

APEX - OBJECTS

An instance of class is called Object. In terms of Salesforce, object could be of class or you could create an object of sObject as well.

Object creation from Class

You could create an object of class as you might have done in Java or other object oriented programming language.

Below is an example Class called MyClass:

```
//Sample Class Example
public class MyClass {
    Integer myInteger = 10;
    public void myMethod (Integer multiplier) {
        Integer multiplicationResult;
        multiplicationResult=multiplier*myInteger;
        System.debug('Multiplication is '+multiplicationResult);
    }
}
```

This is an instance class, that is to call or access the variables or methods of this class, you must create an instance of this class and then you could do all the operations.

```
//Object Creation
//Creating an object of class
MyClass objClass = new MyClass();

//Calling Class method using Class instance
objClass.myMethod(100);
```

sObject creation

As you know, sObjects are the objects of Salesforce in which you store the data. For example, Account, Contact, etc. are custom objects. You could create object instances of these sObjects.

Below is the example of sObject initialization and how you could access the field of that particular object using dot notation and assign the values to fields.

```
//Execute the below code in Developer console by simply pasting it
//Standard Object Initialization for Account sObject
Account objAccount = new Account(); //Object initialization
objAccount.Name = 'Testr Account'; //Assigning the value to field Name of Account
objAccount.Description = 'Test Account';
insert objAccount; //Creating record using DML
System.debug('Records Has been created '+objAccount);

//Custom sObject initialization and assignment of values to field
APEX_Customer_c objCustomer = new APEX_Customer_c ();
objCustomer.Name = 'ABC Customer';
objCustomer.APEX_Customer_Description_c = 'Test Description';
insert objCustomer;
System.debug('Records Has been created '+objCustomer);
```

Static Initialization

Static methods and variables are initialized only once when a class is loaded. Static variables aren't transmitted as part of the view state for a Visualforce page.

Below is the example of Static method as well as Static variable.

```
//Sample Class Example with Static Method
public class MyStaticClass {
    Static Integer myInteger = 10;
    public static void myMethod (Integer multiplier) {
        Integer multiplicationResult;
        multiplicationResult=multiplier*myInteger;
        System.debug('Multiplication is '+multiplicationResult);
    }
}

//Calling the Class Method using Class Name and not using the instance object
MyStaticClass.myMethod(100);
```

Static Variable Use

Static variables will be instantiated only once when class is loaded and this phenomenon can be used to avoid the trigger recursion. Static variable value will be same within same execution context and any class, trigger or code which is executing could refer it and prevent the recursion.