

## Practical Exercise

### Getting started

In this exercise you will review and make modifications to the digital stop clock program you saw in the workshop.

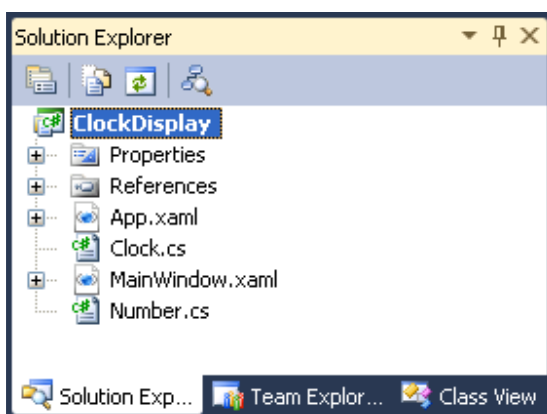
The program uses **C# classes** to **model the information** (the time) and **logic** (how the time is updated) of the clock. It also has a simple graphical user interface (GUI) to display the clock information and to allow the user to interact with the clock (start it and stop it). The GUI class also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

In this exercise you will look only at the code for the C# classes which implement the clock component, and use the GUI simply to test the program.

### Task 1: Reviewing the program

#### Open the program:

1. Start Microsoft Visual Studio.
2. Download the file ClockDisplay.zip using the link [www.paterson.co.uk/gcal/lahti/ClockDisplay.zip](http://www.paterson.co.uk/gcal/lahti/ClockDisplay.zip), 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:



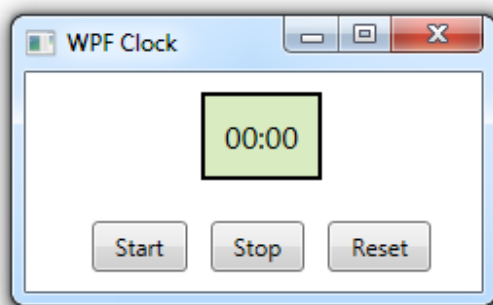
The files *Clock.cs* and *Number.cs* contain the C# classes. The files *App.xaml* and *MainWindow.xaml* define the GUI.

**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. Think about the information and logic defined in the C# classes. There are some comments in the code to help you.
3. Answer the following questions. You should refer to the program code, the slides and your observation of the program as it runs

**Question 1** What are the names of the C# classes in the ClockDisplay program?

**Question 2** List the instance variables of Number. and their types

**Question 3** List the instance variables of Clock, and their types

**Question 4** List the methods of Number

**Question 5** List the methods of Clock

**Question 6** How many digits can a Number object display?

**Question 7** How is the initial time value of the clock set?

**Question 8** What is the value of Limit for a Number object which represents hours?

**Question 9** How many objects exist once a Clock has been created?

**Question 10** How does the clock set up its Number components to display hours and minutes

**Question 11** What is the purpose of the modulus, or remainder, operator (%) in the Increment method of Number?

**Question 12** What is the purpose of the *if* statement in the GetDisplayValue method of Number?

**Question 13** What message needs to be sent to a clock object to make it move on by one minute?

**Question 14** What messages does the clock object send to other objects?

**Question 15** What is the purpose of the *if* statement in the TimeTick method of Clock?

## 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 by Visual Studio 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

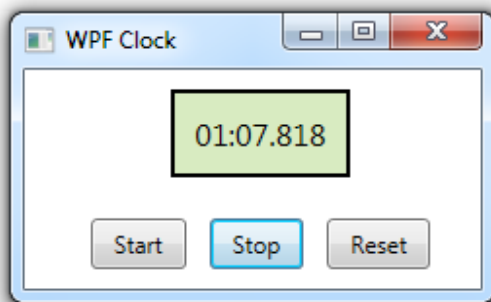
The interface has a **Reset button** which doesn't work. It doesn't work because the Reset method in the Clock class has no code in it. Add code so that the clock display is reset to 00:00. Test your modification by clicking the Reset button while the program is running. The GUI code has been written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

