Programming in Perl
Midterm Exam review

The exam will cover everything we’ve done so far
The exam will have 4 or 5 sections:

• True/False – simple statements about Perl
• What’s wrong here? – You will be given a chunk of code that does not do what it’s supposed to. Explain why, and fix it.
• Short answer – 2- or 3-sentence answers to questions about Perl.
• What’s the output? – given a chunk of code, what does it print? (similar to class handouts)
• Code it – write a couple simple programs.

There will be one bonus question, worth either 5 or 10 points

Sample questions:

• The first member of the array @ARGV is always the name of the Perl script. True or False?

• If there are no command-line arguments, the <> operator acts on STDIN.  True or False?

• This code fragment does not work. Why? Fix it:

```perl
$a = 30;
$b = 100;
if ($a lt $b){
   print "$a is less than $b\n";
} else {
   print "$a is greater than or equal to $b\n";
}
```

• This code fragment also does not work. Why? Fix it:

```perl
if ($a == 10){
   print "$a is equal to ten\n";
} else if ($a > 10){
   print "$a is greater than ten\n";
} else {
   print "$a is less than ten\n";
}
```
• Briefly list and describe the three main types of Perl variables

• Briefly explain when/how variables $1, $2, $3, etc get set.

• What is the output of this code?
  ```perl
  @animals = ("cat", "dog", "bird", "lizard");
  print "These are four animals: ". @animals . "\n";
  ```

• What is the output of this code?
  ```perl
  foreach ('a'..'z'){
      next if /[aeiou]/;
      print;
      print "\n";
  }
  ```

• In the following chunk of code, assume the user enters “I love Perl!” and “Perl is my friend” at the prompt. What is the output of the code?
  ```perl
  while (<>){
      $i=0;
      foreach (split / /){
          print "$i:$_ ";
          $i++;
      }
      print "\n";
  }
  ```

• Write ONE regular expression that will replace all instances of < with &lt; and all instances of > with &gt;

• Write a program that reads an unknown number of words from the command line, and prints the words out in the opposite order from which they were entered. Change all upper-case letters to lower-case, and all lower-case letters to upper-case. Do not use the reverse keyword
  Example: if the user enters
  ```bash
  prog1.pl Hello, My name is Paul
  ```
  your program will output:
  ```bash
  pAUL IS NAME mY hELLO,
  ```