



Introduction to C#.Net

In this lab you will become familiar with:

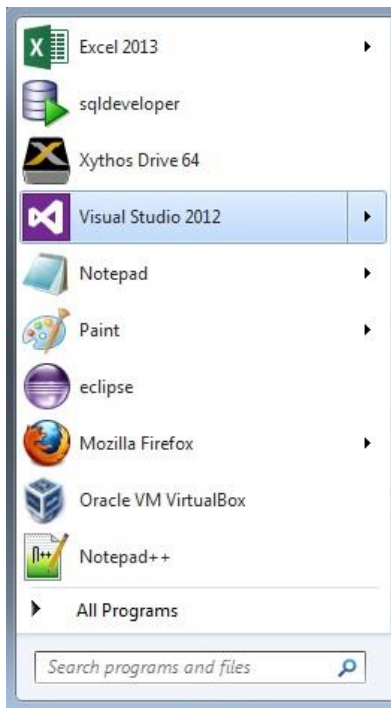
- The .Net Development Environment
- Form Design
- Control Properties
- Object/Form Control Naming Conventions
- Multiple Forms and Form Navigation

Before you start:

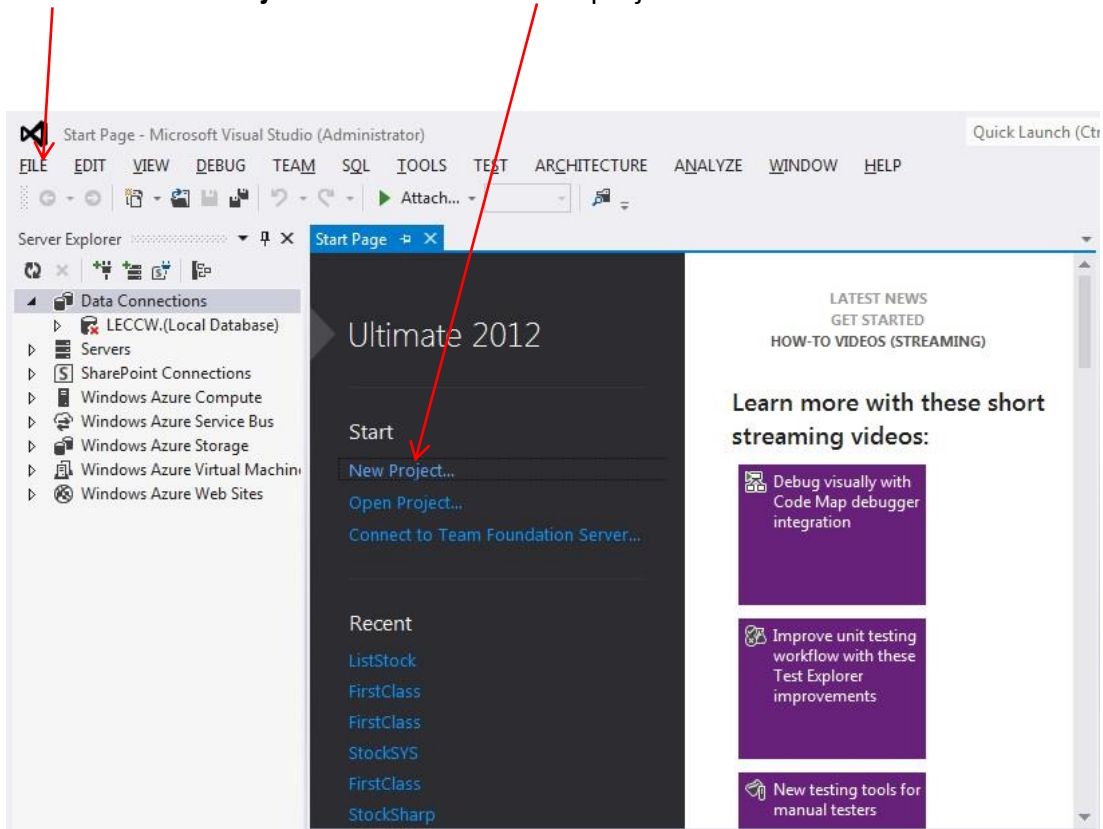
- All material relating to this module (notes, lab sheets, database files, scripts, images etc) should be organised such that you can readily access them.
- Create a folder at **X:\Student\SWE_Project** to store any material used during project support. You may require sub-folders to organise your work.

Using Microsoft Visual Studio 2012

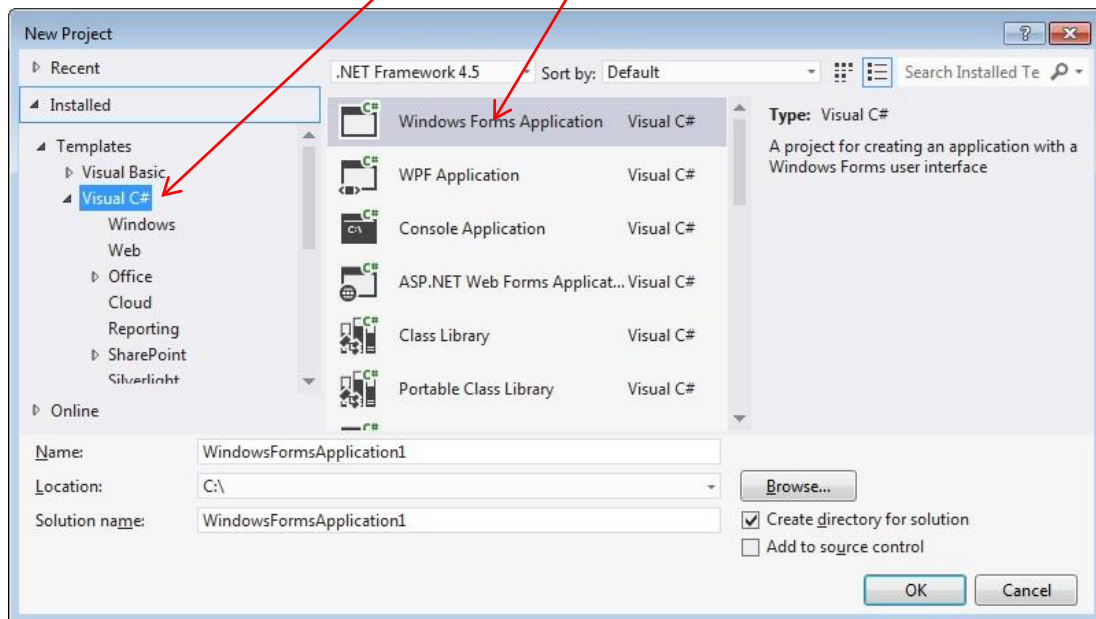
Select Start → All Programs → Microsoft → Microsoft Visual Studio 2012



Select **File** → **New** → **Project** or 'click' on the New project icon.



By default, the selected template is Visual C# / Windows Forms Application.



Before you click 'OK', you must enter the **NAME**, **LOCATION** and **SOLUTION NAME** for your project.

NAME:

This is the name of your C#.Net project.

Use meaningful names. For example StockSYS or LibSYS or CarRental.

LOCATION:

It is possible for two computers to have two different system times. This can cause compiler confusion when the server time appears to be more recent than your development systems (desktop) time. When timestamps are not synchronised you will need to reload your application each time you run it. If these times were synchronised, this is not a problem. Unfortunately, the present installation is such that using the X: drive for .Net development encounters this problem.

To avoid this problem with synchronising time-stamps, create your application on the C: drive. Complete all of your development work on the C: drive. You can later store a copy on your X: drive.

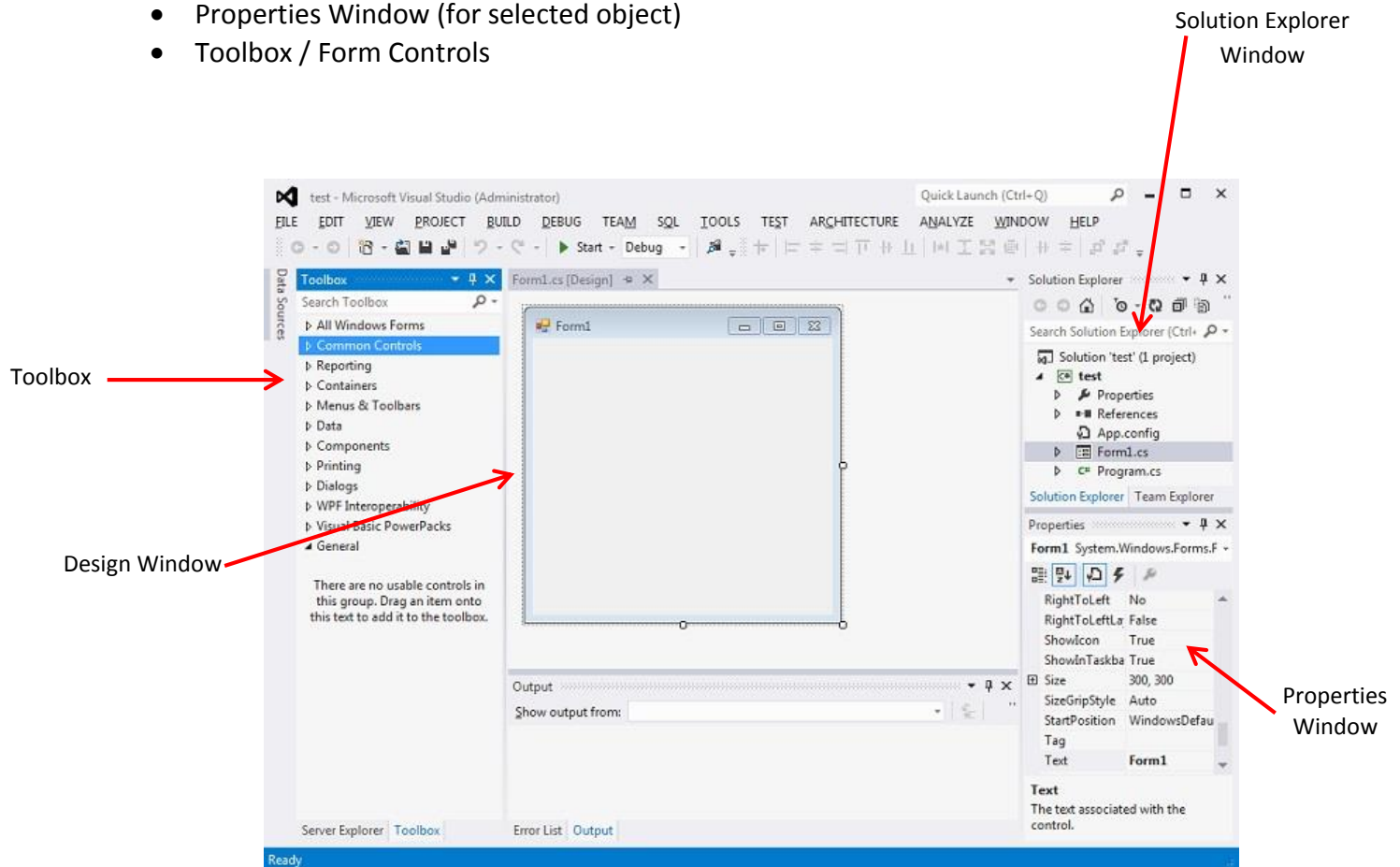
SOLUTION NAME:

By default, this is the same as the project name. You may change this as you wish. Remember, use a meaningful name.

Click 'OK'. You are then presented with a blank form and are ready to design your first form.

Familiarise yourself with the different parts of the form shown above :

- Design Window
- Code Window
- Solution Explorer window
- Properties Window (for selected object)
- Toolbox / Form Controls



Setting Object/Form Control Properties

The properties window lists the properties and their current settings for a **selected** object or form control.

Each object or control has a different set of properties. For example, a **Form** object has a different set of properties than a **Text Box** form control. You will become familiar with these properties when you design your forms.

It is good practice to set some properties as soon as the object is created or a form control is placed on a form. Such properties include:

- Name
- Text
- Size

Setting the **name** property in particular, allows the object or form control to be easily referenced in any code that you might write.

Consider a C#.Net solution with 15 forms. If forms were left with default **name** property (form1.cs, form2.cs,, form15.cs) it would be extremely difficult to identify form functionality when we examine the solution explorer window. Setting meaningful **Name** properties for each form (frmNainMenu.cs, frmStockMenu.cs, frmSales.cs) resolves this problem.

Object/Form Control Naming Conventions

There are many different types of objects and form controls. To distinguish between types of objects and form controls **when examining source code**, we use an accepted naming convention.

There is no de-facto standard. Different conventions have been established within different organisations. Using accepted naming conventions ensures that applications developers can easily identify the types of objects and form controls defined and referenced in source code.

The tables below illustrate the naming conventions which you **MUST** use when developing your .Net applications. Please note that this list is not conclusive – only the more frequently used form controls have been included.

Pre-fixes for form control names

| Control | Prefix | Example |
|------------------------------|--------|-----------------------------|
| Calendar | cal | calArrDate; calDeptDate |
| Check Box | chk | chkMarried; chkWithdraw |
| Combo Box | cbo | cboStock; CboAccTypes |
| Command Button | btn | btnExit; btnGet; btnAmend |
| Data Grid | grd | grdStock; grdMembers |
| Date Time Picker | dtp | dtpAppointment |
| Group Box | grp | grpStock; grpMemberDetails |
| Horizontal Scroll Bar | hsb | hsbSpeed; hsbTemperature |
| Label | lbl | lblSurname; lblDate |
| List Box | lst | lstItems; lstCountyCodes |
| Option Button (radio button) | opt | optSurname; optPrice |
| Picture | pic | picStock |
| Text Box | txt | txtStockNo; txtDescription; |
| Vertical Scroll Bar | vsb | vsbHeight; vsbAttendance |

Table 1.1

Pre-fixes for variable names

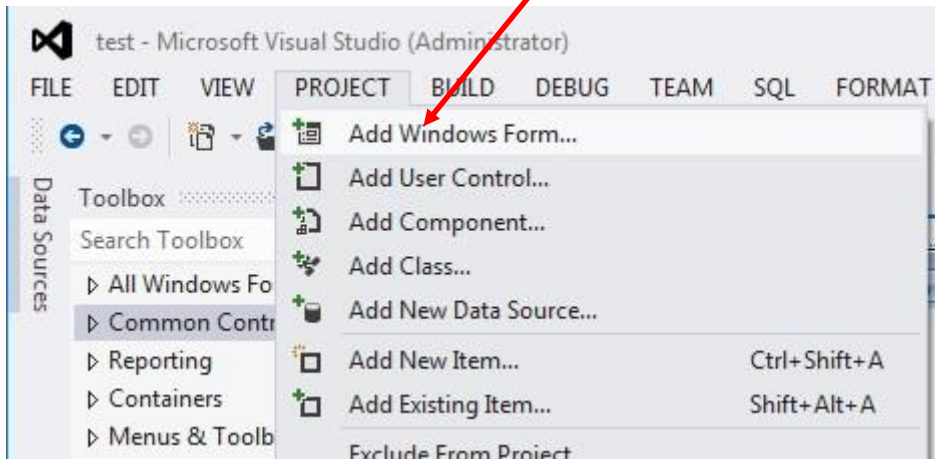
| Data Type | Prefix | Example |
|-----------|--------|---------------------|
| String | str | strSurname; strDate |
| Integer | int | intQty; intNoDays |
| Double | dbl | dblPayRate; |
| Date | dte | dteDateOut |

Table 1.2

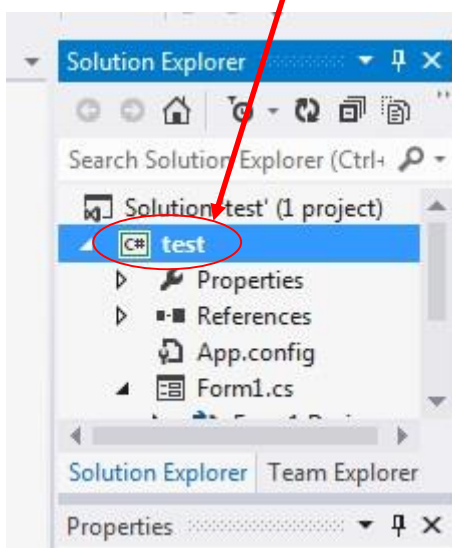
Multiple Forms

A .Net project (solution) may have multiple forms.

To add a form to your solution, select **PROJECT** → **Add Windows Form** or



Alternatively, you can right click on the solution and select Add → New Item → Windows Form Control.



When you click 'Add', you will notice the new item appears in your solution explorer window.

Navigating Between Forms

See sample application CS_01_FirstClass.

Events and Event Handlers

See SWEP_01_Select Event Handler

Startup Form

See SWEP_02_Change Startup Form