System Verilog   
Object Oriented Programming Language (OOPS)

Submitted To: Dr Rita Mahajan   
Submitted By: Harshit Kaundal

**OPPs**

OOP stands for object-oriented programming. This type of programming gets its inspiration from real word objects where every object has its own property, characteristics defined by a blueprint. As this   
programming model is inspired by reads-life objects writing code becomes more easier as we can relate it to real objects.

**Some key features of OOP are:**   
Inheritance   
Polymorphism   
Encapsulation   
Data Abstraction

**Inheritance**   
This is the property of OOP by virtue of which a class can inherit properties and behaviour of another class called parent or base class. The child or derived class can add more property or behaviour on the base class.

|  |  |
| --- | --- |
|  |  |

**Encapsulation**

This is a property by which we can bundle all the data and methods into one unit. This also helps in black boxing a unit where users can focus more on using the unit without knowing the underlying process or complexity.

|  |  |
| --- | --- |
|  |  |

**Polymorphism**   
Polymorphism is a concept in which same method can act differently in child class or when inputs are different. OOPs provide us two ways to enable polymorphism – function overloading and function overriding

|  |  |
| --- | --- |
|  |  |

**Data Abstraction**   
Data Abstraction means hiding unnecessary data and representing only what is necessary for the user basically that particular use case.

|  |  |
| --- | --- |
|  |  |

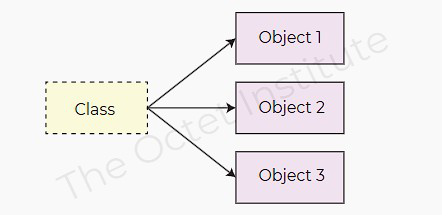
**OOPs features in System Verilog**

System Verilog is an HDL, i.e., hardware define language and thus all features of OOP are not needed in SV. Some of the complex OOP feature are omitted in the latest version of OOP. Some of the supported features are:  
●Single & multi level inheritance  
●Function overriding  
●Virtual classes and pure virtual functions  
●Virtual functions

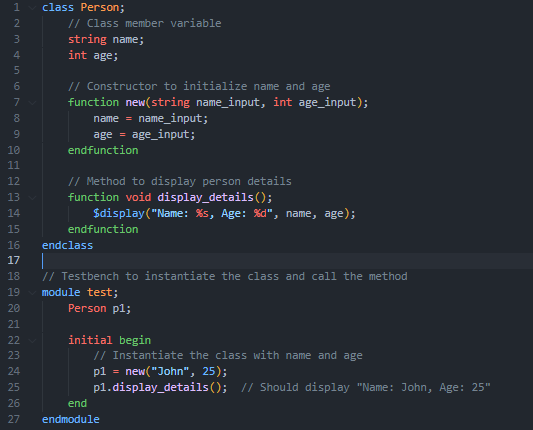
**Class and Objects**

**Class** is a blueprint which defines the properties and behaviour of an object. In OOP classes are the entity which encapsulates all the data and methods.

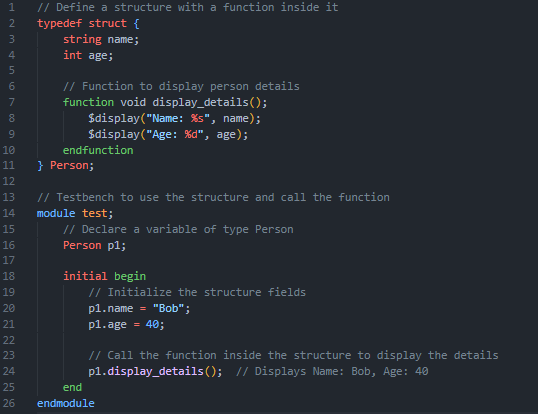
**Objects** are the unique entity created from class. Objects are dynamic in nature, i.e, it is created dynamically during runtime.



**Class :**



**Structure :**



**References:**

●<https://vlsiverify.com/systemverilog/>  
●[https://www.chipverify.com/tutorials/](https://vlsiverify.com/systemverilog/)[systemverilog](https://www.chipverify.com/tutorials/systemverilog)  
●www.chatgpt.com

**Thank You**