School of Information Technology

Practical 05: State Management with Objects

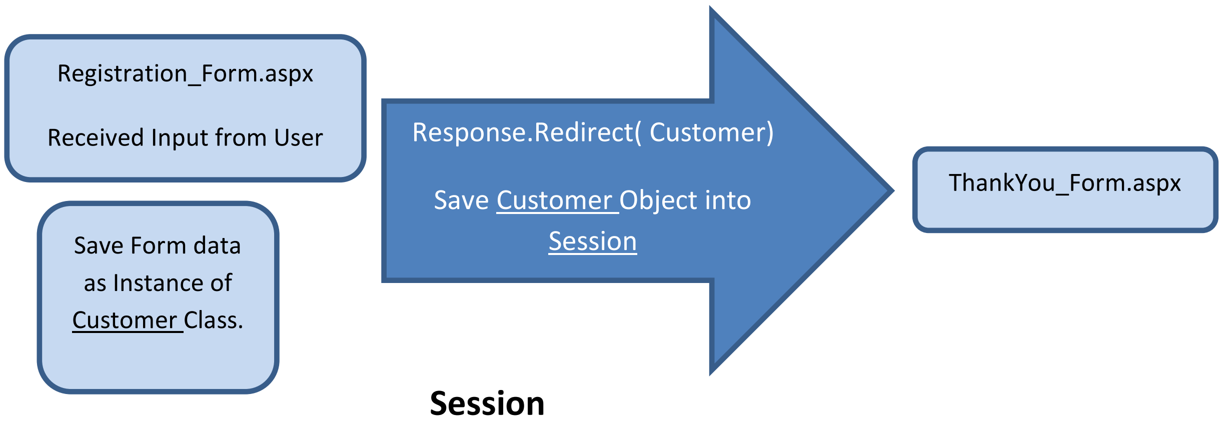
OBJECTIVES:

By the end of this Practical students should be able to:

- Save form data into class object
- Save the instance of a class into Session
- Extract data from class object

Instructions

For this practical exercise, we will continue from Practical 04 where we continue to use Session to temporarily store and send data between forms. In today's practical, we will improve Lab2_Registration_Session by saving the Form information into a Class object (e.g Customer Class). We will save the Instance of the Customer into the Session and learn how to extract the Form data as class properties. We will continue to use Registration Web application that allows user enter registration information in Registration_Form.aspx and send it to the ThankYou_Form.aspx for display. We will use the Session technique to pass information from one form to the other.

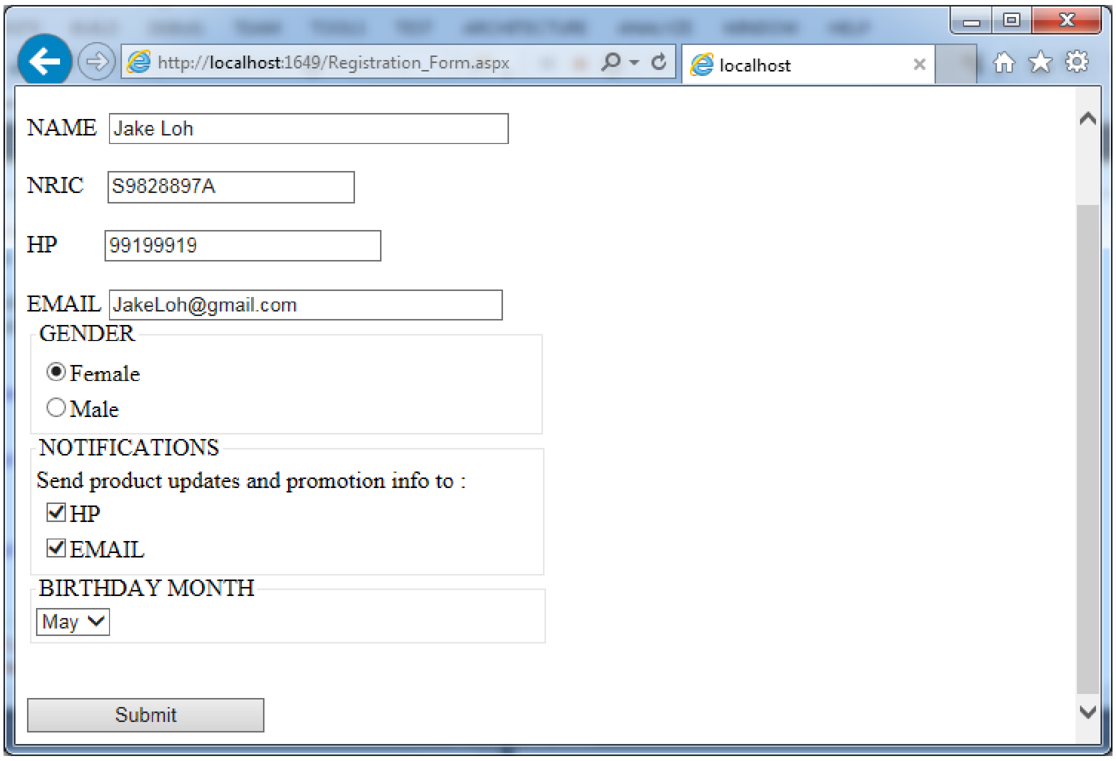

Registration_Form.aspx
Received Input from User
Save Form data as Instance of Customer Class.
Response.Redirect(Customer)
Save Customer Object into Session
ThankYou_Form.aspx
Session

Classes and Properties

A class is a group of related methods and variables. A class describes these things (e.g Customer), and in most cases, you create an instance of this class, now referred to as an object. On this object, you use the defined methods and variables. Of course, you can create as many instances of your class as you want to.

1. Create Customer class to keep an Instance of customer data when the Form is submitted.
2. Create class properties to access the data (Get, Set) of the Class or Object.

1. Run VS 2015 and open Lab2_Registration web site



http://localhost:1649/Registration_Form.aspx

localhost

NAME	Jake Loh
NRIC	S9828897A
HP	99199919
EMAIL	JakeLoh@gmail.com

GENDER

● Female
○ Male

NOTIFICATIONS

Send product updates and promotion info to :

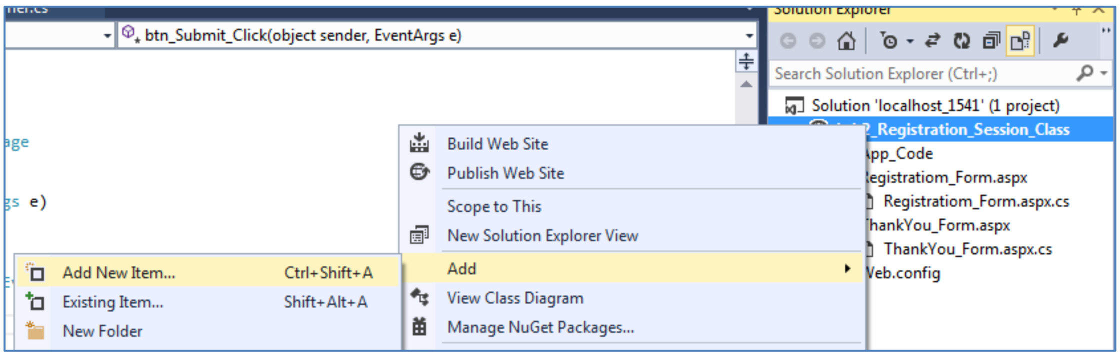
✓ HP
✓ EMAIL

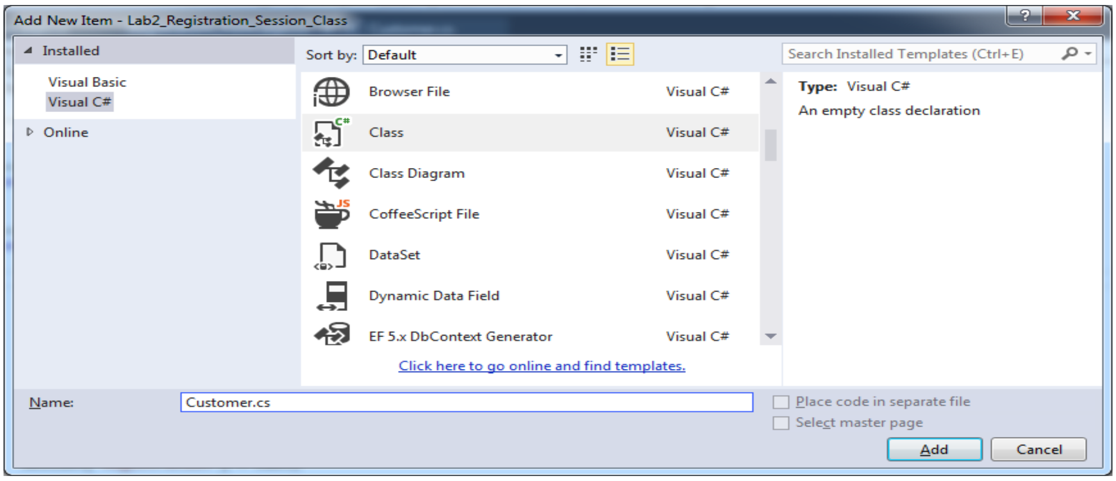
BIRTHDAY MONTH

May

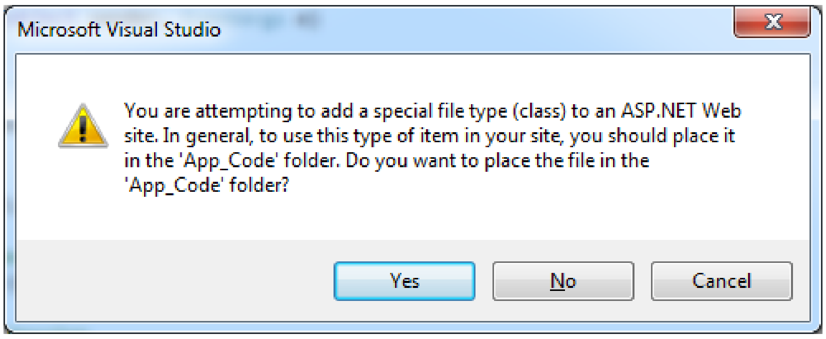
Submit

2. Create a new Class named it “Customer”. Right-click “Web site name” → choose “Add” → New Item”→ Class

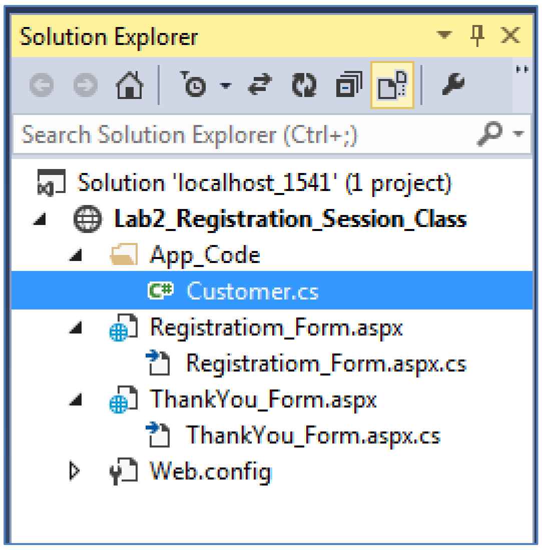

btn_Submit_Click(object sender, EventArgs e)
Add New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Folder
Build Web Site
Publish Web Site
Scope to This
New Solution Explorer View
Add
View Class Diagram
Manage NuGet Packages...
Search Solution Explorer (Ctrl+;)
Solution 'localhost_1541' (1 project)
_Registration_Session_Class
pp_Code
egistratiom_Form.aspx
Registration_Form.aspx.cs
hankYou_Form.aspx
ThankYou_Form.aspx.cs
Veb.config


Add New Item - Lab2_Registration_Session_Class
Installed
Visual Basic
Visual C#
Online
Sort by: Default
Browser File Visual C#
Class Visual C#
Class Diagram Visual C#
CoffeeScript File Visual C#
DataSet Visual C#
Dynamic Data Field Visual C#
EF 5.x DbContext Generator Visual C#
Click here to go online and find templates.
Search Installed Templates (Ctrl+E)
Type: Visual C#
An empty class declaration
Name: Customer.cs
Place code in separate file
Select master page
Add
Cancel

3. Select "Yes" to put the Customer Class into the "App_Code" folder.


Microsoft Visual Studio

You are attempting to add a special file type (class) to an ASP.NET Web site. In general, to use this type of item in your site, you should place it in the 'App_Code' folder. Do you want to place the file in the 'App_Code' folder?


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'localhost_1541' (1 project)
Lab2_Registration_Session_Class
App_Code
Customer.cs
Registratiom_Form.aspx
Registratiom_Form.aspx.cs
ThankYou_Form.aspx
ThankYou_Form.aspx.cs
Web.config

4. Describe the Class and define its properties

   Class

   a. Set the Customer Class as Public
   b. Create 2 Constructors
      i. Default Empty Constructor
      ii. A Constructor that takes in 7 Arguments
   c. Create 7 private variables to store the Form values

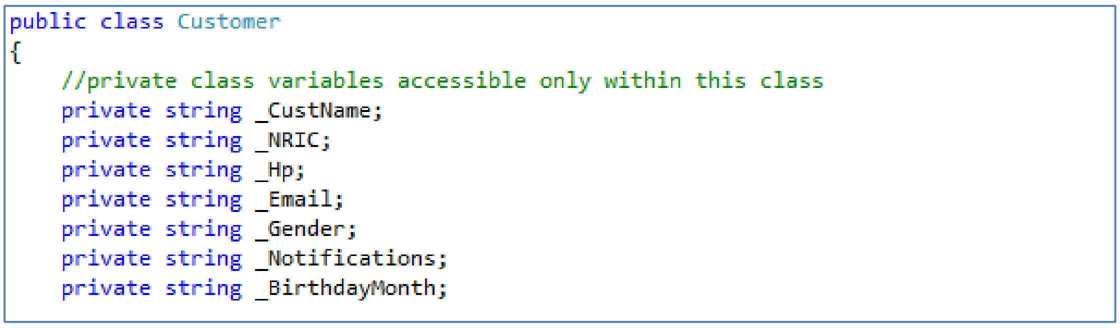
   Properties

   a. Create 7 Class properties.
   b. Set each class properties to the following settings :
      i. Data type as string,
      ii. Access Modifier – Public-Read, Public-Write

   Methods

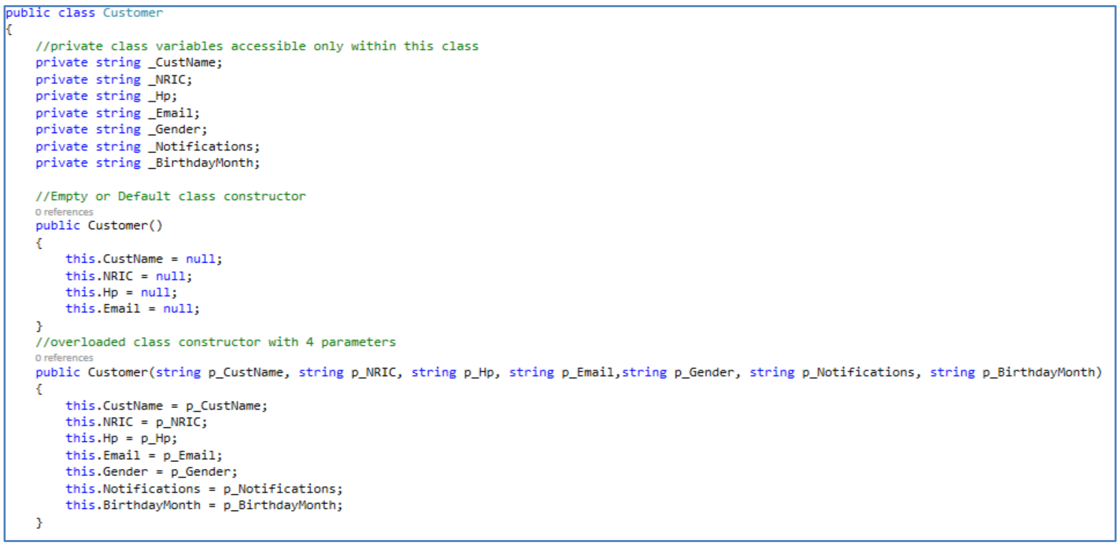
   a. Not required in this practical. We will revisit this in ADO.Net database.

5. Create the Class definition.
   a. Class variables



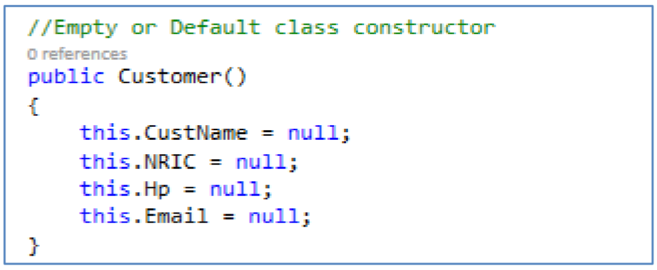
```
public class Customer
{
    //private class variables accessible only within this class
    private string _CustName;
    private string _NRIC;
    private string _Hp;
    private string _Email;
    private string _Gender;
    private string _Notifications;
    private string _BirthdayMonth;
```

   a. Constructors.

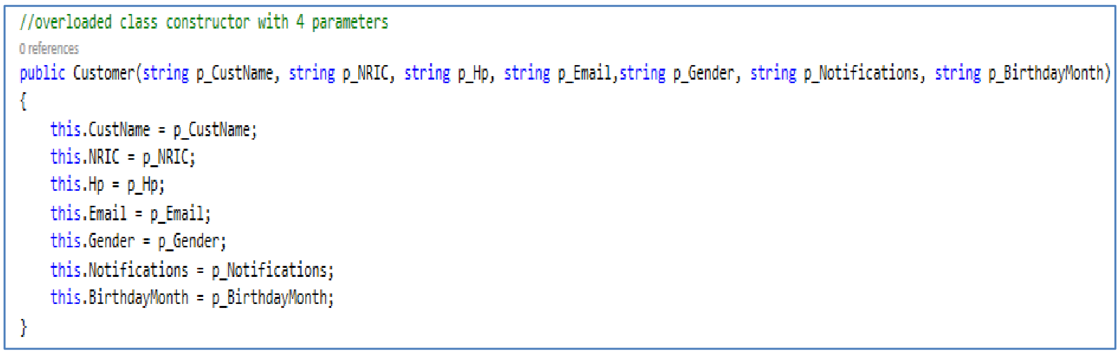


```
public class Customer
{
    //private class variables accessible only within this class
    private string _CustName;
    private string _NRIC;
    private string _Hp;
    private string _Email;
    private string _Gender;
    private string _Notifications;
    private string _BirthdayMonth;

    //Empty or Default class constructor
    0 references
    public Customer()
    {
        this.CustName = null;
        this.NRIC = null;
        this.Hp = null;
        this.Email = null;
    }
    //overloaded class constructor with 4 parameters
    0 references
    public Customer(string p_CustName, string p_NRIC, string p_Hp, string p_Email,string p_Gender, string p_Notifications, string p_BirthdayMonth)
    {
        this.CustName = p_CustName;
        this.NRIC = p_NRIC;
        this.Hp = p_Hp;
        this.Email = p_Email;
        this.Gender = p_Gender;
        this.Notifications = p_Notifications;
        this.BirthdayMonth = p_BirthdayMonth;
    }
```

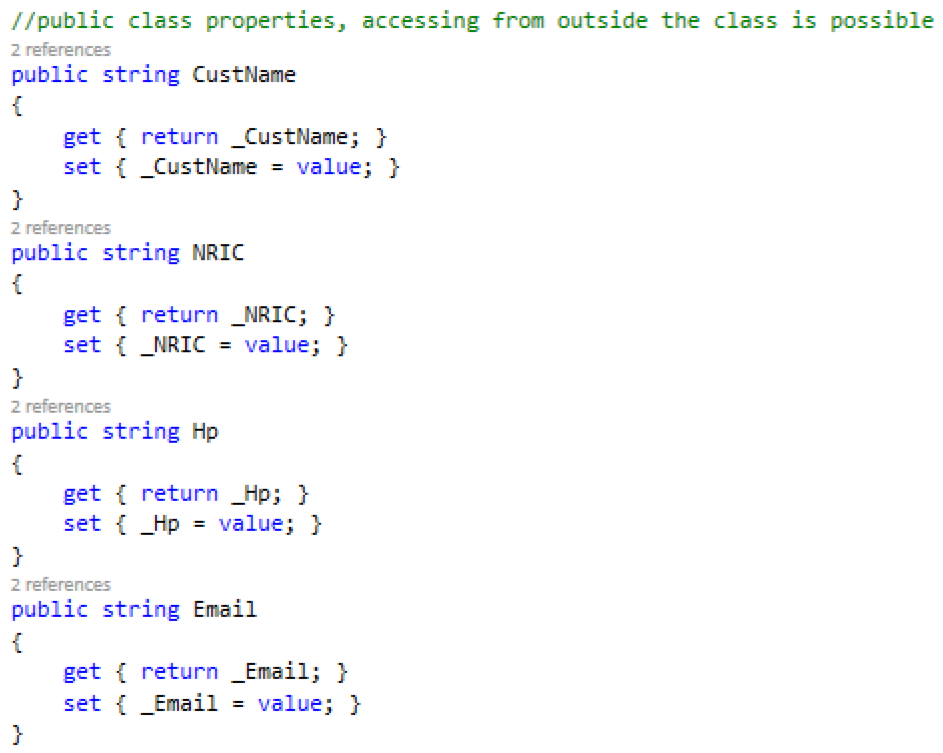


```
//Empty or Default class constructor
0 references
public Customer()
{
    this.CustName = null;
    this.NRIC = null;
    this.Hp = null;
    this.Email = null;
}
```

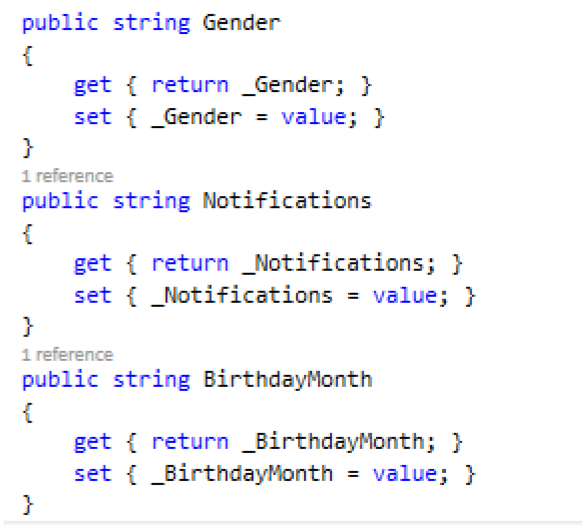


```
//overloaded class constructor with 4 parameters
0 references
public Customer(string p_CustName, string p_NRIC, string p_Hp, string p_Email,string p_Gender, string p_Notifications, string p_BirthdayMonth)
{
    this.CustName = p_CustName;
    this.NRIC = p_NRIC;
    this.Hp = p_Hp;
    this.Email = p_Email;
    this.Gender = p_Gender;
    this.Notifications = p_Notifications;
    this.BirthdayMonth = p_BirthdayMonth;
}
```

6. Create the 7 Class properties with the necessary Get/Set methods.

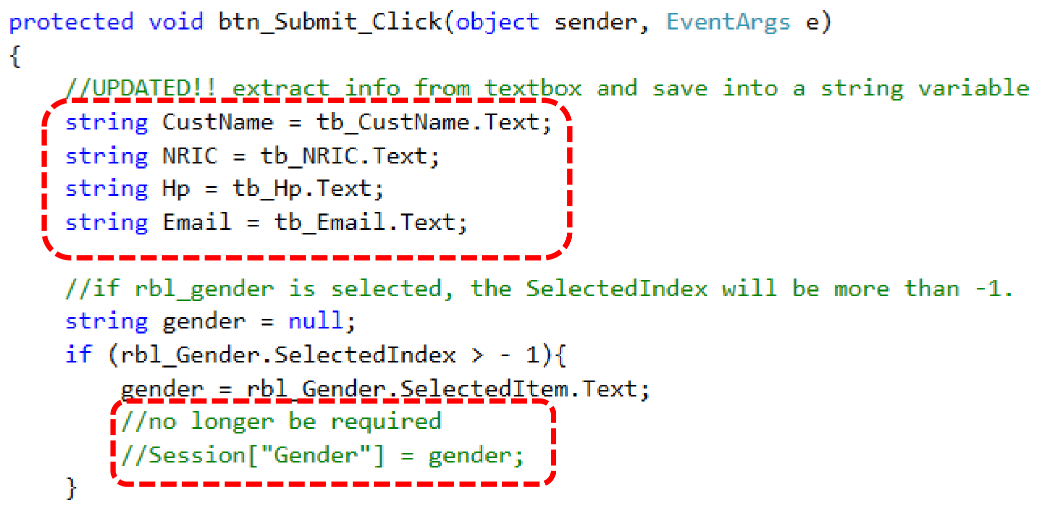


```
//public class properties, accessing from outside the class is possible
2 references
public string CustName
{
    get { return _CustName; }
    set { _CustName = value; }
}
2 references
public string NRIC
{
    get { return _NRIC; }
    set { _NRIC = value; }
}
2 references
public string Hp
{
    get { return _Hp; }
    set { _Hp = value; }
}
2 references
public string Email
{
    get { return _Email; }
    set { _Email = value; }
}
```



```
public string Gender
{
    get { return _Gender; }
    set { _Gender = value; }
}
1 reference
public string Notifications
{
    get { return _Notifications; }
    set { _Notifications = value; }
}
1 reference
public string BirthdayMonth
{
    get { return _BirthdayMonth; }
    set { _BirthdayMonth = value; }
}
```

7. Once the Customer Class is defined, we will now be able to write codes to save the Form values into the Class properties.
8. In **Registration_Form.aspx** code behind, remove all the Session[…] from the previous practical and create an Instance of a Customer Class to save the Form values into its properties. Create a new Session to store the Customer Instance.



```
protected void btn_Submit_Click(object sender, EventArgs e)
{
    //UPDATED!! extract info from textbox and save into a string variable
    string CustName = tb_CustName.Text;
    string NRIC = tb_NRIC.Text;
    string Hp = tb_Hp.Text;
    string Email = tb_Email.Text;

    //if rbl_gender is selected, the SelectedIndex will be more than -1.
    string gender = null;
    if (rbl_Gender.SelectedIndex > - 1){
        gender = rbl_Gender.SelectedItem.Text;
        //no longer be required
        //Session["Gender"] = gender;
    }
    //if cbl_Notifications is selected, the SelectedIndex will be more than -1.
    string notifications = null;
    if (cbl_Notifications.SelectedIndex > -1) {
        if (cbl_Notifications.Items[0].Selected)
        {
            notifications = notifications + cbl_Notifications.Items[0].Text + ";";
        }
        if (cbl_Notifications.Items[1].Selected)
        {
            notifications = notifications + cbl_Notifications.Items[1].Text + ";";
        }
    }

    //no longer be required
    //Session["Notifications"] = notifications;

    string birthdayMonth = null;
    if (ddl_BirthdayMonth.SelectedIndex > -1) {
        birthdayMonth = ddl_BirthdayMonth.SelectedItem.Text;
    }

    //no longer be required
    //Session["BirthdayMonth"] = birthdayMonth;

    //NEW! Create an instance of a new Customer Class. Use the NEW keyword
    Customer cust = new Customer(CustName, NRIC, Hp, Email,gender,notifications,birthdayMonth);

    //NEW! Save Customer object into Session
    Session["Registration"] = cust;

    //Forward your request to ThankYou.aspx for procesing
    Response.Redirect("ThankYou_Form.aspx");

}
```

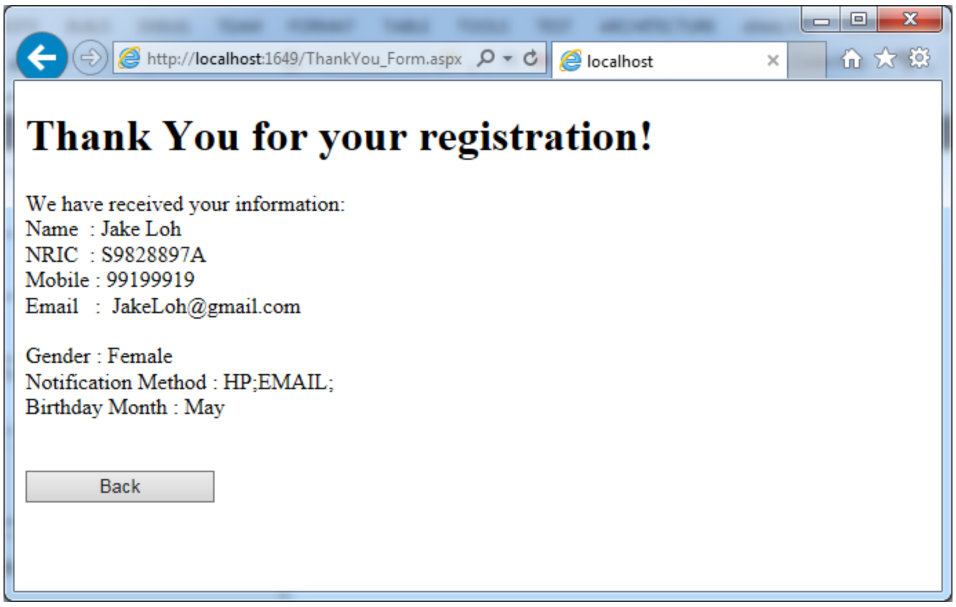
9. In **ThankYou_Form.aspx** code behind, extract the Instance of Customer object from the session. Access the Form values from the Customer Properties (e.g cust.CustName.ToString()).

```
public partial class ThankYou_Form : System.Web.UI.Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        //Create a Customer Instance with default Constructor (with Null fields)
        Customer cust = new Customer();

        //Extract data from Session object "Registration and save it to a new Customer Object.
        cust = (Customer)Session["Registration"];

        //Extract data from CUSTOMER object and display them on the label
        lbl_CustName.Text = cust.CustName.ToString();
        lbl_NRIC.Text = cust.NRIC.ToString();
        lbl_Hp.Text = cust.Hp.ToString();
        lbl_Email.Text = cust.Email.ToString();
        lbl_Gender.Text = cust.Gender.ToString();
        lbl_Notifications.Text = cust.Notifications.ToString();
        lbl_BirthdayMonth.Text = cust.BirthdayMonth.ToString();
    }
    0 references
    protected void btn_Back_Click(object sender, EventArgs e)
    {
        Response.Redirect("Registration_Form.aspx");
    }
}
```

10. Test your application now. The output is based on the input submitted by Registration_Form.aspx.



Thank You for your registration!

We have received your information:
Name : Jake Loh
NRIC : S9828897A
Mobile : 99199919
Email : JakeLoh@gmail.com

Gender : Female
Notification Method : HP;EMAIL;
Birthday Month : May

End of Practical