Portfolio Notes

April 17, 2014

PDF Portfolio Requirements

This document describes how you create your team Web portfolio.

Getting Started

- Download the compilers2014.zip file from the shared Google Drive link:

  https://drive.google.com/file/d/0B_d2Jw0DsGMPVHJqTWVXd3RoeGc/edit?usp=sharing

- Unzip this file in the location where you want to work on your portfolio.
- Look in Contents > 02_Portfolios, where you will find the folder 00_Team_0.
- Rename folder 00_Team_0 to represent your team. For example, if you team number is 9, then the folder should be renamed 09_Team_9.
- You will be making modifications to the contents of your renamed team folder.
- As you construct your portfolio, DO NOT make any changes to filenames, stylesheets, scripts, or hyperlinks.
- For including code, we are using Google's prettify Javascript program. The tags for doing so are already in the pages you need to fill in, but you may need to modify the tag that surrounds source code to indicate which programming language you are using. For details, visit: https://code.google.com/p/google-code-prettify/.
- NOTE THAT YOU WILL NOT BE ABLE TO SEE THE MENU LINKS FOR NAVIGATING PAGES. THESE ARE GENERATED AUTOMATICALLY BY A PERL SCRIPT ON THE SERVER WHEN THE WEBSITE IS PUBLISHED.

Website Portfolio Requirements

The instructions given here are listed per required section of the portfolio. For each required section there is a folder in the website for your team. The folder names are the names of the sections below. There is a Department capstone portfolio requirement that in some cases is not entirely relevant to the Compilers course. The Department requirements follow the sections below, except for the Introduction. The Department requirements are listed in italics, and then amended as necessary for your portfolios.

NOTE: IN WHAT FOLLOWS, THE STATEMENTS IN *ITALICS* ARE THE GENERAL REQUIREMENTS SPECIFIED FOR THE PORTFOLIO. BE CAREFUL TO READ THE PARAGRAPH THAT FOLLOWS EACH *ITALICIZED* STATEMENT FOR THE SPECIFIC YOUR REQUIREMENTS FOR THE COMPILER PORTFLOIO, AS THESE REQUIREMENTS OFTEN MODIFY THE GENERAL REQUIREMENTS.

0_Introduction

This section contains cover.html. On this page you just fill in your team member information as shown. No narrative is needed.

1_Program Folder

The 1_Program folder contains four other folders, 1_Scanner, 2_Parser, 3_Symbol_Table, and 4_Semantic_Analyzer. Each of these folders contains two outlined HTML pages. The first HTML page is for commentary (to satisfy the technical writing component of the portfolio) and the second contains just your source code for the given section (Scanner, Parser, Symbol_Table, and Semantic_Analyzer).

1-1. The Commentary HTML Page

Fill in one commentary section per team member. The idea is for each team member to do one sample of technical writing, so you only need to fill in as many of these commentary sections as you have team members. However, for a complete portfolio it would be nice to have all filled in (however, having more filled in that there are team members is not a

requirement that will affect your grade). ALTHOUGH THE WRITING PORTION OF THE COMMENTARIES CAN BE BRIEF, DON'T ENTIRELY BLOW THEM OFF, AS THES SECTIONS ARE INTENDED TO SHOW THAT YOU ARE CAPABLE OF PRODUCING TECHNICAL DOCUMENTS THAT ARE TRULY INFORMATIVE AND HELPFUL TO READERS.

1-2. The Source Listing HTML Page

Attach the source listing of the program that you wrote. Include the specifications for the program.

Since our portfolios constitute a website you will be including your code in Web pages. This section contains HTML pages for the source code of your scanner, parser, symbol table, and semantic analyzer. You are to put your program code in these sections for the scanner, parser, symbol table, and semantic analyzer, respectively. The Department requirement says that you are to include the specifications there, but I provide the specifications, with some room for choice. So, each of these pages has a link to another page that gives the specifications. Leave that link in each page. I will be filling in the specifications. In a later section you will have opportunity to discuss design decisions.

NOTE: BE SURE THAT THE CODE YOU ENTER IS NICELY READABLE. ESPECIALLY IMPORTANT IS TO ALLOW NO LINE WRAPS. INSTEAD MANUALLY BREAK LINES THAT ARE TOO LONG SO THAT THEY PRESERVE PROPER INDENTATION AND READABILITY.

2_Teamwork

Describe how your team worked on this capstone project. List each team member's primary contributions and estimate the percentage of time that was spent by each team member on the project.

Fill this section in as specified. Be sure to use proper HTML coding for headings, paragraphs, bulleted points, etc., as appropriate.

3_Design_Patterns

Identify one design pattern that was used in your capstone project and describe exactly where in the code it is located. Highlight the design pattern in yellow. Explain why you used the pattern and didn't just code directly.

This section about design patterns is not generally relevant to the class. We used a standard Facade design pattern and standard patterns for designing both the front-end of a compiler and for the design of the scanner, parser, and semantic analyzer. However, some of you may have chosen a design pattern for the symbol table or for some other aspect of the project where you deviated from the recommended approaches If so, be certain to include an example of one such pattern you chose and follow the instructions given directly in this HTML page for design patterns.

4_Technical_Writing

Leave this page as is. I will be modifying it to point readers to the commentary sections of the 1_Program folder.

5_UML

*Attach the UML design diagrams for your capstone project that were created **before** you began coding.*

Since we did not have an assignment for you to come up with a UML diagram before you began coding, you can instead do one of the following.

1. Give a UML diagram of one part of the compiler, such as the symbol table, that you may have had an initial design for that you later changed. In this case give the UML diagram as best as you can recall for the original design and the UML diagram for the current design. THERE IS A SAMPLE UML DESIGN IN THE PORTFOLIO SKELETON THAT YOU CAN USE AS AN EXAMPLE, BUT YOURS SHOULD REFLECT YOUR PROGRAM.
2. If you have not encountered a situation in which you changed your original design, just say so and give the UML diagram of one of the components of the compiler.

You can create your UML diagram and then save it as a .jpg file. You can insert this image into a page with the <img> source tag as in:

<img width="720px" src="relative path to the image file.jpg">

6_Design_Tradeoffs

Describe a design trade-off decision (e.g. execution time vs. space requirements or compile time) in your capstone project and justify the design decisions that you made.

Use any interesting case you can recall for the design of your compiler. The obvious one would be a discussion of why you chose your particular ADT for the symbol table, and what the time complexity of, say, identifier lookups in the symbol table is over compilation of the entire program.

7_Life_Cycle
Describe the model that you used to develop your capstone project. How did this model help and/or hinder your team?

Use your understanding of software engineering to provide input for this section.

Submission Requirements

When you have completed your team portfolio, zip JUST YOUR TEAM FOLDER and drop it in the Portfolio dropbox on D2L. Note that we will not be able to give the team members final grades until the portfolio is turned in and works on the Portfolio Website.