

## 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 contains "Alan Cook". The "Address" field contains "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 contains "1235". The "Type of Credit Card" dropdown menu is open, showing "Visa" (selected), "Mastercard", and "American Express". The "Credit Card Number" field contains "12345". To the right of these fields, the "Amount Due" is displayed as "\$76.81". Below the "Credit Card Number" field, there are two groups of radio buttons. The first group, "Select a Car Size", has "Compact" selected. The second group, "Discounts", has "Corporate" selected. Below these, there are three more input fields: "Number of Days Rented" (3), "Beginning Odometer" (1500), and "Ending Odometer" (2000). To the right of these fields, there are two buttons: "Calculate" and "Close". At the bottom of the window, there's a status bar that says "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 Corporate rate sub-class of RentalRate
  - [CorporateRate Class](#)
  - Properties shown below
  - Methods
    - Constructor shown below
    - FindAmountDue() overrides base method
  - Add enum Discount 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();
    }
}
```

