```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;


/*  TITLE:      Loancalculator
 *  AUTHOR:     Paul McKillop
 *  DATE:       October 2022
 *  PURPOSE:    To calculate the payment that will be made against a loan
 */


namespace McKillopMotoring
{
    public class LoanCalculator
    {
        //-- create a constant used multiple time in calculations
        private const int MonthsPerYear = 12;

        #region Monthly payments
        /// <summary>
        /// Calculate the monthly costs
        /// </summary>
        /// <param name="myLoan"></param>
        /// <returns>decimal</returns>
        public static decimal LoanMonthlyPayment(Loan myLoan)
        {
            //-- create and initialise the return variable
            double payment = 0;

            //-- create local variables from the argument class
            var loanTermMonths = myLoan.LoanTermYears * MonthsPerYear;
            var interestRate = Convert.ToDouble(value: myLoan.LoanRate);
            var loanAmount = myLoan.CarPrice - myLoan.CarDeposit;

            if (myLoan.LoanTermYears > 0) //-- Check term years not 0
            {
                if (myLoan.LoanRate != 0)
                {
                    var rate = (double)((interestRate / MonthsPerYear) / 100);
                    var factor = (rate + (rate / (Math.Pow(x: rate + 1, y: loanTermMonths) - 1))); //-- use built-in library for compound interest
                    payment = (loanAmount * factor);
                }
                else
                {
                    payment = (loanAmount / (double)loanTermMonths);
                }
            }

            return Math.Round(d: (decimal)payment, decimals: 2);
        }
        #endregion


        #region Annual payments
        /// <summary>
        /// Annual Payments
        /// </summary>
        /// <param name="myLoan"></param>
        /// <returns>decimal</returns>
        public static decimal LoanAnnualPayment(Loan myLoan)
        {
            return Math.Round(d: LoanMonthlyPayment(myLoan) *
              12, decimals: 2);
        }
        #endregion


        #region Weekly payments
        /// <summary>
        /// Weekly payments
        /// </summary>
        /// <param name="myLoan"></param>
        /// <returns></returns>
        public static decimal LoanWeeklyPayment(Loan myLoan)
        {
            return Math.Round(d: LoanAnnualPayment(myLoan) / 52, decimals:
              2);
        }

        public static decimal LoanTotalPayment(Loan myLoan)
        {
            return Math.Round(d: LoanAnnualPayment(myLoan) *
              myLoan.LoanTermYears, decimals: 2);
        }
        #endregion

    }
```