```
// Program 2
// CIS 200-01
// Fall 2017
// Due: 10/23/2017
// By: C5503

// File: AddressForm.cs
// This form is used to input information about the person shipping the package and is
used to insert additional addresses into the list of addresses.
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 Prog2
{
    public partial class AddressForm : Form
    {
        ErrorProvider errorProvider1 = new ErrorProvider(); // Error provider created to
test validation

        public AddressForm()
        {
            InitializeComponent();
        }
        //Preconditions: None
        //Postconditions: Name input is returned
        public string NameInput
        {
            get
            {
                return txtName.Text;
            }

            set
            {
                txtName.Text = value;
            }
        }

        //Preconditions: None
        //Postconditions: AddressOne input is returned
        public string AddressOneInput
        {
            get
            {
                return txtAddressOne.Text;
            }

            set
            {
                txtAddressOne.Text = value;
            }
```

```
}

//Preconditions: None
//Postconditions: AddressTwo input is returned
public string AddressTwoInput
{
    get
    {
        return txtAddressTwo.Text;
    }

    set
    {
        txtAddressTwo.Text = value;
    }
}
//Preconditions: None
//Postconditions: City input is returned
public string CityInput
{
    get
    {
        return txtCity.Text;
    }

    set
    {
        txtCity.Text = value;
    }
}
//Preconditions: None
//Postconditions: State input is returned
public string StateInput
{
    get
    {
        return cbState.Text;
    }
}
//Preconditions: None
//Postconditions: Zip input is returned
public string ZipInput
{
    get
    {
        return txtZip.Text;
    }

    set
    {
        txtZip.Text = value;
    }
}



//Preconditions: None
//Postconditions: Name textbox is tested for proper validation
```

```
private void TxtName_Validating(object sender, CancelEventArgs e)
{
    if (string.IsNullOrWhiteSpace(txtName.Text))
    {
        e.Cancel = true;
        errorProvider1.SetError(txtName, "Error");
    }
}
//Preconditions: Input is required
//Postconditions: Input validated and error will not show
private void TxtName_Validated(object sender, EventArgs e)
{
    errorProvider1.SetError(txtName, "");

}

//Preconditions: None
//Postconditions: Address textbox is tested for proper validation
private void TxtAddress_Validating(object sender, CancelEventArgs e)
{
    if (string.IsNullOrWhiteSpace(txtAddressOne.Text))
    {
        e.Cancel = true;
        errorProvider1.SetError(txtName, "Error");
    }
}

//Preconditions: Input is required
//Postconditions: Input validated and error will not show
private void TxtAddress_Validated(object sender, EventArgs e)
{
    errorProvider1.SetError(txtAddressOne, "");
}

//Preconditions: None
//Postconditions: City textbox is tested for proper validation
private void TxtCity_Validating(object sender, CancelEventArgs e)
{
    if (string.IsNullOrWhiteSpace(txtCity.Text))
    {
        e.Cancel = true;
        errorProvider1.SetError(txtCity, "Error");
    }
}

//Preconditions: Input is required
//Postconditions: Input validated and error will not show
private void TxtCity_Validated(object sender, EventArgs e)
{
    errorProvider1.SetError(txtCity, "");
}

//Preconditions: None
//Postconditions: State combobox is tested for proper validation
private void CbState_Validating(object sender, CancelEventArgs e)
{
    if (cbState.SelectedIndex == -1)
    {
```

```
                e.Cancel = true;
                errorProvider1.SetError(cbState, "Error");
            }
        }

        //Preconditions: Input is required
        //Postconditions: Input validated and error will not show
        private void CbState_Validated(object sender, EventArgs e)
        {
            errorProvider1.SetError(cbState, "");
        }

        //Preconditions: None
        //Postconditions: Zip textbox is tested for proper validation
        private void TxtZip_Validating(object sender, CancelEventArgs e)
        {
            const int MIN_ZIP = 0;
            const int MAX_ZIP = 99999;

            if (!int.TryParse(txtZip.Text, out int zip) || zip < MIN_ZIP || zip >
MAX_ZIP)
            {
                e.Cancel = true;
                errorProvider1.SetError(txtZip, "Error");
            }
        }

        //Preconditions: Input is required
        //Postconditions: Input validated and error will not show
        private void TxtZip_Validated(object sender, EventArgs e)
        {
            errorProvider1.SetError(txtZip, "");
        }

        //Preconditions: OK button needs to be pressed
        //Postconditions: All input is validated on button press
        private void BtnOK_Click(object sender, EventArgs e)
        {
            if (this.ValidateChildren())
                this.DialogResult = DialogResult.OK;
        }

        //Preconditions: Cancel button needs to be pressed
        //Postconditions: AddressForm closes
        private void BtnCancel_Click_1(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

```
// Program 2
// CIS 200-01
// Fall 2017
// Due: 10/23/2017
// By: C5503

// File: LetterForm.cs
// This form is used to insert the origin address and destination address of a letter

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 Prog2
{
    public partial class LetterForm : Form
    {
        List<Address> addresses = new List<Address>();

        ErrorProvider errorProvider1 = new ErrorProvider(); // Created errorprovider to
be used for validation of LetterForm inputs

        public LetterForm()
        {
            InitializeComponent();
        }

        //Preconditions:None
        //Postconditions:LetterForm is loaded and addresses are added
        private void LetterForm_Load(object sender, EventArgs e)
        {
            foreach (Address address in addresses)
                cbOriginAddress.Items.Add(address.Name);

            foreach (Address address in addresses)
                cbDestinationAddress.Items.Add(address.Name);
        }

        //Preconditions: Origin address must be selected from combobox
        //Postconditions: Origin address is returned
        public int OriginAddress
        {
            get
            {
                return cbOriginAddress.SelectedIndex;
            }
        }

        //Preconditions: Destination address must be selected from combobox
        //Postconditions: Destination address is returned
        public int DestinationAddress
        {
```

```
            get
            {
                return cbDestinationAddress.SelectedIndex;
            }
        }



        //Preconditions: None
        //Postconditions: Origin address combobox is tested for proper validation
        private void CbOriginAddress_Validating(object sender, CancelEventArgs e)
        {
            if (cbOriginAddress.SelectedIndex == -1)
            {
                e.Cancel = true;
                errorProvider1.SetError(cbOriginAddress, "Origin Address must be
selected");
            }
        }
        //Preconditions: origin address must be selected
        //Postconditions: After validation, error will not show
        private void CbOriginAddress_Validated(object sender, EventArgs e)
        {
            errorProvider1.SetError(cbOriginAddress, "");
        }

        //Preconditions: None
        //Postconditions: Destination address combobox is tested for proper validation
        private void CbDestinationAddress_Validating(object sender, CancelEventArgs e)
        {
            if (cbDestinationAddress.SelectedIndex == -1)
            {
                e.Cancel = true;
                errorProvider1.SetError(cbDestinationAddress, "Destination Address must
be selected");
            }
        }
        //Preconditions: origin address must be selected
        //Postconditions: After validation, error will not show
        private void CbDestinationAddress_Validated(object sender, EventArgs e)
        {
            errorProvider1.SetError(cbDestinationAddress, "");
        }

        //Preconditions: OK button needs to be pressed
        //Postconditions: All input is validated on button press
        private void BtnOK_Click(object sender, EventArgs e)
        {
            if (ValidateChildren())
                this.DialogResult = DialogResult.OK;
        }
        //Preconditions: Cancel button needs to be pressed
        //Postconditions: LetterForm closes
        private void BtnCancel_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
```

```
}
```