

CSE445/598 Assignment/Project 1 (100 Points)

Summer 2016

Due: Saturday, May 28, 2016 by 11:59pm (Arizona Time)

Introduction

The aim of this assignment is to make sure that you understood the concepts covered in the lectures and in the text, including SOA, SOC, SOD, and their applications in software development. You will also follow a tutorial to obtain hands on experience of developing a simple service and a simple application that uses services. By the end of the assignment, you should have applied these concepts in developing simple services and simple applications that uses remote Web services.

Section 1 Practice Exercises (No submission required)

Reading: Textbook chapter 1.

No submission is required for this section of exercises. However, doing these exercises can help you better understand the concepts and thus help you in writing quizzes and exams.

1. Answer the multiple choice questions in Textbook section 1.7. Compare your answers with the answers given the course web page in the folder “Test and Exam Info”. Doing these exercises will help you prepare your weekly **lecture exercises** and biweekly **quizzes**, as scheduled in the course calendar.
2. What are SOA, SOC, SOD, SOE, SOI, and SOSE? Briefly state their definitions based on your understanding.
3. What are the main differences between requirement analyses in the OOC paradigm and in the SOC paradigm?
4. What are the major benefits of separating an application builder from the service providers?
5. What are the main techniques in SOSE (service oriented system engineering)? For each technique, write one or two sentences to describe its purpose.
6. Compare and contrast the traditional software development process and the Service-oriented software development process. For each step of the development, write a paragraph to describe the purposes, responsibilities, functions of the step.
7. What is a service registry? What is a service repository? What are their differences?
8. An electronic travel agency needs to be developed. What is your responsibility if you are:
 - 8.1 a service provider?

- 8.2 a service broker?
- 8.3 an application builder?
9. You plan to invent a unique online game.
 - 8.1 Describe what you must do as an application builder and what you can expect the service providers to do for you.
 - 8.2 Describe your invention idea and list everything you must do as an application builder.
 - 8.3 List everything that you can possibly find through service brokers.
10. List a few application areas where you believe SOC is a better fit than OOC. State your reasons and justifications.
11. What are the impacts of SOC paradigm to the IT market and to computer science graduates?

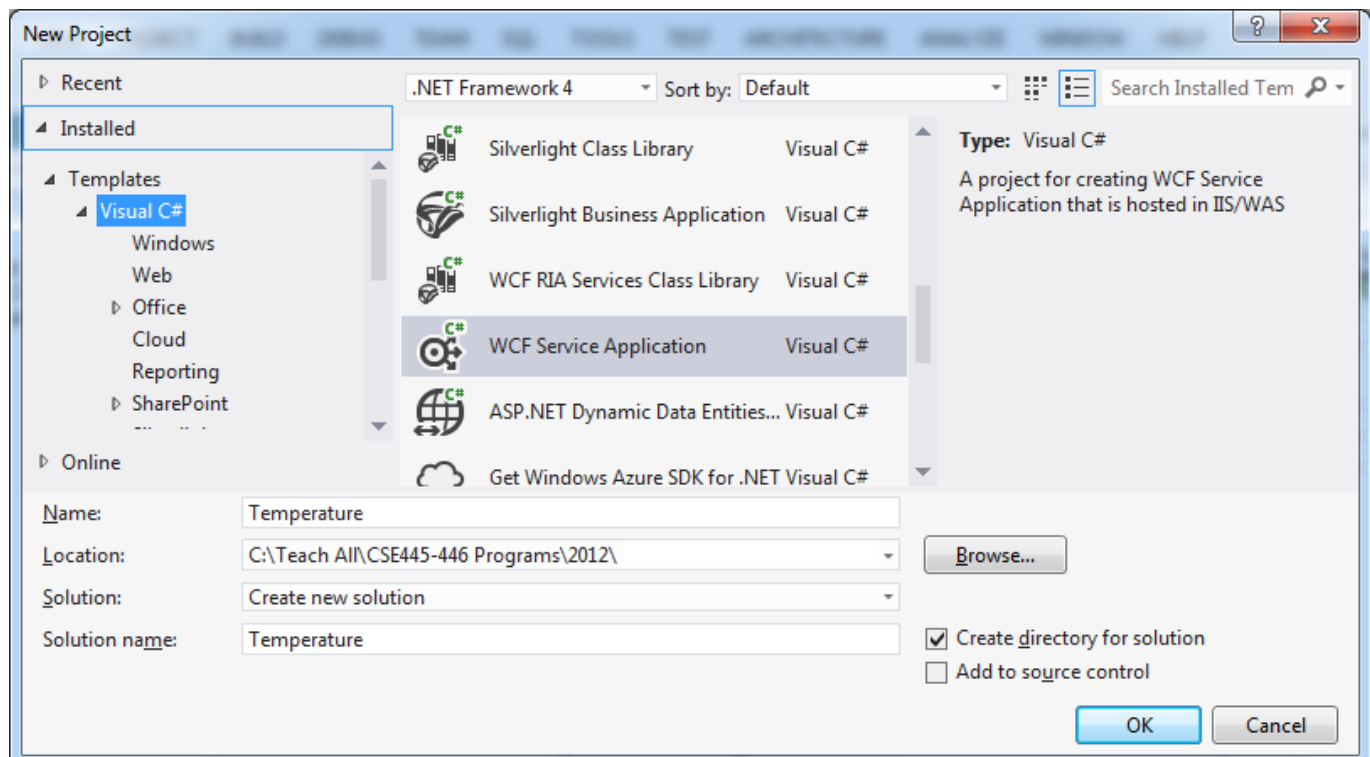
Section 2 Tutorials: Using a Web Service in Your Application

These tutorials help you to complete the assignment questions in Section 3. If the services are not available, use another given in text appendix C instead.

Tutorial 1: Creating a simple Web service

This tutorial shows how a service provider develops services for the application builders' use.

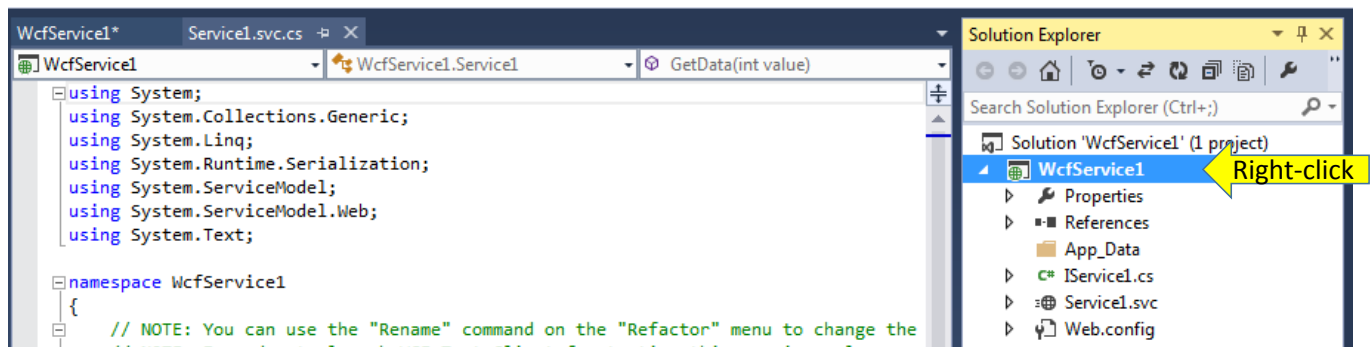
Step 1: In Visual Studio's File menu, choose New → Project..., and then choose "WCF Service Application" Template. This step will allow you to create a new Web service.



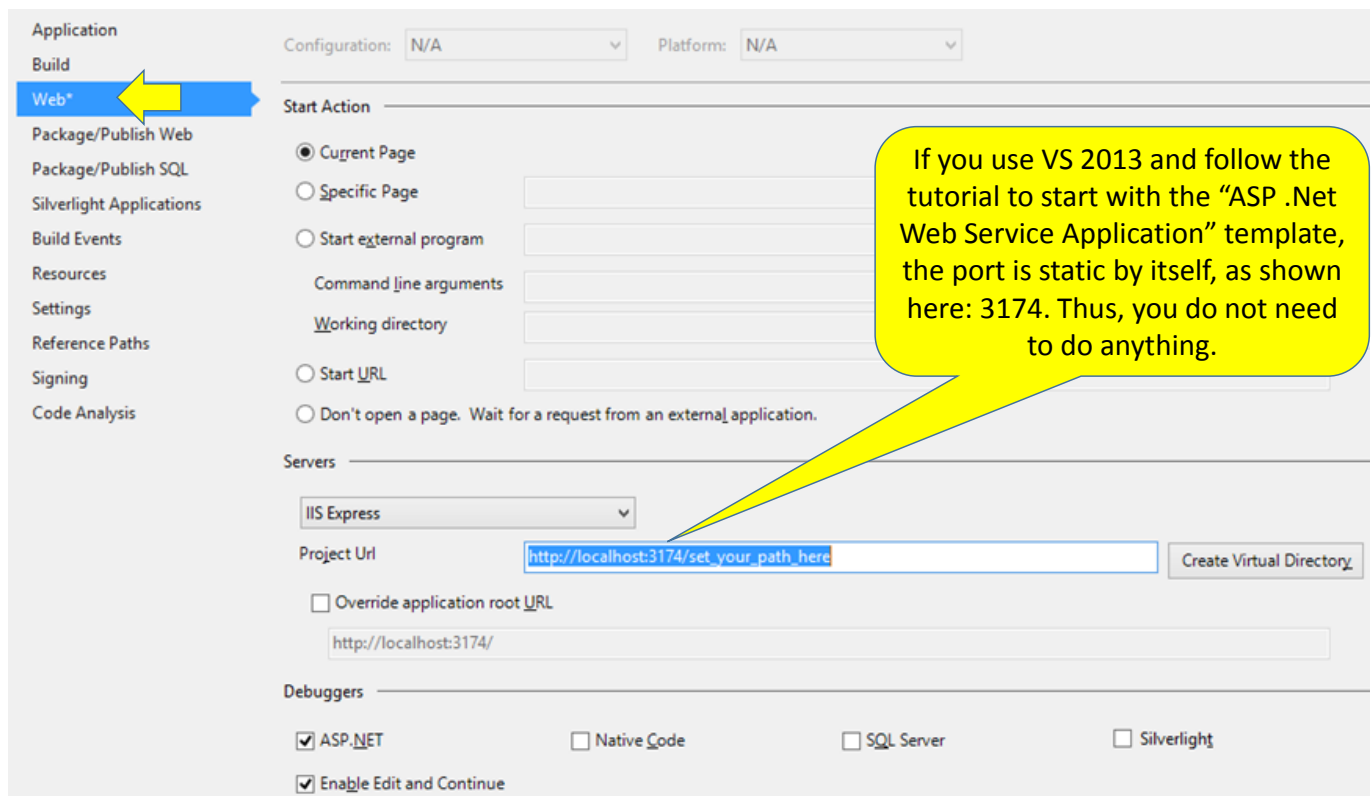
Step 2: Follow textbook, Chapter 3, Section 3.2.1, to develop a simple service using Visual Studio.

Step 3: Setting up the service port number.

If you are using [Visual Studio 2013](#), right-click the Project folder in the Solution stack:



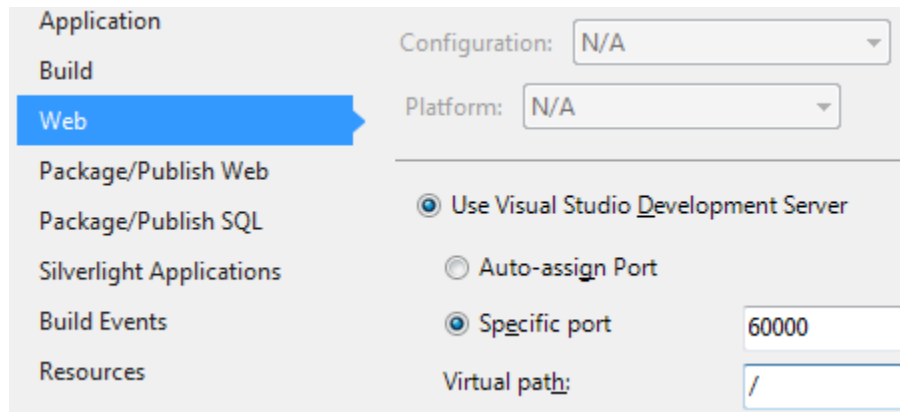
In the Properties window, choose Web*, as shown in the figure below, the IIS Express server will choose a static port number by default.



If you create a Web application project in [Visual Studio 2012](#) that uses the Visual Studio Development Server, a local host port number is selected for the project randomly. This will cause a problem when you send the project to another computer, as a different port number will be generated, resulting in an incorrect URL of the service. To avoid the problem, you can specify a static port number as follows:

1. In Solution Explorer, right-click the project name and choose the “Properties” pane;
2. A new window will open, as shown in the figure below. Choose the “Web” tag and then choose “Use Visual Studio Development Server” and choose “Specific port”.
3. Choose Enter a new five-digit number for the Port number for the localhost. Note, if the port number is not valid, an error message will occur.

4. To verify the change, press CTRL+F5 (Debug → Start Without Debugging) to run the project. The new port number should appear in the address bar of the browser.



Tutorial 2: Using WCF Test Client

Follow the text Chapter 3, Section 3.2.2 to test your service using WCF test client

Tutorial 3: Creating a Windows Forms application to consume the service that you developed in tutorial 1.

Follow the textbook, Appendix A.1 and Chapter 3 Section 3.6.2 to develop a Windows Forms application. This tutorial shows you how an application builder makes use of remote services to create an application that provides a GUI for accessing Web services.

Notice that, in order to test your application, you **must** have the service started first to make the object an active object! You can start the service by right-clicking the file **Service.svc** in the project and choose “View in Browser.” Then, you will see the service URL in the browser address bar. Use this URL when you chose “Add Service Reference...”.

Tutorial 4: Creating a Web Site application to consume the service that you developed in tutorial 1. In Visual Studio, Choose File → New Project. Then, choose C# and ASP .Net Web Site Application. A project with a Default.aspx page will be created, among other files.

Next, you can follow textbook, Section 3.6.3 to develop a Web Site application.

This tutorial shows you how an application builder makes use of remote services to create an application that provides a Web GUI for accessing Web services.

Tutorial 5: Creating an image verifier using the image service

Follow the tutorial given in the textbook, Appendix sections A.3 to develop a Web application that verifies if a human user is entering a Web form. An image verifier is frequently used for preventing programmed attacks to a Web site that allows self-registration.

Section 3 Web Services (Submission required, 100 points)

This section of the assignment is an **individual assignment**. Each student must perform and submit **independent** work.

1. Follow the Tutorial 1 given in Section 2 to develop a temperature convention Web service. The service contains two operations: [10 points]

```
int c2f(int c); // convert Celsius temperature to Fahrenheit temperature
```

```
int f2c(int f); // convert Fahrenheit temperature to Celsius temperature
```

Static service address must be used in order for the TA to test your service and application on a different computer.

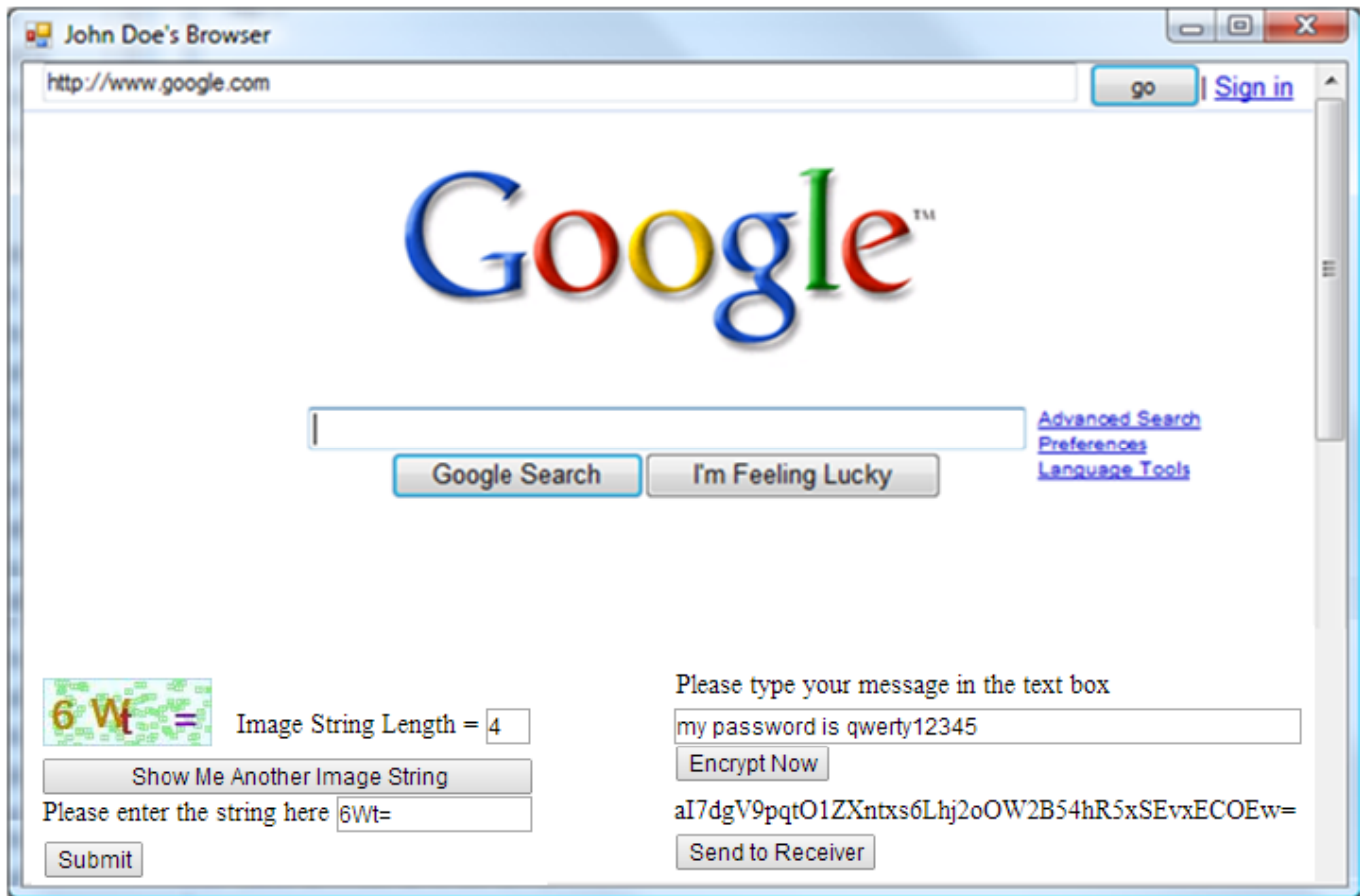
2. Follow the Tutorial 3 given in Section 2 to develop a Windows Forms Application to consume (access) the temperature conversion service. [10 points]
3. Follow the Tutorial 4 given in Section 2 to develop a Web Site Application to consume the temperature conversion service. The service must be running on localhost. [10 points]
4. Follow the tutorial in text Appendix Section A.1 to create a Web browser that can take any URL and display the content of the page in the window. [10 points]

In the following questions, you will add more features into the browser that you created in the previous question. Choose **two** questions only from the following set of questions. If you do more than **two**, we will grade the first **two** only. Each question is 30 points.

5. Add text encryption decryption function in your browser. Follow the example in text 3.6.3. However, instead of using the localhost service, you must use the remote service in the ASU Repository at: <http://venus.eas.asu.edu/WSRepository/Services/EncryptionWcf/Service.svc> [30 points]
6. Add the Get Stock Quote function in your browser. You can use, e.g., the Web service (<http://venus.eas.asu.edu/WSRepository/Services/Stockquote/Service.svc>) to build your application. For a given stock symbol, e.g., IBM, GOOG, you must display all the values returned in a readable format. [30 points]
7. Follow Tutorial 5 to add the image verifier into your Web browser. [30 points]

The figure below shows a sample layout of your browser. Notice that the sample components in the sample is different from what is required in this assignment. You must design your own layout to best display the required information. However, all parts of the information must be displayed in a **single page**.

If a particular service is not working, you can use another one with the same level of complexity, for example, the same number of input parameters.



Submission Requirement

All submissions must be electronically submitted to the assignment folder where you downloaded the assignment paper. The entire solution with all the files must be zipped into a single file.

If you have saved a project/Website in a different folder, you can copy the folder containing the project/Website to the directory where the other projects are saved. Then go into Visual Studio and delete the project/website that was in a different place. Then right click the solution in Visual Studio and add existing project/website, browse to the new location and select the project/website to link the moved project/website into the solution.

Submission preparation notice: The assignment consists of multiple **distributed** projects and components. They may be stored in different locations on your computer when you create them. You must copy these projects into a single folder for blackboard submission. To make sure that you have all the files included in the zip file and they work together, you must test them before submission. You must also download your own submission from the blackboard. Unzip the file on a different location or machine, and test your assignment and see if you can run the solution in a different location, because the TA will test your application on a different machine.

Grading

We will grade your programs following these steps:

(1) We will read your program and give points based on the points allocated to each component, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementations of each function.

(2) Compile the code. If it does not compile, 40% of the points given in (1) will be deducted. For example, if you are given 20 points in step (1), your points will become 12 if the program fails to compile.

(3) If the code passes the compilation, we will execute and test the code. If, for any reason, the program gives an incorrect output or crashes for any input, 20% of the points given in (1) will be deducted.

Please notice that we will not debug your program to figure out how big or how small the error is. You may lose 40% or 20% of your points for a small error such missing a comma or a space!

Late submission deduction policy:

- No penalty for late submissions that are received within 24 hours of the given deadline;
- **2%** grade deduction for every hour after the first 24 hours! No submission after Monday.