

CSE445/598 Assignment/Project 1 (100 Points)

Fall 2013

Due: Saturday, September 7, 2013 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 “Lecture Slides”. Doing these exercises will help you prepare your **quiz 1** test at the end of chapter 1 lecture.
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 service

Follow the tutorial in textbook, Chapter 3, Section 3.2.1, to develop a simple service using Visual Studio. This tutorial shows how a service provider develops services for the application builders' use.

Tutorial 2: Creating an application using the service that you developed in tutorial 1.

You can follow the process of the tutorial given in the textbook, Section 3.6.3, but use the service that you developed in tutorial 1 instead.

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

Tutorial 3: Creating an application using the weather service

Follow the tutorial given in the textbook, Appendix sections A.4 and A5, to develop an application.

Section 3 Web Services (Submission required, 100 points)

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

1. Follow the tutorial 1 given in Section 2 to develop a Web service that has two operations: [5 points]

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

`int f2c(int f); // convert Fahrenheit temperature to Celsius temperature`
2. Follow the tutorial 1 given in Section 2 to develop a Console application to consume (access) the temperature conversion service. [5 points]

3. Follow the tutorials 2 and 3 given in Section 2 to develop a Windows Forms Application that consumes the temperature conversion service. [5 points]

Notice that, in order to test the client, 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 solution and choose “View in Browser.”

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 upper part of the window. [10 points]

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

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> [25 points]
6. Add the Get Stock Quote function in your browser. You can use, e.g., the Mortgage Web service (<http://www.websvcex.net/WS/WSDetails.aspx?CATID=2&WSID=9>) 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]
7. Add the mortgage payment calculator into your browser. You can use, e.g., the Mortgage Web service (<http://www.websvcex.net/WS/WSDetails.aspx?WSID=15&CATID=2>) to build your application. [25 points]
8. Add weather service into your browser. Based on the user-entered zip code, the Web browser will display the location name and its five-day weather forecast in the lower part of the window. [25 points]

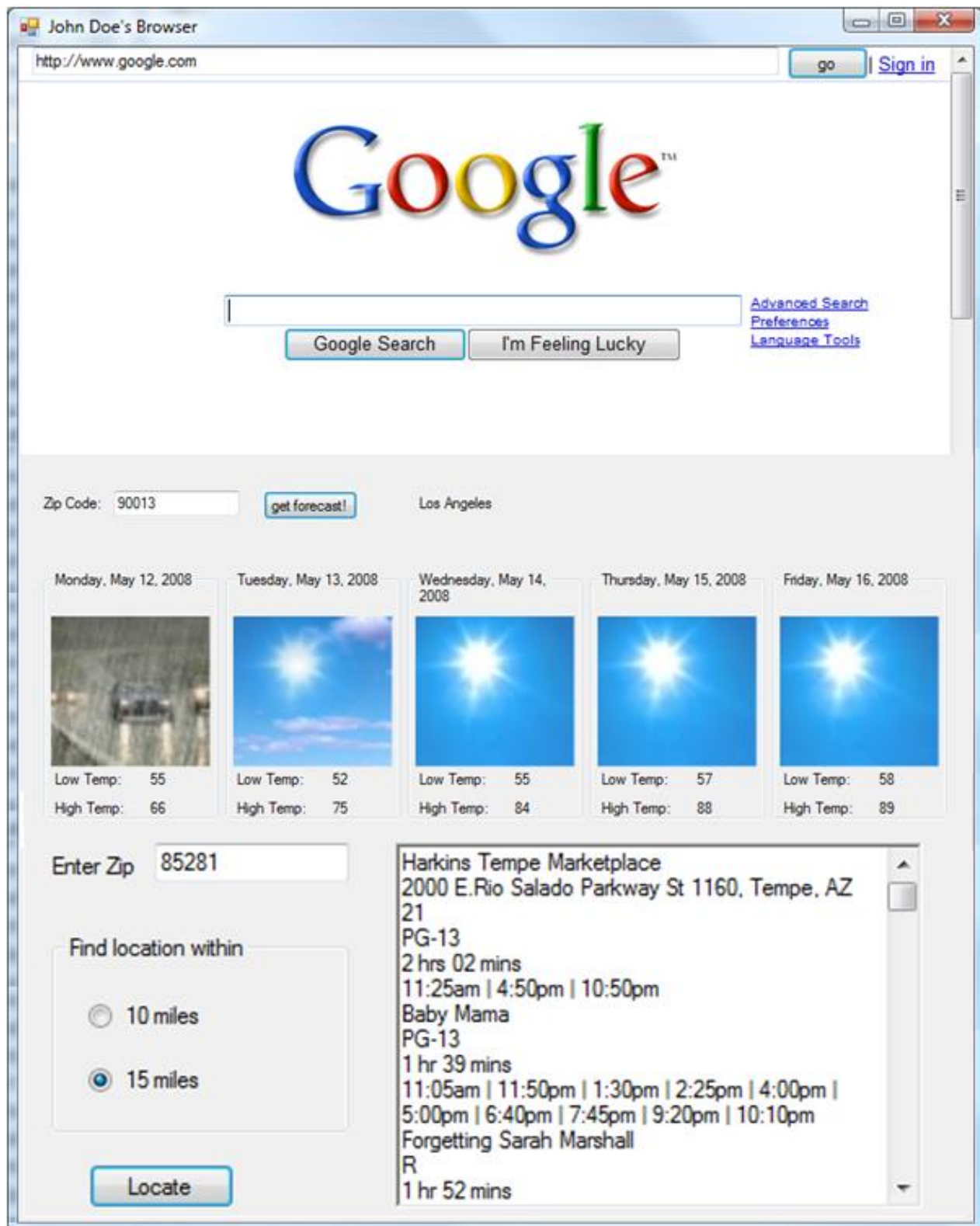
Note, if the weather services used in the tutorials are not available, you can find an alternative service to replace the service. For example, the following services also provide weather information:

http://graphical.weather.gov/xml/SOAP_server/ndfdXMLserver.php?wsdl [Ref: <http://graphical.weather.gov/xml/>]
or <http://ws.cdyne.com/WeatherWS/Weather.asmx?WSDL>, or from

<http://www.websvcex.net/WS/wscatlist.aspx>. These services may or may not have the exact functions given in the tutorial, and you must adjust the functionality of your application.

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 three parts of the information must be displayed in a **single page**.

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



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!

Submission Requirement

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

Late submission deduction policy:

- No penalty for late submissions that are received within 24 hours of the given deadline;
- 1% grade deduction for every hour after the first 24 hours of the grace period!