upcasting – casting e into a – (a = e) **LEGAL**

```
e e1; a a1;
e1 = new;
a1 = e1;
```

Downcasting – a into e – (e = a) **ILLEGAL**

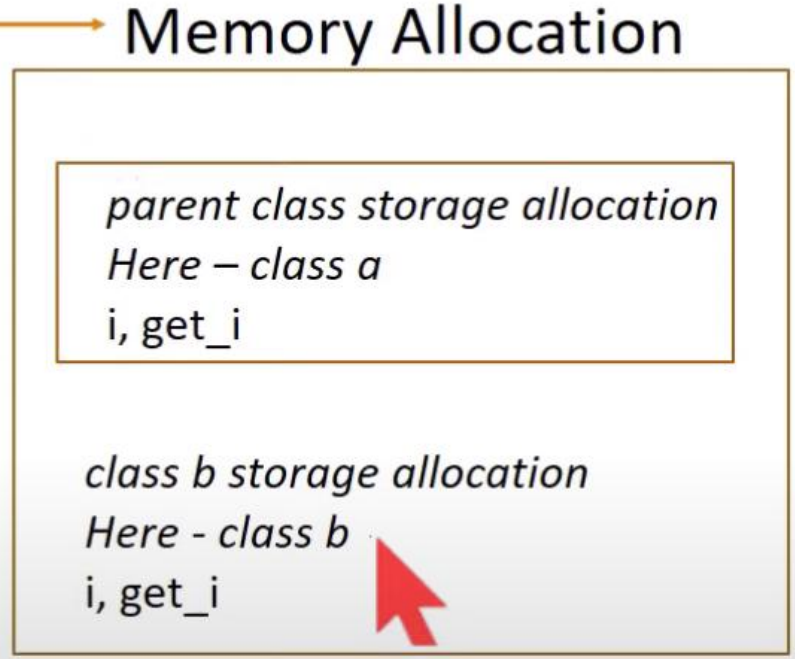
```
a a1; e e1;
a1 = new;
$cast(e1, a1);
```

Inheritance memory allocation

```
class a;
   int i;
   function get_i; endfunction;
endclass : a

class b extends a;
   int i;
   function get_i; endfunction;
endclass : b

b = new();
```



Chain constructor

super.new()

```
class a;
endclass : a


class b extends a;
function new();
   super.new();
endfunction
endclass : b
```

1 level of inheritance

super.super.new 

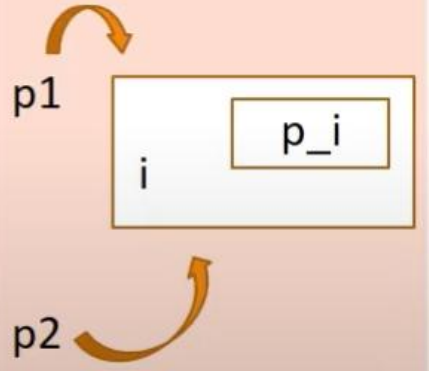
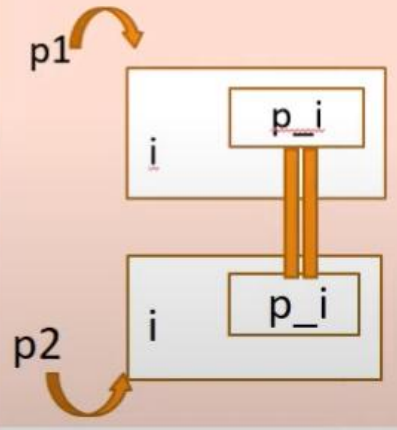
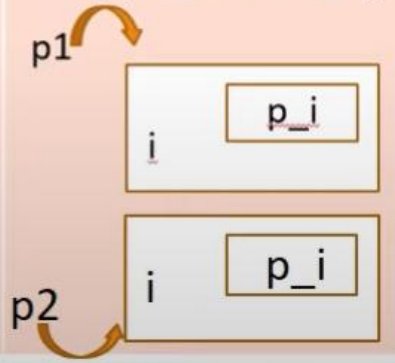
this, super & local(data hiding)

this	super

to access current class properties
 	
class a;
  int i;
  int delay=5;
  local int j;
 	
  function new(int i);
  this.i = i;
  endfunction
 	
endclass
	to access parent class properties
	
	class b **extends** a;
	  int result;
	  int delay=10;
	
	  function print;
	  result = **this**.delay + **super**.delay;
	  endfunction
	
	endclasss

Object assignment, shallow copy & deep copy

How many times – ***new*** is called?

Packet Class	Assignment	Shallow Copy	Deep Copy
class packet; **int** i; other_p p_i; **endclass** : packet **module** a(); packet p1; p1 = **new**; **endmodule** : a	packet p2; p2 = p1; 	packet p2; p2 = **new** p1; 	packet p2; p2 = **new**; p2.copy(p1); - copy is another f^n 

Agenda

- ✓ Object assignment, shallow copy & deep copy
- ✓ Inheritance
- ✓ this, super & local(data hiding)
- ✓ Chain constructor
- ✓ Inheritance memory allocation
- ✓ Overridden members – Upcasting & Downcasting
- ✓ Upcasting (Polymorphism)
- ✓ Abstract class & pure virtual method

SystemVerilog for Verification

CLASS & OOPS – PART II