An Undergraduate Course in Perl: An All Purpose Programming Language

Jason Moore, Mitchell A. Thornton
Southern Methodist University
Dallas, Texas

Ronald W. Skeith
University of Arkansas
Fayetteville, Arkansas

Abstract

Perl programming is a skill that students will not fully appreciate until they have mastered it. Perl is best known as a scripting language. Even people who know how to program in Perl generally only think about using it for automating other processes. Perl is also known for its powerful text manipulation capabilities through the use of regular expressions. However, Perl's greatest strength is its versatility. This versatility allows for students to learn about web-development through Common Gateway Interface (CGI) scripts, Object Oriented Practices, and regular expressions all in the context of one language. The flexibility of Perl allows students to incorporate for previous individual assignments into a single application. One of the later course assignments is to parse in different sports teams rosters from their web pages. Once the rosters are parsed in, the program must allow the user to search for different players by name, height, weight, team, or number. Students also learn how to use Perl's text manipulation to take output from one program and prepare it for input to a different program. Even though Perl programming is not a skill that companies advertise for, it is a used in many companies and can help students keep their future jobs. Former students have found their Perl programming skills extremely useful not only in other Computer Engineering courses but in Business classes and industry. This paper will examine this course from the perspective of student, teaching assistant, and professor.

Introduction

This paper describes a class that teaches students a large variety of needed programming skills. Such programming skills include shell scripting, object oriented programming, text manipulation, and Internet web-development. The main obstacle in teaching all these skills in one course is finding a language versatile enough. If such a language is not found, too much time is wasted teaching the basics of new languages to cover all of those programming topics. Perl is a versatile enough language to cover all of these topics.

At first, students are given small text manipulation programs to learn how to use regular expressions, vi which is a common Unix text editor, and develop in a Linux environment. The assignments gradually get larger with each requiring an additional skill to complete. All of the assignments build to a final project that requires a mastery of several skills to successfully finish. The class size is usually around 30 students. Students are given three examinations that test the programming skills learned from the assignments. Students are expected to use an object-oriented approach when developing their program for the final project.

The remainder of this paper focuses on what students are asked to do and how they react to it. The paper gives examples of how the assignments build on one another and common problems encountered by students.

Building from one assignment to the next

Even with early assignments, students are required to divide the program into sub-routines in logical manner. Even if a student was able to pull off the assignment as a one line Perl command, he or she will not get credit for this assignment. This is announced several times during the class.

The initial assignment for the class is to print out the second largest number and the line on where the number was found from a file containing a number on each line. This assignment gives the students experience with sub-routines, text manipulation, and vi. In the next assignment a big jump is made as students are asked to use object oriented principles to complete the assignment using packages in Perl. Students are asked to access a text file containing employee information. The following features are required for the program: display all employees whose last name ends with son, percentage of employees over 44, the average salary of all the employees, and display an alphabetically sorted list of all employees who are over 44 and make less $35,000. Assignment 3 serves two purposes to start preparing students for the final project and to give them time before assignment 4 to fix any problems with assignment 2. Assignment 3 is to use a text version of the roster of the Kansas City Chiefs to print the roster in the following format: number:first name:last name:height:weight:birthday:age:team. Figure 3 shows the initial roster. In assignment 4, students are asked to modify their second program to add, delete, and edit employees. Students must also add the capability to handle hourly employees. The difficulty level of assignment 4 is completely dependant on how good of a job the student did on assignment 2. Assignment 4 introduces the idea of inheritance to students. Students are required to inherit from an Employee class to a SalaryEmployee class and an HourlyEmployee class. Figure 1 shows a UML diagram for this assignment. Assignment 5 is the development of a small CGI script in order to give students some experience with using Perl for web-development. Students are asked to add a poll to their home page.
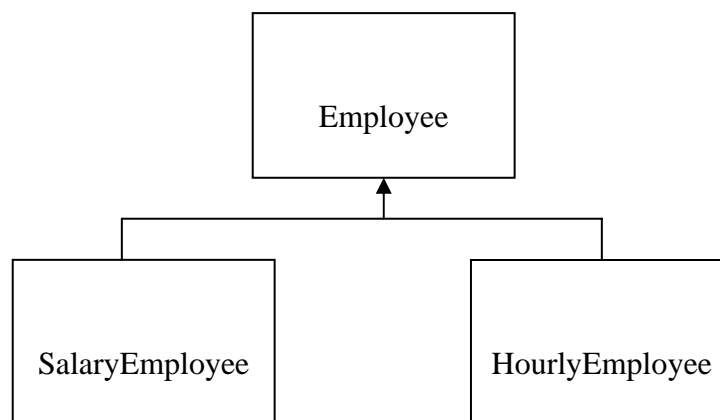

Figure 1: Inheritance in assignment 4

Final Project

The final project builds on assignment 4. Students are asked to turn the downloaded text version of the alphabetical rosters of the Kansas Chiefs, Denver Broncos, Dallas Cowboys and New York Giants into a readable format. The rosters are from 1999 because after 1999 the NFL changed all their official websites to be the same format. All four of the rosters downloaded have a different format, which makes the assignment more challenging. Like assignment 3, students are forced to use inheritance, but they must also use polymorphism through inheritance. In assignment 3, they were still able to call SalaryEmployee and HourlyEmployee directly. In the final project, they are forced to write a class of sub-routines that accept type Roster in order to accomplish the following tasks: display a formatted roster, display all quarterbacks over 6 feet tall, all runningbacks under 5'10'', and search for a player by last name. In addition to those functions, students were also asked to come up with their own tasks to do with the data. Figure 2 illustrates the layout of the final project. Figure 3 shows part of the unformatted Kansas City roster, and figure 4 shows part of an example output for the formatted roster.
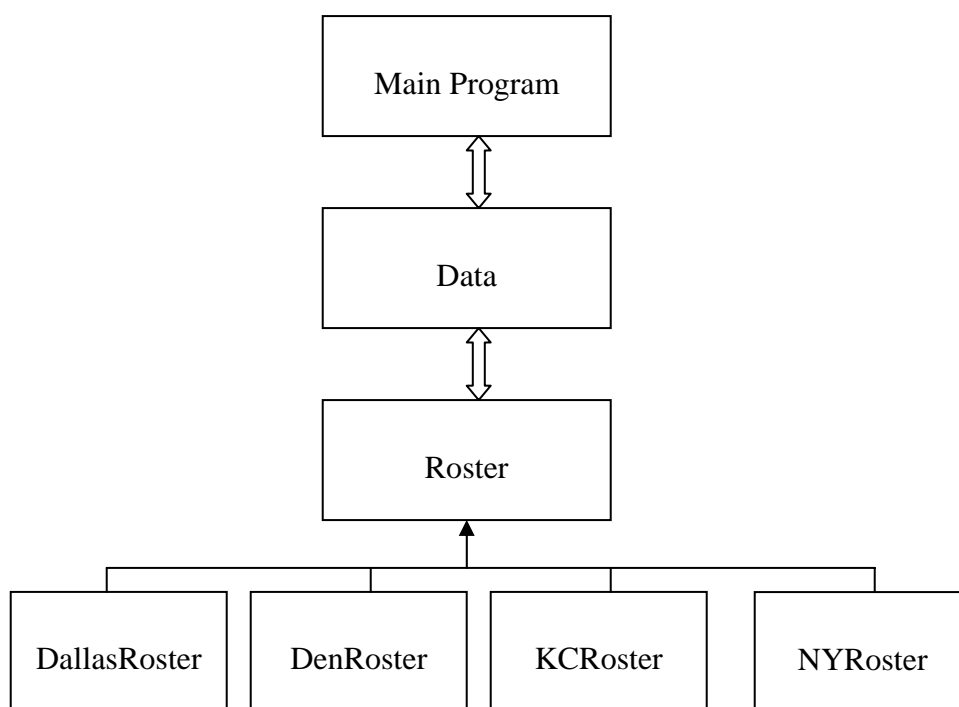

Figure 2: Final Project Layout

Student Reaction

Students seem to really enjoy the class. Some students have been known to bring their laptop computers to class to try out what the professor says as soon as he says it. If at least one student has a laptop computer running Linux and one running Windows with cygwin, this gives the professor the unique opportunity to illustrate in class how some Perl scripts run differently on Linux than cygwin under Windows.

The most intriguing student reaction came after a particular student had completed the

course. He was assigned in his business class to check a website daily for the price of certain stocks. He made the observation that he could modify his final project to download the website directly and output the stock prices to a text file that would be readable by Excel. He recruited another former student to complete this task. He then scheduled the program to run several times a day. He was able to compile some interesting data on the stocks and create some very detailed charts.

Alphabetical Roster

1999 KANSAS CITY CHIEFS - ALPHABETICAL ROSTER
As of February 20, 2000
This table may take awhile to load.

NO.NAMEPOS.HT.WT.BIRTHDATEAGEEXP.COLLEGEHOMTOWNHOW
ACQ.
82Derrick
AlexanderWR6-221011/06/1971276MichiganDetroit, MIUFA(Balt)-98
35Larry
AtkinsS6-323007/21/197524RUCLAVenice, CAD3b-99
30Donnell
BennettRB6-024509/14/1972266Miami (Fla.)Ft. Lauderdale, FLD2-94
34Mike
CloudRB5-1020507/01/197524RBoston CollegePortsmouth, RID2-99
15Todd
CollinsQB6-422811/05/1971275MichiganWalpole, MAW(Buff)-98
26Cris
DishmanCB6-019608/13/19653312PurdueLouisville, KYFA-99
18Elvis
GrbacQB6-523708/13/1970287MichiganCleveland, OHUFA-97
    (SF)

Figure 3: Initial KC Chiefs Roster

Kansas City Chiefs Roster

Number: 82
Name: Derrick Alexander
Position: WR
Height: 6-2
Weight: 210
Birth Day: 11/6/1971
Team: Kansas City Chiefs
Age: 28

Figure 4: Formatted KC Roster

Conclusion:

Perl's versatility allowed for the class to cover several different aspects of software development without teaching multiple programming languages. Students were exposed to object oriented techniques such as inheritance and polymorphism. They were also able to gain experience writing scripts and manipulating text using regular expressions. They have used the skills in their jobs, computer classes, and other classes.

JASON MOORE
Jason Moore received the Masters degree in computer science from Southern Methodist University in December 2004. He has 2 years of industrial experience. Currently, Jason is a PhD student in computer science and engineering department at Southern Methodist University.

MITCHELL A. THORNTON
Mitchell A. Thornton received the Ph.D. in computer engineering from Southern Methodist University in August 1995. He has 6 years of industrial experience and 11 years of academic experience. Currently, Mitch is an Associate Professor of computer science and engineering at Southern Methodist University.

RONALD W. SKEITH
Ronald W. Skeith received the Ph.D. in Engineering from Arizona State University. He is currently employed as a faculty member at the University of Arkansas where he has served for over 40 years and he is a Professor of computer engineering.