Syntax and Comprehension

Below is code you should have mastered at this point. You should understand what each piece means and be able to produce it without copying and pasting from previous examples. If you can’t do these, you should do them over and over, looking up less and less each time, until you can do them easily.

1. Open a file called “myfile.txt” from your computer (NOT from the web). Loop through each line of that file and print out each line.

*If you struggle with this, revisit the notes from week 3 - Files.*

open (FILE, “myfile.txt”);

while ($line = <FILE> ) {

 print $line;

}

2. Below is a piece of code I have written:

1. print “Enter a book into our system. First, enter the title: “;
2. $title = <>;
3. chomp $title;
4. print “Next, enter the author: “;
5. $author = <>;
6. chomp $author;
7. print “Do you want to enter another (Y/N)?”;
8. $another = <>;
9. chomp $another;

Put that code in a loop that will repeat until the user enters “N” in response to the last question.

*If you struggle with this, please revisit the notes on loops from week 2*

while ($another ne “N”) {

print “Enter a book into our system. First, enter the title: “;

$title = <>;

chomp $title;

print “Next, enter the author: “;

$author = <>;

chomp $author;

print “Do you want to enter another (Y/N)?”;

$another = <>;

chomp $another;

}

3. You have an array called %students. The keys of the array are student names and the values are their scores in the class. Loop through the array and print out a table of data with the student name and score that looks like this:

Bob Smith – 90

Priya Subramaniam – 95

*If you struggle with this, please revisit the notes on hashes from week 2*

foreach $student (keys %students) {

 print “$student – “;

 print $students{$student};

 print “\n”;

}

4. You have a comma separated file of student grades that looks like this:

Bob Smith, 90

Priya Sumbramaniam,95

Using your code from 1, open the file and loop through it. For each line, print out the student name *only* if their score is 90 or higher.

*If you struggle with this, first try to loop through the file. If you can’t do that, see #1.*

*If you struggle with splitting the line to get the score out, revisit the notes on regular expressions and the split command*

*If you struggle with deciding when to print, revisit the notes from week 1 on conditional statements*

open (FILE, “myfile.txt”);

while ($line = <FILE> ) {

 ($student,$grade) = split(/,/,$line);

 if ($grade>=90) {

 print “$student\n”;

 }

}

close FILE;

5. You have the same file described in 4. Store the data in the hash called %students where the student name is the key and the value is the score.

*If you struggle with storing data in hashes, revisit the notes from week 2 on hashes.*

open (FILE, “myfile.txt”);

while ($line = <FILE> ) {

 ($student,$grade) = split(/,/,$line);

 $students{$student}=$grade;

}

close FILE;

6. Consider the code from #2. Add a line before the current line 6 that checks that the user has actually typed in a title and author. If they left either blank (i.e. the just hit enter after the prompt without typing anything), print a message that says “You forgot to enter a title or author.”

*If you struggle with conditional statements, revisit the notes from week 1 on conditions.*

print “Enter a book into our system. First, enter the title: “;

$title = <>;

chomp $title;

print “Next, enter the author: “;

$author = <>;

chomp $author;

if ($author eq “” || $title eq “”) {

 print “You forgot to enter a title or author\n”;

 }

print “Do you want to enter another (Y/N)?”;

$another = <>;

chomp $another;