CSE445/598 Project 1 Assignments 1 and 2 – 50+50 Points)
Summer 2020

A1 and A2 Due: Saturday, May 30, 2020 by 11:59pm (Arizona Time), plus a one-day grace period

---

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 quizzes, the chapter tests, and the exams, 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.

9.1 Describe what you must do as an application builder, and what you can expect the service providers to do for you.

9.2 Describe your invention idea and list everything you must do as an application builder.

9.3 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?

12. Download Visual Studio in Windows, find Azure in MyASU MyApps, click on it and sign in with your official ASU email (not any alias), then you'll have access to all the software and they'll be giving you a key for the products. Enter the key to activate the software. You can download Visual Studio 2017 or 2019. If you download 2019 you'll need to include Windows communication Foundation to be able to create a WCF service. Mac VS does not support full VS features.

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. The recommended platform is Windows Visual Studio 2019.

Step 1: Start Windows Communication Foundation Project

Start Visual Studio 2019 and choose "Create a new project". You can search "WCF" and then choose "WCF Service Application" template, as shown in Figure 3.4.

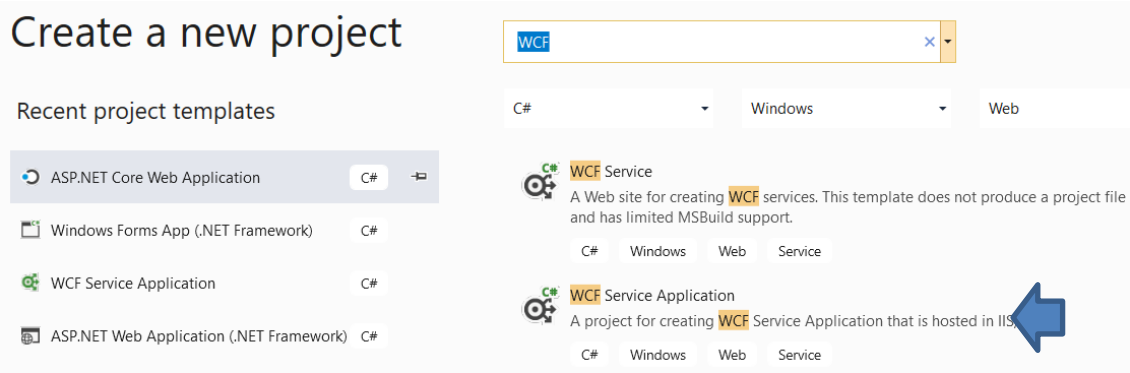


Figure 3.4. Creating a new project and choose WCF Service Application template

If you do not find this template, you need to start Visual Studio Installer. Run Visual Studio Installer and modify your Visual Studio adding components. When you add new components, you will not see Windows Communication Foundation in the Workloads tag. You need to click the "Individual components" tag and add Windows Communication Foundation, as shown in Figure 3.5.

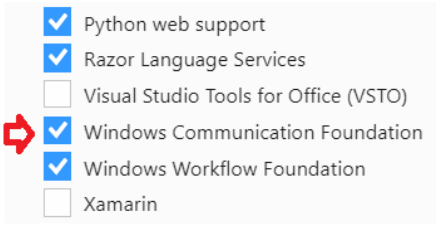


Figure 3.5. Install Windows Communication Foundation through individual components

Step 3: Follow Textbook, Chapter 3, Section 3.2.1 or A1 Tutorial, to develop a simple service using Visual Studio.

Tutorial 2: Using WCF Test Client

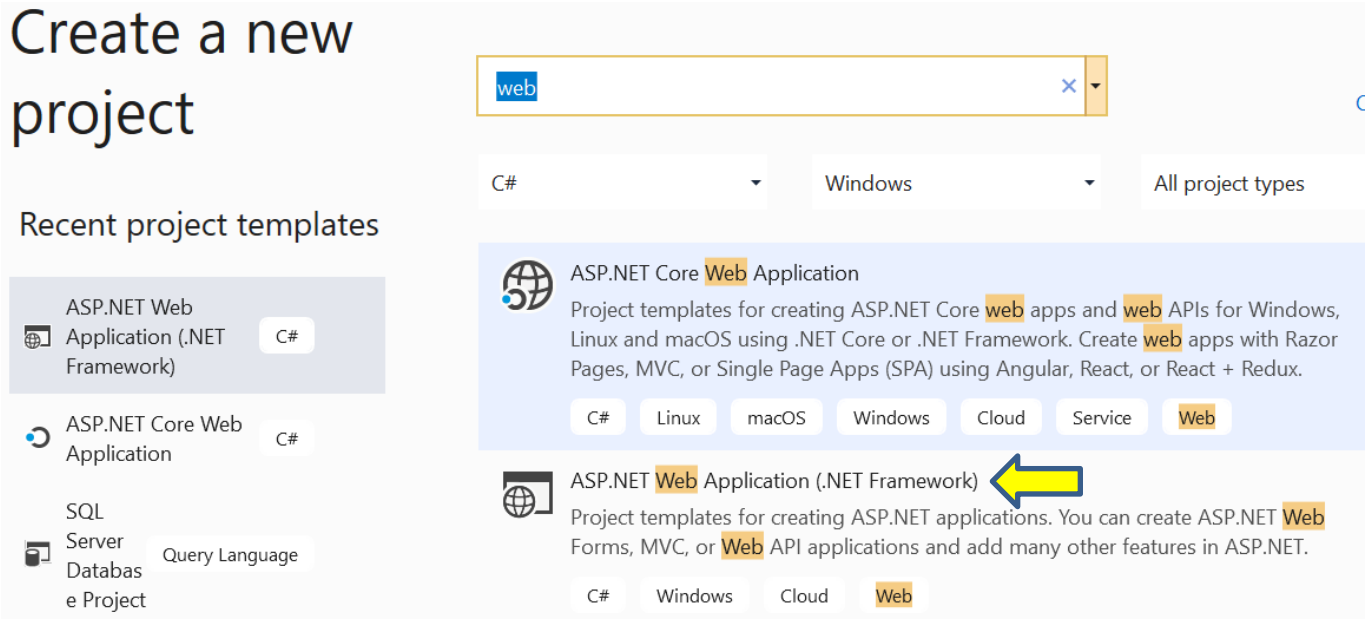
Follow the text Chapter 3, Sections 3.2.2 AND 3.2.3 or lecture slides to test your service using WCF test client

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

In this tutorial, you will follow the textbook, Appendix A.1 and Chapter 3 Section 3.6.2, or follow Microsoft document to develop a Windows Forms application, and follow Section 3.6.3 to use the WCF services. This tutorial shows you how an application builder makes use of remote services (.svc 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 Application to consume the service that you developed in tutorial 1. In Visual Studio, Choose Create a new project. Search "web" and choose ASP .Net Web 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.

Assignment 1: Web Services (Submission required, 50 points)

This 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

2. Develop a Web service to sort a string of numbers, separated by commas. It returns a string of numbers, sorted and separated by commas. The service contains one operation: [10 points]

   string sort(string s); // sort a string of numbers, separated by commas

   In this question, you can use any algorithm or any library function to sort the numbers.

3. Follow the Tutorial 3 given in Section 2 to develop a Windows Forms Application to consume (access) the temperature conversion service. Since the service that you developed in question 1 is a .svc service, you also need to follow Tutorial 4 to call the .svc service. The service must be running on localhost when your application calls the service. [10 points]

4. Follow the Tutorial 4 given in Section 2 to develop a Web Site Application to consume the number sorting service that you developed in Question 2. The service must be running on localhost when your application calls the service. [10 points]

5. 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]

Assignment 2: Adding Features in Web Browser (Submission required, 50 points)

In the following questions, you will add more features into the browser that you created in Assignment 1. 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 25 points.

1. Add a simple calculator in your Web browser, with floating point +, -, *, and / operations. No service call is required in this question. [25 points]

2. 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 service in the ASU Repository at: http://neptune.fulton.ad.asu.edu/WSRepository/Services/EncryptionWcf/Service.svc [25 points]

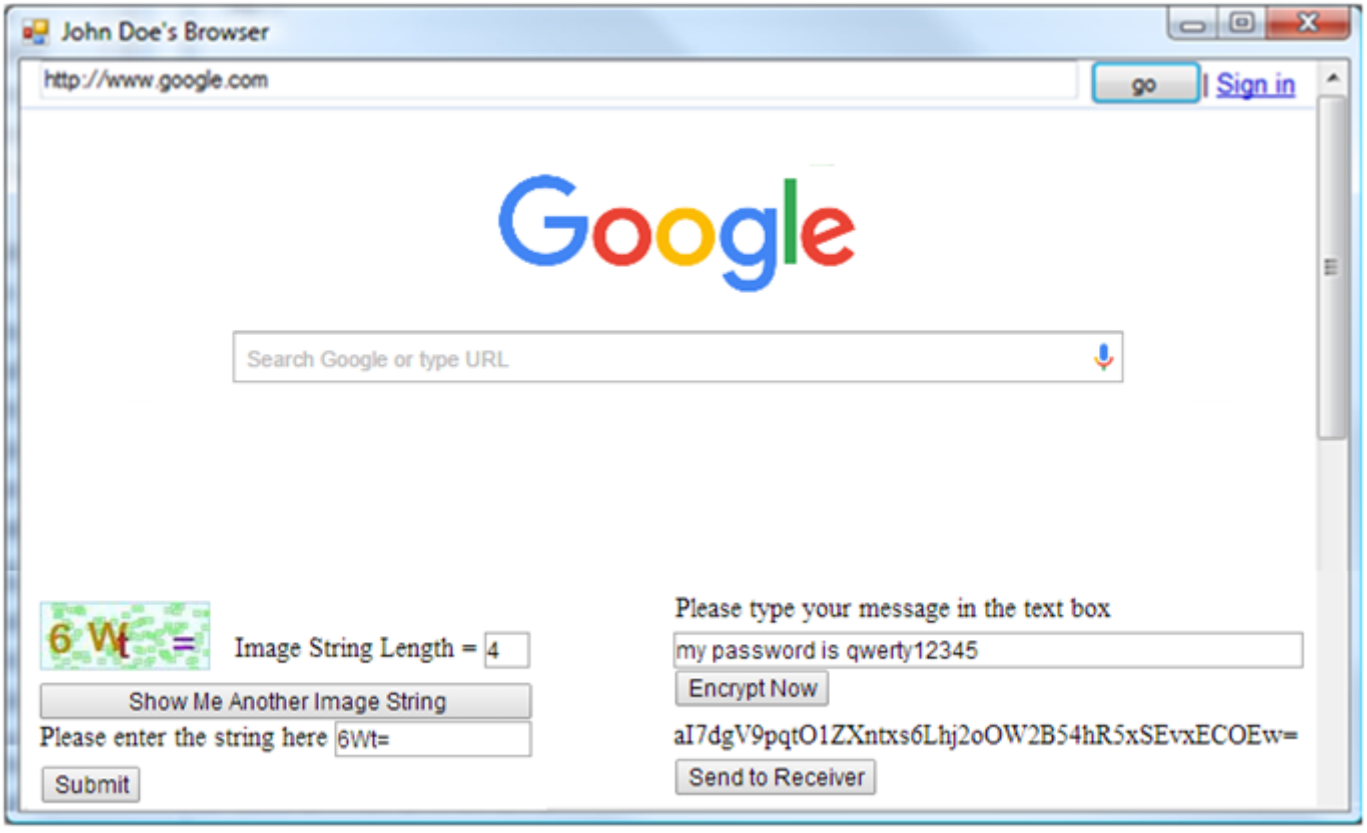
3. Add the Get Stock Quote function in your browser. You can use the stock simulator service (http://neptune.fulton.ad.asu.edu/WSRepository/Services/Stockquote/Service.svc), or discover your own stock service, to build your application. For a given stock symbol, e.g., IBM, GOOG, you must display all the values returned in a readable format. [25 points]

4. Follow Tutorial 5 to add the image verifier into your Web browser. [25 points]

5. Find a temperature Web service, and add the temperature into your browser. [25 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.



A1 and A2 Submission Requirement

All submissions must be electronically submitted to the assignment folder on Canvas. The all solutions with all the files must be zipped into a single file, so that the solutions can be unzipped and tested.

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 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. If you submitted an empty project folder, an incomplete project folder, or a wrong folder, you cannot resubmit after the submission linked is closed! We grade only what you submitted in the blackboard. We cannot grade the assignment on your computer or any other storage, even if the modification date indicated that the files were created before the submission due dates.

The blackboard submission may take a few minutes if the file is big.

Grading and Rubrics

Each sub-question (component) has been assigned certain points. We will grade your programs following these steps:

(1) Compile the code. If it does not compile, 50% of the points given for the code under compilation will be deducted. Then, we will read the code and give points between 50% and 0, as shown in right part of the rubric table.

(2) If the code passes the compilation, we will execute and test the code using test cases. We will assign points based on the left part of the rubric table.

In both cases (passing compilation and failed compilation), we will read your program and give points based on the points allocated to each component (sub-question), 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.

Please notice that we will not debug your program to figure out how big or how small the error is. You may lose 50% of your points for a small error such missing a comma or a space!

We will apply the following rubrics to **each sub-question** listed in the assignment. Assume that points assigned to a sub-question is *pts*:

Rubric Table

Major	Code passed compilation				Code failed compilation		
Points	pts * 100%	pts * 90%	pts * 80%	pts * 70% - 60%	pts * 50% - 40%	pts * 30% - 10%	0
Each sub-question	Meeting all requirements, well commented,	Working correctly in all test cases.	Working with minor problem, such as not	Working in most test cases,	Failed compilation or not working	Failed compilation, showing some effort,	No attempt

	and working correctly in all test cases	Comments not provided to explain what each part of code does.	writing comments, code not working in certain uncommon boundary conditions.	but with major problem, such as the code fail a common test case	correctly, but showing serious effort in addressing the problem.	but the code does not implement the required work.	

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.