MALMÖ HÖGSKOLA
Centrum för teknikstudier
School of Technology

Programmering med C#, grundkurs
Programming Using C#, Basic Course

Optional Exercise
The Clock Program

Do not submit!

Farid Naisan
University Lecturer
Department of Computer Sciences

The Clock Program

This exercise is intended to give training in developing a console application made of at least two classes. We will accomplish this task in several steps. Although you may certainly use Visual Studio to solve this exercise, the description here shows how a C# program can be developed using a simple editor such as Windows NotePad, compiled and run from the Command Line.

1. Version 1

Write a console application that shows the current date and time. Write a class ClockV1 with a Main method and let the current time be saved in a local variable inside the Main method. Writing an application with only one class is not a good practice but we will optimize this in the later versions.

The current date and time can be obtained from a .NET structure named **DateTime**. This structure has a member called **Now** that stores ALL date and time details for the very current moment. An object of **DateTime** needs not be instantiated with **new** as it is a value type (like **int** or **double**). This will be explained later on.

```
static void Main()
{
  //DateTime has everything about the current system time and
  //is a part of the namespacen System just as Console.

  DateTime time;
  time = DateTime.Now;  //Get the time (as an object)
  . . .
```

Compile and run the program.

2. Version 2

Change Version 1 and let an instance variable save the current date and time. In other words, make the local variable mentioned above as an instance variable (field or attribute). Remember that the attribute "time" MUST be declared static; otherwise it will not be accessed by static methods such as the Main. Compile and run the program.

3. Version 3

Follow version 2 but let the job be done by a separate method in the same class.

```
static void PrintDateAndTime()
```

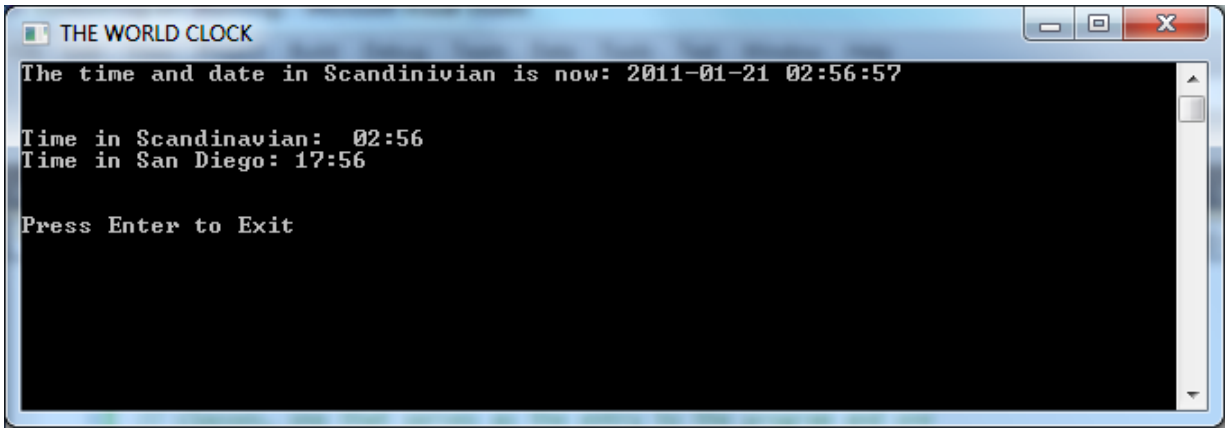
Again, the method must be declared static so it can make use of the attribute "time". This solution is still not a good practice but we will optimize it in the next version. Compile and run the program.

4. Version 4

Finally, we get to the point. The application should contain at least two classes, one that serves as the entry to the program and one that performs the actual job. This a good practice of course to keep in mind in the later projects, **to write a class for every object and the let the object take care of as much of its stuff as it can**. Let for example a House object paint itself, crack Then you can write a class that starts the program. Please also note that by using a separate class, you do not need to have members, attributes as well as methods, as static!.

Compile and run the program.

Then write a method that calculates the time somewhere else by adding a positive or negative number of hours, for example if you are in Scandinavian, the time in San Diego will be -9 hours.



5. Hints

The easiest way to compile from the command line is to open the Visual Studio.NET Command Prompt. This console window is specially adapted to the .NET SDK. You don't have to worry about the environment settings. The Command Prompt is installed with the VS.NET under Visual Studio.NET tools as shown below.

To compile the two cs-files and build into exe program, you can write:

```
csc /out:clock.exe clockv4.cs clockv4prog.cs
```

or:

```
csc clockv4.cs clockv4Prog.cs
```

In both cases, clockv4Prog.exe is built provided no compilation error made. If you have more files that you would like to compile into a program, you can simply write as follows:

```
csc /out:clock.exe *.cs
```

or just:

```
csc *.cs
```

Make sure you do not have irrelevant cs files in the directory.

To run the program you simply write at prompt the name of the exe-file and press Enter.

clockv4prog.exe

6. Some Command Line Commands

Press Enter after every command:

CLS	clears the screen
CD folder name	changes directory to folder name.
CD..	(CD with 2 dots) changes directory one step higher
**CD **	changes to root directory
D:	changes to drive D:
CSC	C# compiler, lists all attributes that can be used with the C# compiler.

7. Links

Compilation :
http://msdn.microsoft.com/en-us/library/2fdbz5xd(v=VS.100).aspx

God Luck!

Programming is fun. Don't give up. Ask for help!

Farid Naisan
Course Responsible and Instructor