# MSc OOP - Programming Windows CE Lab Sessions, 15th October 2008.

The lab sessions will be split into 2 groups the first group will go to the Centre for Learning, Innovation & Collaboration (CLIC, N425) for supervised hands-on device programming at 10am and the second will carry out the unsupervised programming exercises in the normal lab. At 2pm the second group will attend CLIC (N425) for the supervised hands-on device programming, and the first group will carry out the unsupervised programming exercises in the normal lab.

| Group 1 | Group 2 |
|---------|---------|
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |

# MSc OOP - Programming Windows CE Lab Sessions, 15th October 2008.

Please note that documentation on the .Net Compact Framework is available at the Microsoft website: <u>http://msdn.microsoft.com/en-us/library/f44bbwa1.aspx</u> . Please ensure that you are able to access that website and have it available for reference during your exercises.

Note: Those with access to real devices, please repeat the exercise with the devices provided.

# Exercise 1 – Verify that your development environment is set up and the Windows CE emulator is ready.

- Start Visual Studio.
- Create a new project.
- Select Smart device, Pocket PC 2003, and then Device Application.
- Change the name of the project to Ex1.
- Expand the toolbox and drag a button and a textbox on to the form.
- Double-click the button and the code editor will appear.
- Add the line: textBox1.Text = "Hello!"; directly where the cursor appears in the code editor.
- Click "Debug" and then "Start Debugging" on the menu bar.
- Select Pocket PC 2003 Emulator and then click "Deploy". (If you are using a real device, select "Pocket PC" and deploy);
- Wait till your application appears in the emulator, and then click the button. Verify that the textbox changes to "Hello!".
- Verify that you are able to debug your program by inserting a breakpoint in the editor. Click on the line that you just added in the code, and then click "Debug" and "Toggle Breakpoint" on the menu.
- Go back to the emulator and click the button. Verify that your editor highlights the line in the code and execution of your program has indeed paused.
- Click "Debug" and "Step Into" on the menu, and repeat until the emulator appears and then click the button in your application again to verify that you have successfully stepped through the code in your program.
- Click "Debug" and then "Stop Debugging" on the menu.
- On the emulator, click "File" then "Save state and exit".

This is the end of Exercise 1.

Exercise 2 – By using a textbox (or label) and buttons, create a simple calculator with add, subtract, multiply, divide and clear functions.

## Exercise 3 – Key events

Create a new Pocket PC 2003 project. Call it Ex3

- If the "Properties" window is not visible, display it by clicking on "View" then "Properties Window" on the menu.
- On the properties windows, click the "Events" button (lightning bolt icon).
- Drag new textbox onto the form.
- Make sure that the textbox is selected, and then on the properties windows, double-click "Keydown" event.
- Insert the following code in the code window in the event handler "textBox1 KeyDown(object sender, KeyEventArgs e)":

```
if ((e.KeyCode == System.Windows.Forms.Keys.Up))
{
   textBox1.Text = Convert.ToString((Convert.ToInt16(e.KeyCode)));
}
```

- Run the program and click the cursor up button on the device emulator (or device).
- Verify that the text in the textbox changes according.

## Exercise 4 – Extend the program in Exercise 3 to find out the keycodes of all hardware button are.

Exercise 5 – Write a program to move a "panel" control around the form using the hardware cursor keys.

### Exercise 6 – Hardware "soft" keys.

• Look up the documentation for the class "Microsoft.WindowsCE.Forms.HardwareButton" on the Microsoft .Net Compact Framework documentation website and write a program to detect presses on these button.

### Exercise 7 – Battery Level.

• Look up the class "SystemProperty.PowerBatteryStrength" in the documentation and write a program to display the battery level.

### **Exercise 8 – Network Connection**

Knowing when your device is connected to the network is important if your application needs to interact with a server.

The following code fragment checks if the system is connected to a network. It also shows how to call into the native Win32 API.

```
using System.Runtime.InteropServices;
public class InternetCS
{
    [DllImport("wininet.dll")]
    private extern static bool InternetGetConnectedState(out int Description, int
ReservedValue);
    public static bool IsConnectedToInternet()
    {
        int Desc;
        return InternetGetConnectedState(out Desc, 0);
    }
}
```

The class can be used as follows:

bool connected = InternetCS.IsConnectedToInternet();

Using the above code, write a program to show whether your device has a connection to the network.