OOPS-ASSIGNMENT-2

1. In what scenarios would you prefer using shallow copy over deep copy in system Verilog? Discuss the advantages and disadvantages of using shallow copy?

Ans:

In the shallow copy the memory creation is done by creating a copy of a object to handle where a copy of the object is created in the handle by this the handle is converted as object. The properties updated through the copied class will not effect but changing through created object will change all the copied handles too. The memory conception is loo compare to deep copy. Temporary access for the object.

Advantage:

memory efficient - only a new reference is created for object.

Faster execution – no need to clone inner object.

Consistent state – changes through any handle will effect all the other references.

Disadvantages :

shared data risk – A change in one reference will affect all other reference.

No data isolation – not suitable when independent behavior is needed.

Hard to find bug – unintended change can lead to race condition and inconsistencies.

1. What happens to the original object if a nested object’s property is modified after performing a shallow copy? Explain your answer with an example?

Ans:

if the nested object is modified using handle then it will effect the original objects value too.

Example:

class a;

int team;

function new();

this.team = 5;

endfunction

endclass

class b;

a a1;

int mem;

function new();

this.mem = 11;

a1 = new();

endfunction

function void display();

$display("mem => %0d team => %0d",mem,a1.team);

endfunction

endclass

module top;

initial begin

b b1,b2;

b1 = new();

b2 = new b1;

b1.display();

b2.display();

b2.mem = 12;

b1.display();

b2.display();

b1.mem = 15;

b1.display();

b2.display();

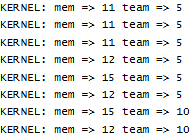
b2.a1.team = 10;

b1.display();

b2.display();

end

endmodule



1. What is the difference between extend and super in System Verilog?

Ans:

extend key word is used to extend a class property and member to another class.

Super key word is used to access the property and method of a extended class.

class a;

int team;

function new();

this.team = 5;

endfunction

endclass

class b extends a;

int mem;

function new();

this.mem = 11;

endfunction

function void display();

$display("mem => %0d team => %0d",mem,super.team);

endfunction

endclass

module top;

initial begin

b b1;

b1 = new();

b1.display();

end

endmodule

1. What Is the purpose of super in System Verilog inheritance, and how is it used in a derived class?

Ans:

The super key word is used to access the parent class property and method where the both class has same name. This is used to specify the usage of variable to clarify the compiler.

1. Define a base class BankAccount with a protected member balance and methods to deposit() and withdraw() money. Create a drived class SavingsAccount that modifies balance with interest after a withdrawal. Add a display\_balance() method in SavingsAccount that uses the protected balance variable.

Ans:

class BankAccount;

protected shortreal balance;

function new();

this.balance = 10000;

endfunction

function void deposit(int amount);

this.balance += amount;

endfunction

function void withdraw(int amount);

if(amount<=this.balance)

this.balance -= amount;

else $display("insuficient amount");

endfunction

endclass

class SavingsAccount extends BankAccount;

function void withdraw(int amount);

super.withdraw(amount);

super.deposit((super.balance)\*(0.05));

endfunction

function void display();

$display("Balance => ",super.balance);

endfunction

endclass

module top;

initial begin

SavingsAccount s;

s = new();

s.withdraw(5000);

s.display();

end

endmodule

1. Create a class B extended from class A with 2 different data type variable.

Ans:

class a;

int date;

string day;

function new();

this.date = 12;

this.day = "mon";

endfunction

endclass

class b extends a;

function void display();

$display("Date => %0d Day => %s",super.date,super.day);

endfunction

endclass

module tb;

initial begin

b b1;

b1 = new();

b1.display();

end

endmodule

KERNEL: Date => 12 Day => mon

1. Create a class A with 4 different data types variable.

Ans:

class a;

int quantity;

string ProductName;

shortreal discount;

bit[12:0] price;

function new();

this.quantity = 12;

this.ProductName = "apple";

this.discount = 0.12;

this.price = $rtoi ((int'(13'd300)\*quantity)-((int'(13'd300)\*quantity)\*discount));

display();

endfunction

function display();

$display("quantity => %0d ProductName => %s discount => %f price => %0d",quantity,ProductName,discount,price);

endfunction

endclass

module tb;

initial begin

a a1;

a1 = new();

end

endmodule

1. Create a class G extended from class F. Add a class constructor to assign the value for the class F Variables.

Ans:

class F;

int quantity;

string ProductName;

shortreal discount;

bit[12:0] price;

endclass

class G extends F;

function new();

super.quantity = 12;

super.ProductName = "apple";

super.discount = 0.12;

super.price = $rtoi ((int'(13'd300)\*quantity)-((int'(13'd300)\*quantity)\*discount));

display();

endfunction

function display();

$display("quantity => %0d ProductName => %s discount => %f price => %0d",quantity,ProductName,discount,price);

endfunction

endclass

module tb;

initial begin

G g1;

g1 = new();

end

endmodule

1. Create a class H with a static variable and a static function and increment the value of the static variable.

Ans:

class F;

static int a = 1;

static function void display();

a = (++a) + (a++);

$display("A => %0d",a);

endfunction

endclass

module tb;

initial begin

F f1;

f1 = new();

f1.display();

end

endmodule

1. Invoke static function for ‘N’ number of times and finally display the value of static variable.

Ans:

class F;

static int a = 1;

static function void display();

a++;

$display("A => %0d",a);

endfunction

endclass

module tb;

initial begin

F f1;

f1 = new();

for(int n=0; n<=3; n++) begin

f1.display();

end

end

endmodule

1. Create an object for the class and invoke display function. Convert display as pure virtual method and invoke the method.

Ans:

class G;

virtual function void display();

endfunction

endclass

class F extends G;

static int a = 1;

function void display();

a++;

$display("A => %0d",a);

endfunction

endclass

module tb;

initial begin

F f1 = new();

G g1 = f1;

for(int n=0; n<=3; n++) begin

g1.display();

end

end

endmodule