```
using System.Data;

/*
 * Title:   UtilityCharacter
 * Author:  Paul McKillop
 * Date:    26 December 2019
 * Purpose: Class to validate and score string characters
 */

namespace SignatureGeneratorV
{
    public class UtilityCharacter
    {
        /// <summary>
        /// Find and return a score as a string
        /// In use, check for string.empty
        /// </summary>
        /// <param name="suppliedCode"></param>
        /// <returns></returns>
        public static string CharacterScore(string suppliedCode, string path)
        {
            //-- variables to hold working values
            string foundScore = string.Empty;

            //-- initialise a new DataTable for data handling
            DataTable characterData = new DataTable();

            //-- populate the DataTable
            characterData = UtilityCharacterDb.GetCharacterData(path);

            //-- Error trap and throw error
            try
            {
                foreach (DataRow row in characterData.Rows)
                {
                    //-- object to hold current row values for checking
                    var currentCharacter = new Character()
                    {
                        Code = row.Field<string>(0),
                        Score = row.Field<string>(1)
                    };

                    // -- check if current row's data matches the inputted code
                    if (currentCharacter.Code == suppliedCode)
                    {
                        foundScore = currentCharacter.Score;
                    }
                }
            }
            catch (System.Exception ex)
            {

                throw ex;
            }

            // -- return current score as STRING.
            return foundScore;
        }
    }
}

```