Apex Testing

Get Started With Apex Unit Tests

VerifyDate

```
public class VerifyDate {


    //method to handle potential checks against two dates

    public static Date CheckDates(Date date1, Date date2) {

        //if date2 is within the next 30 days of date1, use date2.  Otherwise use the end of the month

        if(DateWithin30Days(date1,date2)) {

            return date2;

        } else {

            return SetEndOfMonthDate(date1);

        }

    }


    //method to check if date2 is within the next 30 days of date1

    @TestVisible private static Boolean DateWithin30Days(Date date1, Date date2) {

        //check for date2 being in the past

    if( date2 < date1) { return false; }
```

```
        //check that date2 is within (>=) 30 days of date1

        Date date30Days = date1.addDays(30); //create a date 30 days away from date1

                if( date2 >= date30Days ) { return false; }

                else { return true; }

        }


        //method to return the end of the month of a given date

        @TestVisible private static Date SetEndOfMonthDate(Date date1) {

                Integer totalDays = Date.daysInMonth(date1.year(), date1.month());

                Date lastDay = Date.newInstance(date1.year(), date1.month(), totalDays);

                return lastDay;

        }


}
```

TestVerifyDate

```
@isTest

public class TestVerifyDate {


   @isTest static void Test_CheckDates_case1(){

      Date D =VerifyDate.CheckDates(date.parse('01/01/2020'), date.parse('01/05/2020'));

      System.assertEquals(date.parse('01/05/2020'),D);

   }
   @isTest static void Test_CheckDates_case2(){

      Date D =VerifyDate.CheckDates(date.parse('01/01/2020'), date.parse('05/05/2020'));

      System.assertEquals(date.parse('01/31/2020'),D);

   }


   @isTest static void Test_DateWithin30Days_case1(){

      Boolean flag=VerifyDate.DateWithin30Days(date.parse('01/01/2020'),
date.parse('12/30/2019'));

      System.assertEquals(false,flag);

   }


   @isTest static void Test_DateWithin30Days_case2(){

      Boolean flag=VerifyDate.DateWithin30Days(date.parse('01/01/2020'),
date.parse('02/02/2020'));

      System.assertEquals(false,flag);

   }


   @isTest static void Test_DateWithin30Days_case3(){

      Boolean flag=VerifyDate.DateWithin30Days(date.parse('01/01/2020'),
date.parse('01/15/2020'));

      System.assertEquals(true,flag);

   }


   @isTest static void Test_SetEndOfMonthDate(){

      Date returndate=VerifyDate.SetEndOfMonthDate(date.parse('01/01/2020'));
   }



}
```

Test Apex Triggers

RestrictContactByName

```
trigger RestrictContactByName on Contact (before insert, before update) {

        //check contacts prior to insert or update for invalid data
        For (Contact c : Trigger.New) {
                if(c.LastName == 'INVALIDNAME') {    //invalidname is invalid
                        c.AddError('The Last Name "'+c.LastName+'" is not allowed for DML');
                }

        }



}
```

TestRestrictContactByName

```
@isTest

public class TestRestrictContactByName {



    @isTest static void Test_insertupdateContact(){

        Contact cnt=new Contact();

        cnt.LastName='INVALIDNAME';

        Test.startTest();

        Database.SaveResult result=Database.insert(cnt,false);

        Test.stopTest();



        System.assert(!result.isSuccess());

        System.assert(result.getErrors().size()>0);

        System.assertEquals('The Last Name "INVALIDNAME" is not allowed for DML',result.getErrors()[0].getMessage());

    }



}
```

Create Test Data for Apex Tests

RandomContactFactory

```
public class RandomContactFactory {

    public static List<Contact> generateRandomContacts(Integer numcnt,string lastname){
        List<Contact> contacts = new List<Contact>();
        for(Integer i=0;i<numcnt;i++){
            Contact cnt=new Contact(FirstName = 'Test' +i, LastName= lastname);
            contacts.add(cnt);
        }
        return contacts;
    }

}
```