Institute of Technology Tralee
Institiúid Teicneolaíochta Trá Lí

Software Engineering

Prototyping using 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:

- Create a folder on your X:\drive to save your software engineering CA components in.
- Create 2 subfolders: ***Design*** and ***Prototype***

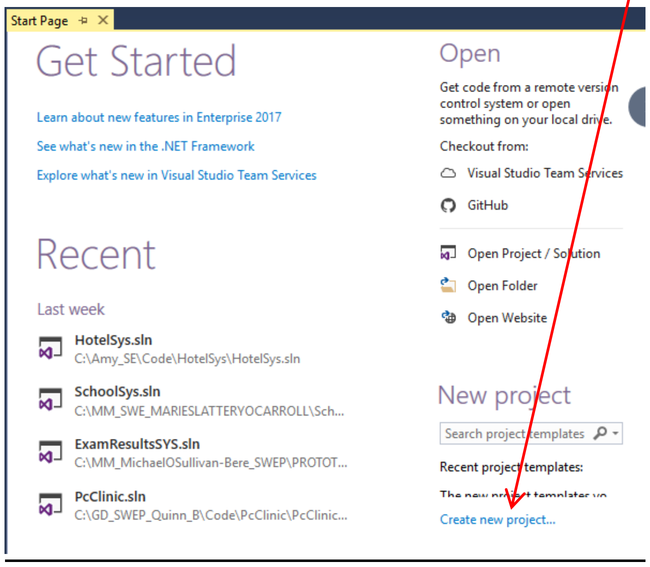
Using Microsoft Visual Studio 2017

Open Visual Studio 2017

Visual Studio 2017

Select Start → Visual Studio 2017
OR
Select Start → All Programs → Microsoft → Visual Studio 2017

Next: Select **File→ New → Project** or 'click' on the ***Create New project*** link.


Start Page

Get Started

Learn about new features in Enterprise 2017
See what's new in the .NET Framework
Explore what's new in Visual Studio Team Services

Recent

Last week

HotelSys.sln
C:\Amy_SE\Code\HotelSys\HotelSys.sln

SchoolSys.sln
C:\MM_SWE_MARIESLATTERYOCARROLL\Sch...

ExamResultsSYS.sln
C:\MM_MichaelOSullivan-Bere_SWEP\PROTOT...

PcClinic.sln
C:\GD_SWEP_Quinn_B\Code\PcClinic\PcClinic...

Open

Get code from a remote version control system or open something on your local drive.

Checkout from:

Visual Studio Team Services
GitHub

Open Project / Solution
Open Folder
Open Website

New project

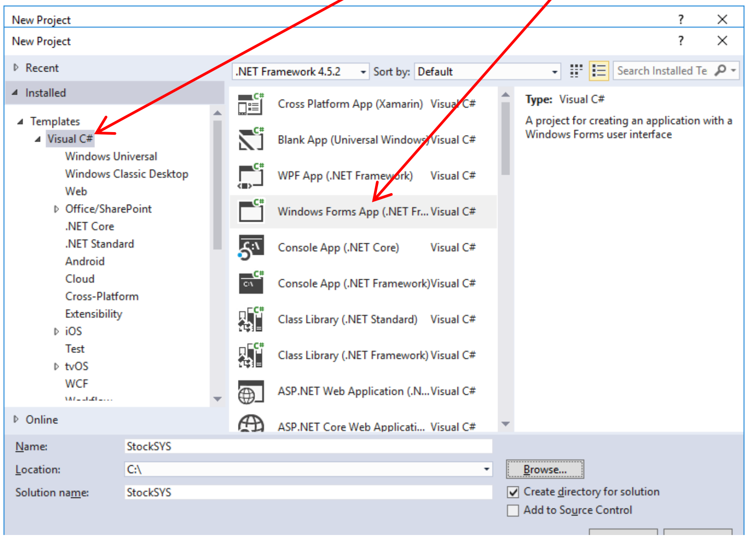
Search project templates

Recent project templates:

The new project templates yo...

Create new project...

Make sure the selected template is **Visual C#** / **Windows Forms Application.**



New Project

New Project

Recent

Installed

Templates

Visual C#

Windows Universal

Windows Classic Desktop

Web

Office/SharePoint

.NET Core

.NET Standard

Android

Cloud

Cross-Platform

Extensibility

iOS

Test

tvOS

WCF

Online

.NET Framework 4.5.2 | Sort by: Default | Search Installed Te

Cross Platform App (Xamarin)	Visual C#
Blank App (Universal Windows)	Visual C#
WPF App (.NET Framework)	Visual C#
Windows Forms App (.NET Fr...	Visual C#
Console App (.NET Core)	Visual C#
Console App (.NET Framework)	Visual C#
Class Library (.NET Standard)	Visual C#
Class Library (.NET Framework)	Visual C#
ASP.NET Web Application (.N...	Visual C#
ASP.NET Core Web Applicati...	Visual C#

Type: Visual C#

A project for creating an application with a Windows Forms user interface

Name: StockSYS

Location: C:\

Solution name: StockSYS

Browse...

[✓] Create directory for solution

[ ] Add to Source Control

<u>**Before you click 'OK'**</u>, 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**

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.

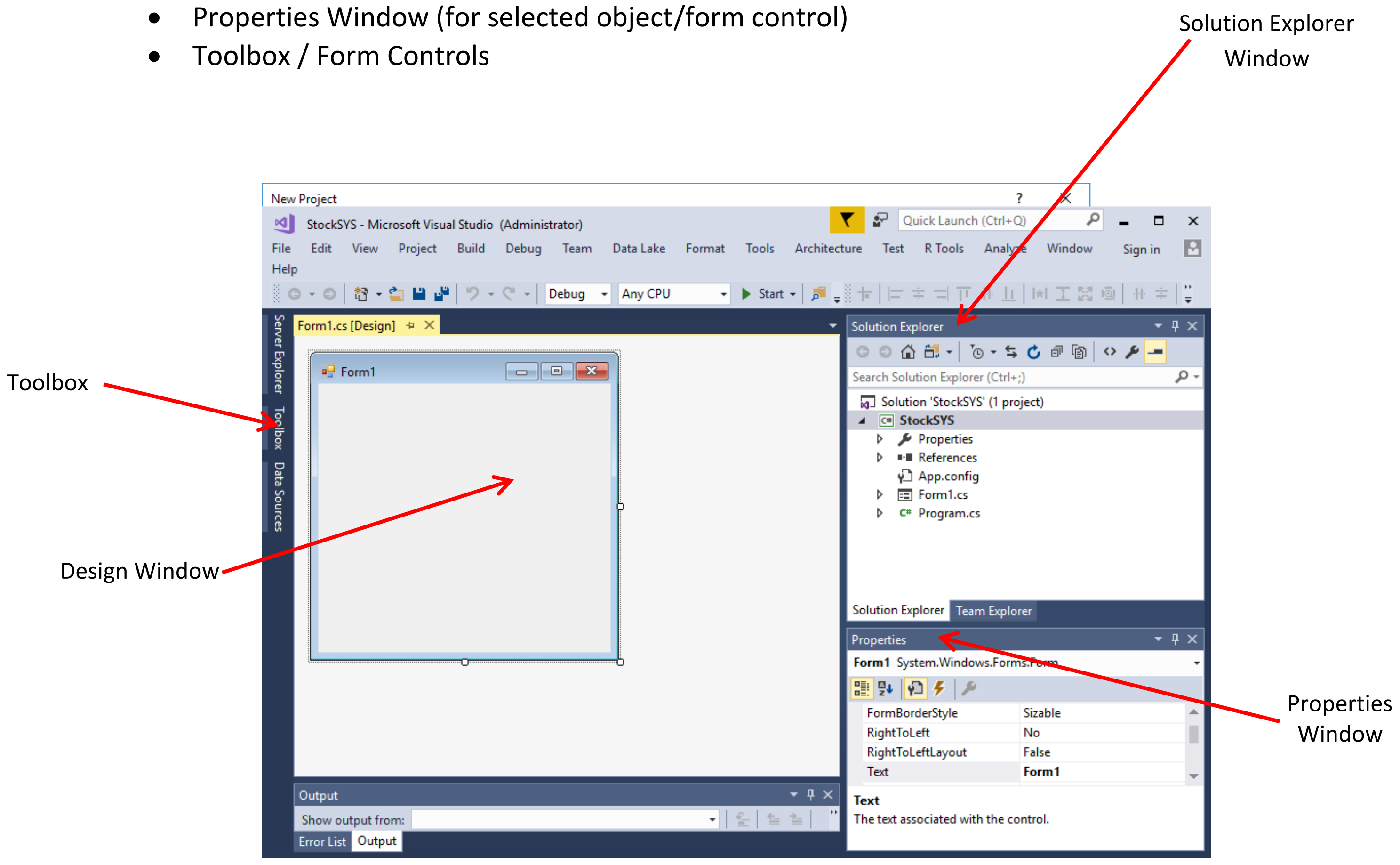
<u>**To avoid this problem**</u> with synchronising time-stamps, create your application on the **C: drive.** Complete all of your development work on the C: drive. At the end of this session, you can store a copy of your C# project in your X: drive folder.

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

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

<span style="color:red">**Next:**</span> Familiarise yourself with the different parts of the form as shown below:

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


Solution Explorer Window
Toolbox
Design Window
Properties Window
New Project
StockSYS - Microsoft Visual Studio (Administrator)
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Data Lake Format Tools Architecture Test R Tools Analyze Window Help
Sign in
Debug
Any CPU
Start
Server Explorer Toolbox Data Sources
Form1.cs [Design]
Form1
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'StockSYS' (1 project)
StockSYS
Properties
References
App.config
Form1.cs
Program.cs
Solution Explorer Team Explorer
Properties
Form1 System.Windows.Forms.Form
Output
Show output from:
Error List Output

FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	Form1

Text
The text associated with the control.

Setting Object/Form Control Properties

The properties window lists the properties and their current settings for the ***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.
Set the properties list in **A→Z order** to easily find a particular property.

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
Form	frm	frmMainMenu
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

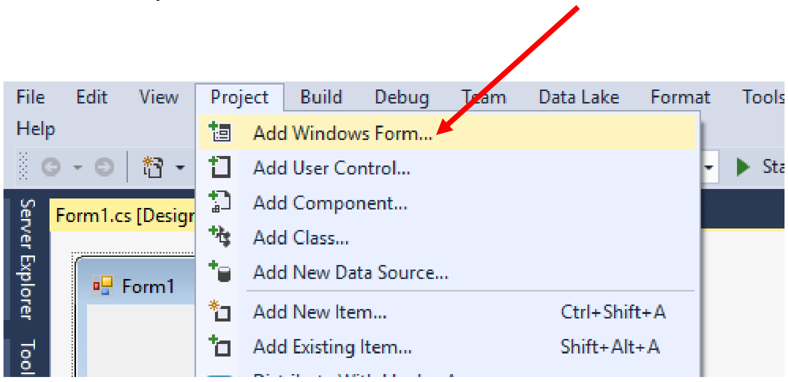
Exercise

1. Build a prototype for the ***StockSYS*** Main Menu application in our StockSYS system.

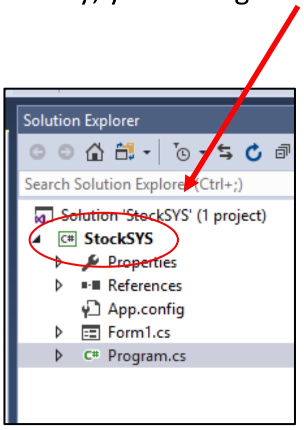
Multiple Forms

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

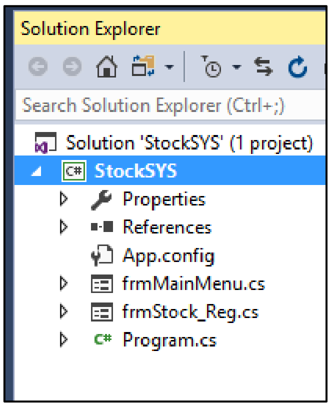
To add a form to your solution, select **PROJECT → Add Windows Form** or



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



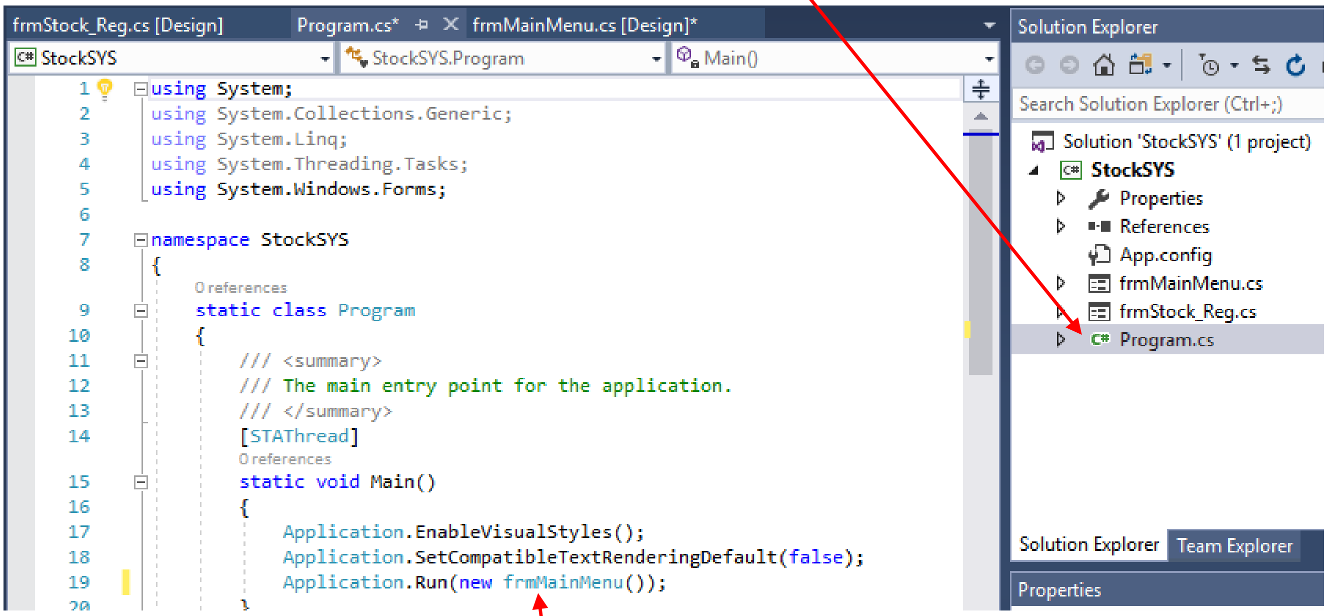
When you click 'Add', you will notice the new item appears in your solution explorer window.



Running the Application

Each application has a ***Start Up*** form. This is the ***first*** form that is executed when the application is executed (e.g. a main menu). The Start Up form can be easily changed.

In the ***Solution Explorer*** window, select the class **Program.cs**.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace StockSYS
{
    0 references
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        0 references
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new frmMainMenu());
        }
```

The Start Up form is specified with the ***Run*** method.

To navigate between forms at run time requires some (two lines) code. We will see this later. For now, simply change the Start Up form when required.

NOTE: While working with C#.Net, you should regularly save your work.

Exercise

2. Build a prototype for the ***Register Stock*** application in our StockSYS system.