

LAB 1: A clock display

Getting started

In this lab you will examine and make modifications to a clock program which counts hours and minutes and displays the elapsed time.

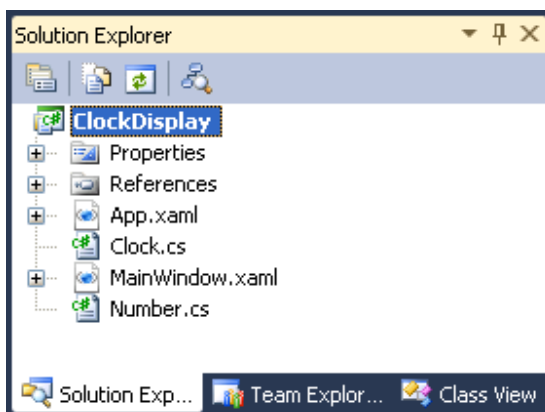
The program uses **C# classes** to **model the information** (the time) and **logic** (how the time is updated) of the clock. It also has a simple graphical user interface (GUI) to display the clock information and to allow the user to interact with the clock (start it and stop it).

In this lab you will look only at the code for the C# model classes, and use the GUI simply to test the program. You will look at creating a GUI later in the module.

Task 1 : Examining the program

Open the program:

1. Start a VM which has Visual Studio 2010 installed, and start Visual Studio.
2. Download the file ClockDisplay.zip from Blackboard, and extract the contents.
3. In Visual Studio, select **File>Open Project..** and browse to find the file *ClockDisplay.sln* inside the *ClockDisplay* folder, then click **Open**. A *.sln* file is a Visual Studio solution file. Once the solution opens, you should see the following files in the **Solution Explorer** window:



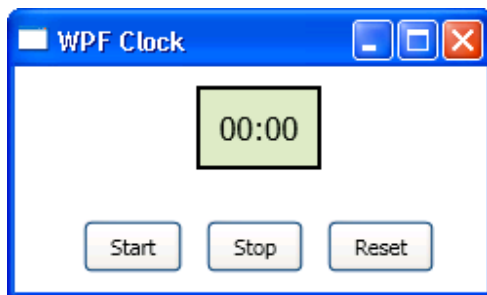
The files *Clock.cs* and *Number.cs* contain the C# classes. The files *App.xaml* and *MainWindow.xaml* define the user interface.

Test the program:

1. Select **Build>Build ClockDisplay**. This will compile all the code in the project. If there are syntax errors in the code these will be reported in the **Errors** window – there shouldn't be any errors, so you should see a message like this in the **Output** window:

```
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
```

2. You can now run the program. Select **Debug>Start Debugging** (or simply press the **F5** key). The program should start up and you should see its GUI window:



3. Try the buttons – Start and Stop should work, Reset doesn't do anything yet. Note that the clock doesn't work in real time – it runs much faster so that you don't have to wait for a whole hour just to see that the hours display works.

Observe the display as the program runs. What happens at the end of 60 minutes? What happens at the end of 24 hours?

Read the code:

1. Open the files *Clock.cs* and *Number.cs*.
2. Spend some time carefully reading the C# code in these files. Look for clues as to how the program works. Think about the information and logic defined in the C# classes. There are some comments in the code to help you.
3. Go to Blackboard and navigate to the **Assignments** section for this module. You should see a test named **lab_1**. This is part of the lab exercise, and does not count towards your module mark.
4. Start the test and answer the questions in it. These questions will help you to find out how well you have understood the code you have read. You should refer back to the C# code to help answer the questions, and you may want to run the program again also.
5. Sketch a class diagram and an object diagram for the program.

Task 2 : Modifying the program

You will now make some modifications to the code to make the program behave slightly differently. Having read the code should help you decide what parts you will need to change.

As you change code you will very probably get some syntax errors (these are reported as you build the program, or may be underlined in red as you work with the code). Try to be systematic in fixing these errors – reading the error messages and use these to help you identify what is wrong with your code. Ask for help if you need it. Don't get frustrated – identifying and fixing errors is a normal part of the process of programming!

Modification 1

Change the program so that it displays a **12 hour clock** instead of a 24 hour clock. You don't need to worry about displaying AM and PM, simply make the clock count up to 11:59 instead of 23:59. Test your modification.

Modification 2

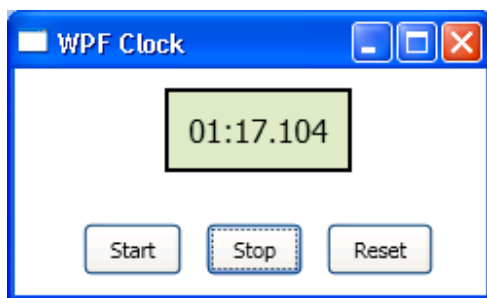
The interface has a **Reset button** which doesn't work. It doesn't work because the Reset method in the Clock class has no code in it. Add code so that the clock display is reset to 00:00. Test your modification by clicking the Reset button while the program is running. This will call your Reset method.

Modification 3

Change the program so that it displays **hours, minutes and seconds**. Test your modification.

Modification 4

Change the program so that it displays **minutes, seconds and thousands of a second**, as shown below. This is a bit more challenging, and there are several ways of doing it! You don't need to modify the GUI at all. Test your modification.



Note that you might want to speed up the timing a bit when testing this version. You can do this by finding the following line in *MainWindow.xaml.cs*, and changing the value 100 to 1 or 0:

```
timer.Interval = TimeSpan.FromMilliseconds(100);
```

