

Car Rentals Project 2

Modify Project 1 to use a Presentation Tier and a Business Services tier

[Project 2 Video](#)

[Project 2 Description](#)

Presentation Tier

The screenshot shows a Windows-style application window titled "Cook's Car Rentals Assignment 2 Multi-tier". Inside, there's a "Rentals" form. The form has several input fields and buttons. The "Customer Name" field is filled with "Alan Cook". The "Address" field is filled with "123 Easy St". Below the address, there are two small dropdown menus, one showing "Anywhere" and another showing "AL". The "Phone" field is empty. The "Driver's License" field is filled with "1235". The "Type of Credit Card" dropdown menu is open, showing "Visa" (selected), "Mastercard", and "American Express". The "Credit Card Number" field is filled with "12345". To the right of the form, the "Amount Due" is displayed as "\$76.81". Below the form, there are two groups of radio buttons: "Select a Car Size" with options "Compact" (selected), "Mid-Size", and "Luxury"; and "Discounts" with options "Corporate" (selected) and "Insurance Account". At the bottom of the form, there are three input fields: "Number of Days Rented" (filled with "3"), "Beginning Odometer" (filled with "1500"), and "Ending Odometer" (filled with "2000"). To the right of these fields are two buttons: "Calculate" and "Close". The status bar at the bottom of the window reads "We are #1 in Car Rentals" on the left and "2/2/2012 8:31:10 AM" on the right.

- Validate beginning odometer reading is less than ending odometer reading
- Validate data entry using Validating Event as shown below
 - Driver's license
 - Credit Card
 - Days rented

```
private void BeginOdometerTextBox_Validating(object sender, CancelEventArgs e)
{
    // Test the entry for numeric.
    Decimal NumberDecimal;

    Decimal.TryParse(BeginOdometerTextBox.Text, out NumberDecimal);
    if (NumberDecimal == 0)
    {
        ErrorProvider1.SetError(BeginOdometerTextBox, "Entry Required.");
        BeginOdometerTextBox.Focus();
        e.Cancel = true;
    }
    else
        ErrorProvider1.Clear();
}
```

Business Services Tier

- Create a RentalRate class **using the code in the link below as is**
 - [RentalRate Class](#)
- Create a Coporate rate sub-class of RentalRate
 - [CoporateRate Class](#)
 - Properties shown below
 - Methods
 - Constructor shown below
 - FindAmountDue() overrides base mehtod
 - Add enum Discont shown below
 -
-

```
public enum Discount : int
{
    Corporate = 1,
    Insurance
}

class CorporateRate : RentalRate
{
    private int DiscountRateInteger;
    const Decimal CORPORATE_DISCOUNT_Decimal = 0.05M;
    const Decimal INSURANCE_DISCOUNT_Decimal = 0.1M;

    public CorporateRate(int BeginningOdometerInteger, int EndingOdometerInteger,
        int CarSizeInteger, int DaysInteger, int DiscountInteger)
        : base(BeginningOdometerInteger, EndingOdometerInteger, CarSizeInteger, DaysInteger)
    {
        DiscountRateInteger = DiscountInteger;
        FindAmountDue();
    }
}
```

