public  class jun30Assignments

{

    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Assignmentno.01\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

    public static void task1(list<Opportunity> oppObj,map<id, Opportunity> oppObj1)

    {

        for(Opportunity opp:oppObj)

        {

            if(opp.StageName=='Closed Won' || opp.StageName=='Closed Lost')

            {

                opp.CloseDate = System.today();

            }

        }

    }

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Assignmentno.02\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

    public static void task2(list<Calculator\_\_c> calObj,map<id,calculator\_\_C> calObj1)

    {

        for(Calculator\_\_c c1:calObj)

        {

                if(c1.Number1\_\_c== null && c1.Number2\_\_c== null)

                {

                         c1.addError(Calculator\_\_c.Number1\_\_c,'Enter the value of Number1');

                         c1.addError(Calculator\_\_c.Number2\_\_c,'Enter the value of number2');

                }

            else

            {

               // if(c1.Number1\_\_c!= calObj1.get(c1.Id).Number1\_\_c || c1.Number2\_\_c!= calObj1.get(c1.Id).Number2\_\_c)

                //{

                    c1.Addition\_\_c= c1.Number1\_\_c + c1.Number2\_\_c;

                    c1.Substraction\_\_c = c1.Number1\_\_c - c1.Number2\_\_c;

                    c1.Multiplication\_\_c= c1.Number1\_\_c \* c1.Number2\_\_c;

                    if(c1.Number2\_\_c!=0)

                    {

                        c1.Division\_\_c= c1.Number1\_\_c / c1.Number2\_\_c;

                    }

                    else

                    {

                        c1.addError('Enter number2 except 0');

                    }

              // }

            }

        }

    }

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Assignmentno.03\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

    public static void task3(list<Account> accObj)

    {

        map<id, Account> acc= new Map<Id, Account>([select id,name,

        (select id from Opportunities where StageName='Closed Won' limit 1) from Account where id= :accObj]);

        for(Account a1: accobj)

        {

            if(!acc.get(a1.id).Opportunities.isEmpty())

            {

                a1.addError('this record can not be deleted by any one');

            }

        }

    }

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Assignmentno.04\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

    public static void task4(list<lead> ldObj)

    {

        list<string> str =new list<string>();

        for(lead l1:ldObj)

        {

            if(l1.Email!= null )

            {

                str.add(l1.Email);

            }

            list<lead> ldObj2= [select ID,email from lead where Email in: str ];

            list<string> str2 =new list<string>();

            for(lead l2: ldObj2)

            {

                str2.add(l2.Email);

            }

            for(lead l2: ldObj)

            {

                if(str2.contains(l2.Email))

                    l2.addError('The entered email that are already existed');

            }

        }

    }

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Assignment no.5\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    public static void task5(list<Account> AccSquare)

    {

        for(Account a1: AccSquare)

        {

                a1.BillingCity = a1.ShippingCity;

                a1.BillingStreet = a1.ShippingStreet;

                a1.BillingPostalCode = a1.ShippingPostalCode;

                a1.BillingState = a1.ShippingState;

                a1.BillingCountry = a1.ShippingCountry;

                a1.BillingGeocodeAccuracy = a1.ShippingGeocodeAccuracy;

                a1.BillingLongitude = a1.ShippingLongitude;

                a1.BillingLatitude = a1.ShippingLatitude;

        }

    }

}