How to avoid recursive trigger in Salesforce

What is recursion?

Recursion in simple terms mean the same piece of code runs again and again which can results in unexpected output or error. This happens when we do not handle the recursive behavior of the trigger. The recursion arises when our trigger runs multiple times due to updates.

So to avoid recursive behavior of trigger we make use of a static variable in Salesforce, for this we define another apex class with static Boolean variable and when our trigger runs for the first time we check the value of this Boolean variable and if it is true we allow further processing and in addition to further processing we set the value of this Boolean variable to false so as to avoid the second run of trigger in case if trigger fire another time due to updates.

Let us understand how we should write our code to avoid recursion.

APEX TRIGGER:

```
trigger AccountMainTrigger on Account (after update) {
   createContactClass obj=new createContactClass();
   if(avoidRecursionClass.booleanvar){   // Checking variable value for first run.
      avoidRecursionClass.booleanvar=false;   // Setting the static variable value to false
toavoid next run.
         if(trigger.isafter && trigger.isupdate){
           obj.method1(trigger.new,trigger.old);
         }
   }
}
```

APEX CLASS:

```
public class avoidRecursionClass {
 public static boolean booleanvar=true;
}
```