

CS306 – Introduction to Perl

Spring 2006

Homework Assignment #3

Due: Monday, February 27th, 2006 (12:20pm)

Please see the guidelines on homework #1 on how to submit your homework.

Homework #3

Question 1. (4 pts) What are filehandles? What are some examples of filehandles that Perl defines for you? What are some examples of a programmer-defined filehandle (include code sample)? How do you use filehandles to interact with files?

Question 2. (4 pts) Explain how the following program behaves. Be sure to consider all possibilities.

```
#!/usr/bin/perl
while ($line = <>) {
    print "$line";
}
```

Question 3. (4 pts) What do the following conversions define when used in a printf statement?

3a. %-10.4f

3b. %+3d

3c. %g

3d. %#d

Question 4. (4 pts) What is the pipeline model?

Question 5. (4 pts) What does the readdir() function return in list context? What does it return in scalar context? Provide three ways to avoid including the files . and .. when reading a directory, including at least one way that is not in your textbook.

Program 1. (10 pts) Let's extend the gas station tab program from homework 2 so that it saves the data between runs. To accomplish this, your expanded program will have to take the name of an input file as a command line argument (not as a prompt to the user). If it exists, it should open it and parse the data from the previous run into memory. If it does not exist, your program should create it sometime before the end of execution. When the user requests to exit the program, the program should write the data back out to disk. You can choose whichever format you prefer for storing the data in the file. If the user does not provide any name of the file on the command line your program should use standard input. Make sure to document your storage format thoroughly.

Program 2. (10 pts) Now extend the gas station tabs program one step further, by introducing functions. Abstract each of your operations (add to a tab, make a payment onto a tab, create a new customer, exit the program, etc...) into its own function. After doing this, the main part of your program should simply become a looping menu, that takes requests from the user and calls the appropriate function. When the function is done, execution should return to the main menu loop (except in the case of the user requesting to exit the program).