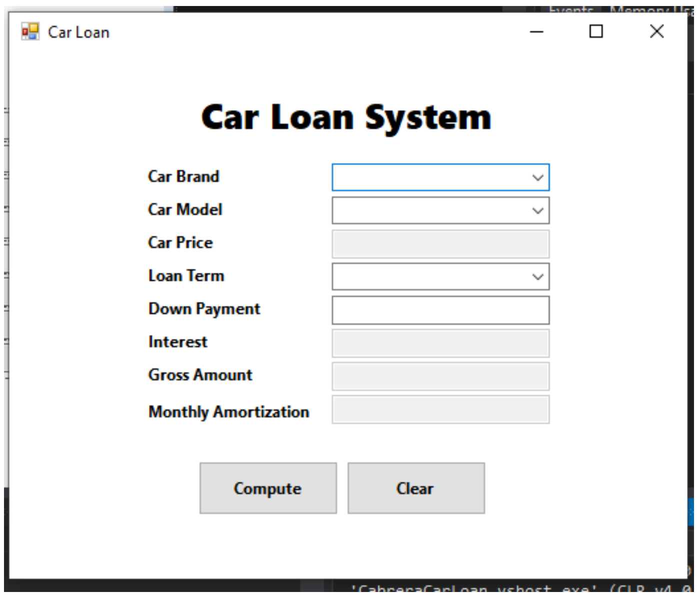
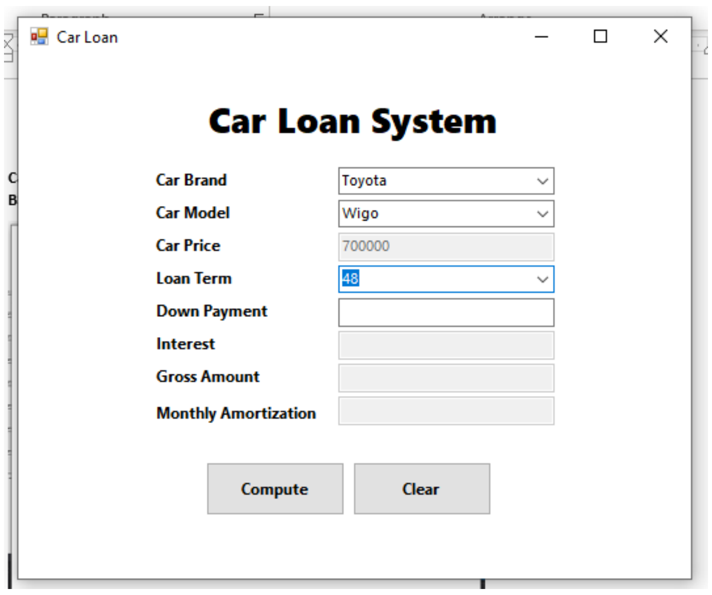
Cabrera, Jen Jade B.
BSCS – 2nd Year



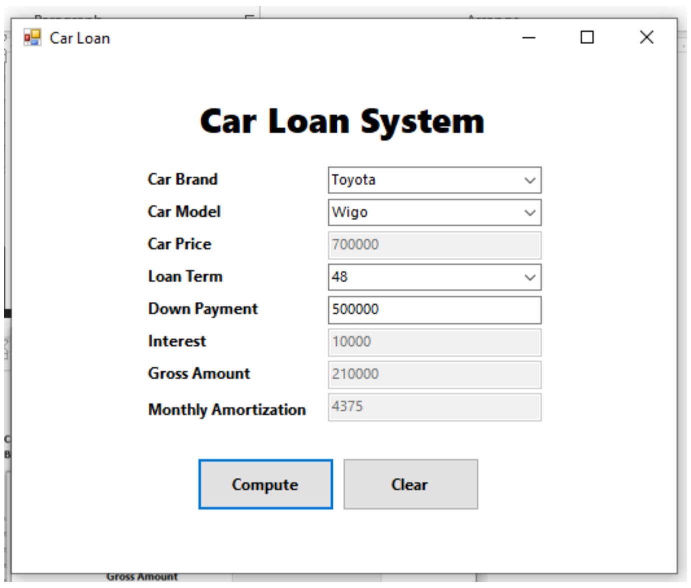
Car Loan System

Car Brand	
Car Model	
Car Price	
Loan Term	
Down Payment	
Interest	
Gross Amount	
Monthly Amortization	



Car Loan System

Car Brand	Toyota
Car Model	Wigo
Car Price	700000
Loan Term	48
Down Payment	
Interest	
Gross Amount	
Monthly Amortization	



Car Loan System

Car Brand	Toyota
Car Model	Wigo
Car Price	700000
Loan Term	48
Down Payment	500000
Interest	10000
Gross Amount	210000
Monthly Amortization	4375

LoanClient

+ BrandModelMap: Dictionary<string, string[]> <<static>>
+ ModelPriceMap: Dictionary<string, double> <<static>>
+ TermsInterestMap: Dictionary<int, double> <<static>>
+ DownPayment: double
+ CarBrand: string
+ CarModel: string
+ LoanTerm: int
- _carPrice: double
- _priceLessDownPayment: double
- _interest: double
- _interestRate: double
- _grossAmount: double
- _monthlyAmortization: double

- ComputeCarPrice(): void
- ComputePriceLessDownPayment(): void
- ComputeInterest(): void
- ComputeGrossAmount(): void
- ComputeMonthlyAmortization(): void
+ GetCarPrice(): double
+ GetInterest(): double
+ GetGrossAmount(): double
+ GetMonthlyAmortization(): double

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

namespace CabreraCarLoan {
    class LoanClient {

        // Declaring as static as every such value of the instances doesn't have to be different
        public static Dictionary<string, string[]> BrandModelMap = new Dictionary<string, string[]> {
            {"Toyota", new string[] {"Raize", "Veloz", "Wigo", "Vios", "Rush", }},
            {"Nissan", new string[] {"Kicks", "NV350", "Livina", "Z", }},
        };

        public static Dictionary<string, double> ModelPriceMap = new Dictionary<string, double> {
            {"Raize", 1000000}, {"Veloz", 1250000}, {"Wigo", 700000},
            {"Vios", 1035000}, {"Rush", 1170000}, {"Kicks", 1500000},
            {"NV350", 2000000}, {"Livina", 1200000}, {"Z", 2500000},
        };

        public static Dictionary<int, double> TermInterestMap = new Dictionary<int, double> {
            {24, 0.15}, {36, 0.10}, {48, 0.05}, {60, 0.03},
        };

        // Properties
        public double DownPayment { get; set; }
        public string CarBrand { get; set; }
        public string CarModel { get; set; }
        public int LoanTerm { get; set; }

        // Fields
        private double _carPrice;
        private double _priceLessDownPayment;
        private double _interest;
        private double _interestRate;
        private double _grossAmount;
        private double _monthlyAmortization;

        // Constructor
        public LoanClient() { }

        // Methods
        private void ComputeCarPrice() {
            _carPrice = ModelPriceMap[CarModel];
        }

        private void ComputePriceLessDownPayment() {
            _priceLessDownPayment = _carPrice - DownPayment;
        }

        private void ComputeInterest() {
            _interestRate = TermInterestMap[LoanTerm];
            _interest = _priceLessDownPayment * _interestRate;
        }

        private void ComputeGrossAmount() {
            _grossAmount = _priceLessDownPayment + _interest;
        }

        private void ComputeMonthlyAmortization() {
            _monthlyAmortization = _grossAmount / LoanTerm;
        }

        public double GetCarPrice() {
            ComputeCarPrice();
            return _carPrice;
        }

        public double GetInterest() {
            ComputeCarPrice();
            ComputePriceLessDownPayment();
            ComputeInterest();
            return _interest;
        }

        public double GetGrossAmount() {
            ComputeCarPrice();
            ComputePriceLessDownPayment();
            ComputeInterest();
            ComputeGrossAmount();
            return _grossAmount;
        }

        public double GetMonthlyAmortization() {
            ComputeCarPrice();
            ComputePriceLessDownPayment();
            ComputeInterest();
            ComputeGrossAmount();
            ComputeMonthlyAmortization();
            return _monthlyAmortization;
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CabreraCarLoan {
    public partial class MainForm : Form {
        LoanClient loanClient = new LoanClient();
        private double _carPrice;
        private double _downPayment;
        private double _interest;
        private double _grossAmount;
        private double _monthlyAmortization;

        public MainForm() {
            InitializeComponent();
        }

        private void LoadItemsInCarBrand() {
            foreach (KeyValuePair<string, string[]> dictionary in LoanClient.BrandModelMap) {
                cmbCarBrand.Items.Add(dictionary.Key);
            }
        }

        private void LoadItemsInCarModel() {
            foreach (string model in LoanClient.BrandModelMap[loanClient.CarBrand]) {
                cmbCarModel.Items.Add(model);
            }
        }

        private void LoadItemsInLoanTerm() {
            foreach (int loanTerm in LoanClient.TermInterestMap.Keys) {
                cmbLoanTerm.Items.Add(loanTerm.ToString());
            }
        }

        private void ComputeOutputFields() {
            _carPrice = loanClient.GetCarPrice();
            _interest = loanClient.GetInterest();
            _grossAmount = loanClient.GetGrossAmount();
            _monthlyAmortization = loanClient.GetMonthlyAmortization();
        }

        private void ClearOutputFields() {
            txtCarPrice.Clear();
            txtInterest.Clear();
            txtGrossAmount.Clear();
            txtMonthlyAmortization.Clear();
        }

        private void ClearInputFields() {
            cmbCarBrand.ResetText();
            cmbCarModel.ResetText();
            cmbLoanTerm.ResetText();
            txtDownPayment.Clear();
        }

        private void MainForm_Load(object sender, EventArgs e) {
            LoadItemsInCarBrand();
            LoadItemsInLoanTerm();
        }

        private void cmbCarBrand_SelectedIndexChanged(object sender, EventArgs e) {
            ClearOutputFields();
            cmbCarModel.Items.Clear();
            cmbCarModel.ResetText();

            loanClient.CarBrand = cmbCarBrand.Text;
            LoadItemsInCarModel();
        }

        private void cmbCarModel_SelectedIndexChanged(object sender, EventArgs e) {
            loanClient.CarModel = cmbCarModel.Text;
            ClearOutputFields();
            txtCarPrice.Text = loanClient.GetCarPrice().ToString();
        }

        private void cmbLoanTerm_SelectedIndexChanged(object sender, EventArgs e) {
            loanClient.LoanTerm = int.Parse(cmbLoanTerm.Text);
        }

        private void btnCompute_Click(object sender, EventArgs e) {
            try {
                _downPayment = double.Parse(txtDownPayment.Text);
                loanClient.DownPayment = _downPayment;

                ComputeOutputFields();
                txtInterest.Text = _interest.ToString();
                txtGrossAmount.Text = _grossAmount.ToString();
                txtMonthlyAmortization.Text = _monthlyAmortization.ToString();
            } catch (Exception) {
                ClearInputFields();
                ClearOutputFields();
            }
        }

        private void btnClear_Click(object sender, EventArgs e) {
            ClearInputFields();
            ClearOutputFields();
        }
    }
}
```