```
namespace CustomerMaintenance
{
    4 references
    public partial class frmAddCustomer : Form
    {
        1 reference
        public frmAddCustomer()
        {
            InitializeComponent();
        }

        private Customer customer = null;

        1 reference
        public Customer GetNewCustomer()
        {
            this.ShowDialog();
            return customer;
        }

        1 reference
        private void btnSave_Click(object sender, EventArgs e)
        {
            if (IsValidData())
            {
                customer = new Customer(txtFirstName.Text, txtLastName.Text,
                    txtEmail.Text);
                this.Close();
            }
        }

        1 reference
        private bool IsValidData()
        {
            return Validator.IsPresent(txtFirstName) &&
                Validator.IsPresent(txtLastName) &&
                Validator.IsPresent(txtEmail) &&
                Validator.IsValidEmail(txtEmail);
        }
```

```
namespace CustomerMaintenance
{
    3 references
    public partial class frmCustomers : Form
    {
        1 reference
        public frmCustomers()
        {
            InitializeComponent();
        }

        private List<Customer> customers = null;

        1 reference
        private void frmCustomers_Load(object sender, EventArgs e)
        {
            customers = CustomerDB.GetCustomers();
            FillCustomerListBox();
        }

        3 references
        private void FillCustomerListBox()
        {
            lstCustomers.Items.Clear();
            foreach (Customer c in customers)
            {
                lstCustomers.Items.Add(c.GetDisplayText());
            }
        }

        1 reference
        private void btnAdd_Click(object sender, EventArgs e)
        {
            frmAddCustomer addCustomerForm = new frmAddCustomer();
            Customer customer = addCustomerForm.GetNewCustomer();
            if (customer != null)
            {
                customers.Add(customer);
                CustomerDB.SaveCustomers(customers);
                FillCustomerListBox();
            }
        }
```

```
private void btnDelete_Click(object sender, EventArgs e)
{
    int i = lstCustomers.SelectedIndex;
    if (i != -1)
    {
        Customer customer = (Customer)customers[i];
        string message = "Are you sure you want to delete "
            + customer.FirstName + " " + customer.LastName + "?";
        DialogResult button = MessageBox.Show(message, "Confirm Delete",
            MessageBoxButtons.YesNo);
        if (button == DialogResult.Yes)
        {
            customers.Remove(customer);
            CustomerDB.SaveCustomers(customers);
            FillCustomerListBox();
        }
    }
}

1 reference
private void btnExit_Click(object sender, EventArgs e)
{
    this.Close();
}
```

```
namespace CustomerMaintenance
{
    3 references
    public static class CustomerDB
    {
        private const string dir = @"C:\C# 2010\Files\";
        private const string path = dir + "Customers.txt";

        2 references
        public static void SaveCustomers(List<Customer> customers)
        {
            // create the output stream for a text file that exists
            StreamWriter textOut =
                new StreamWriter(
                new FileStream(path, FileMode.Create, FileAccess.Write));

            // write each customer
            foreach (Customer customer in customers)
            {
                textOut.Write(customer.FirstName + "|");
                textOut.Write(customer.LastName + "|");
                textOut.WriteLine(customer.Email);
            }

            // write the end of the document
            textOut.Close();
        }

        1 reference
        public static List<Customer> GetCustomers()
        {
            // if the directory doesn't exist, create it
            if (!Directory.Exists(dir))
                Directory.CreateDirectory(dir);

            // create the object for the input stream for a text file
            StreamReader textIn =
                new StreamReader(
                    new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read));

            // create the array list for customers
            List<Customer> customers = new List<Customer>();
```

```
namespace CustomerMaintenance
{
    3 references
    public static class CustomerDB
    {
        private const string dir = @"C:\C# 2010\Files\";
        private const string path = dir + "Customers.txt";

        2 references
        public static void SaveCustomers(List<Customer> customers)
        {
            // create the output stream for a text file that exists
            StreamWriter textOut =
                new StreamWriter(
                new FileStream(path, FileMode.Create, FileAccess.Write));

            // write each customer
            foreach (Customer customer in customers)
            {
                textOut.Write(customer.FirstName + "|");
                textOut.Write(customer.LastName + "|");
                textOut.WriteLine(customer.Email);
            }

            // write the end of the document
            textOut.Close();
        }
```

```
    public static List<Customer> GetCustomers()
    {
        // if the directory doesn't exist, create it
        if (!Directory.Exists(dir))
            Directory.CreateDirectory(dir);

        // create the object for the input stream for a text file
        StreamReader textIn =
            new StreamReader(
                new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read));

        // create the array list for customers
        List<Customer> customers = new List<Customer>();

        // read the data from the file and store it in the ArrayList
        while (textIn.Peek() != -1)
        {
            string row = textIn.ReadLine();
            string[] columns = row.Split('|');
            Customer customer = new Customer();
            customer.FirstName = columns[0];
            customer.LastName = columns[1];
            customer.Email = columns[2];
            customers.Add(customer);
        }

        textIn.Close();

        return customers;
    }
}
```

```
namespace CustomerMaintenance
{
    17 references
    public class Customer
    {
        private string firstName;
        private string lastName;
        private string email;

        1 reference
        public Customer()
        {
        }

        /// <summary>
        /// Establishes a variable to a paramater
        /// </summary>
        /// <param name="firstName">firstName paramater</param>
        /// <param name="lastName">lastName paramater</param>
        /// <param name="email">email parameter</param>
        1 reference
        public Customer(string firstName, string lastName, string email)
        {
            this.firstName = firstName;
            this.lastName = lastName;
            this.email = email;
        }

        /// <summary>
        /// Property for FirstName
        /// </summary>
        3 references
        public string FirstName
        {
            get
            {
                return firstName;
            }
            set
            {
                firstName = value;
            }
```