Virginia Tech ■ ECE/CS 4570: Wireless Networks and Mobile Systems ■ Spring 2005

In-class Laboratory Exercise 3 (L03)

Part I – Objectives and Laboratory Materials

Objectives:

The objectives of this laboratory are to:

- Introduce the C# programming language
- Introduce the .NET compact framework development environment

After completing the assignment, you should be able to:

- Write a *simple* mobile C# application
- Develop and deploy mobile applications to mobile devices using the Visual Studio environment

Hardware to be used in this laboratory assignment:

- Dell Latitude C640 notebook
- iPAQ 3850 with cradle

Software to be used in this laboratory assignment:

- Windows 2000, Microsoft ActiveSync, Microsoft Visual Studio 2003, Microsoft .NET Compact Framework, and Microsoft Visual C# .NET on the notebook computer.

Part II – Pre-laboratory Assignment

This portion of the assignment should be completed *prior* to the in-class lab session.

There are two goals for this pre-laboratory assignment. The first is to get a glimpse of the C# language. All future mobile applications in this class will be designed using C# and the .NETcf, so this is the time to get acquainted with them. The second goal is to learn about the .NETcf and what it does. Together, C# and .NETcf allow fast and easy development of mobile applications.

Reading:

- The simple .NET tutorial (provided at the class Blackboard site)
- Additional C# and .NET tutorials, as described below

C# Tutorials

Listed below are three tutorials that give a good introduction to the C# language and how to use it. It is recommended that you test and experiment with the sample code in the tutorials. To build C# applications create a new project and choose the "Console Application" template from the "Visual C# Projects" section. This template will allow you to use normal console input and output functions, saving you from any graphical programming. The "useful sections" are sections that should be examined, while the "optional sections" are suggested for your personal enrichment. Note that C# is a fully functional language that can be used for normal programming as well as mobile application development.

- Introduction to the C# language
    - Softsteel Solutions C# tutorial – http://www.softsteel.co.uk/tutorials/cSharp/cIndex.html
        - Useful sections: 1-4, 6-13
        - Optional sections: 17

- C# Station Tutorial – http://www.csharp-station.com/Tutorial.aspx
  - Useful sections: 1-5, 7
  - Optional sections: 6,8
- FunctionX C# Tutorial – http://www.functionx.com/csharp
  - Useful sections: 1-10

Numerous books provide an in-depth study of C#. A recommendation is *Programming C#*, 3rd edition, by Jesse Liberty from O'Reilly publishers.

.NET Tutorials

Listed below are two sources for tutorials on .NET. Please review these.

- Introduction to the .NET Compact Framework
  - Fundamentals of Microsoft .NET Compact Framework Development – http://www.msdn.microsoft.com/vstudio/default.aspx?pull=/library/en-us/dnnetcomp/html/net_vs_netcf.asp
- .NET Compact Framework reference
  - .NET CF Class reference list. Shows what members and attributes are available in the .NET CF and the full framework. Does not include documentation of them, but will show you which ones are supported – http://download.microsoft.com/download/4/d/f/4dfc7768-5017-40a4-8ddb-577720a32e5d/net_compact_framework.chm

Several books provide an in-depth study of the .NET Compact Framework. The one used in developing this course is *Microsoft .NET Compact Framework* by Wigley and Wheelwright from Microsoft Press.

Part III – In-class Lab Assignment

1. Complete the tutorial on mobile application development (provided at the class Blackboard site). At the end of the tutorial, you will have created and deployed a simple mobile application.
2. Using what you learn from the tutorial, create a tip calculator for the iPAQ. The application must:
   a. Accept a decimal amount for the bill (no '$')
   b. Calculate the amount for the tip and the total amount (bill + tip) and display both (that way you don't have to do any pesky addition when paying with a credit card), upon the click of a button
   c. Include tip levels of 15, 17.5 and 20 percent
   d. Resemble the application shown in Figure 1.
3. Once you have the calculator working correctly. Modify it to show the tip amount and total in correct dollar format including the '$', cents and a 0 if the total is less than $1 (for example, $0.25 instead of $.25).
4. Upon a click of the calculate button, also change the bill amount to be in the correct dollar format.

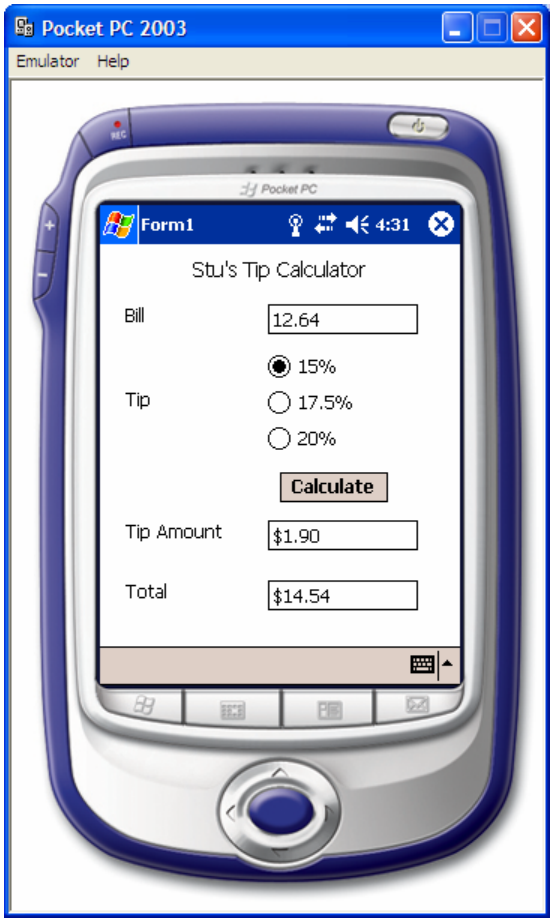

Figure 1. Example user interface for the tip calculator application.