

## Programming in Perl

### Midterm Exam review

The exam will cover everything we've done so far (Regular expressions will be included, but will play a minor part in the exam).

The exam will have 4 or 5 sections:

- True/False – simple statements about Perl
- What's wrong here? – You will 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 VERY simple programs.

There will be no bonus points or extra credit available.

Sample questions:

- The 'default variable' for many functions is \$0. True or False?
  - False. The default variable is \$\_
- If there are no command-line arguments, the  $\langle \rangle$  operator acts on STDIN. True or False?
  - True
- This code fragment does not work. Why? Fix it:

```
$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";
}
```

- It is the string comparison operator. the < operator must be used instead.
- This code fragment also does not work. Why? Fix it:

```
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";
}

```

- else must be followed by a block, not a statement. Either put { } around everything following the first else, or change “else if” to “elsif”.
- Briefly list and describe the three main types of Perl variable
  - Scalar: holds any single value – integer, float, string, char, Boolean, or anything else
  - Array: holds a collection of other values. Indexed numerically starting at 0.
  - Hash: holds a collection of other values. Indexed by anything, not limited to numbers 0, 1, 2, 3,....
- Briefly explain when/how variables \$1, \$2, \$3, etc get set.
  - When part of a pattern match is enclosed in parentheses, and that pattern match succeeds, the values of the patterns within the parentheses are assigned to \$1, \$2, \$3, etc.

- What is the output of this code?

```

@animals = ("cat", "dog", "bird", "lizard");
print "These are four animals: " . @animals . "\n";

```

- “These are four animals: 4”<newline>

- What is the output of this code?

```

foreach ('a'..'z'){
    next if /[aeiou]/;
    print;
    print "\n";
}

```

- b c d f g h j k l m n p q r s t v w x y z (all separated by newline)

- 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?

```

while (<>){
    $i=0;
    foreach (split / /){
        print "$i:$_";
        $i++;
    }
    print "\n";
}

```

- 0:I1:love2:Perl!
- 0:Perl1:is2:my3:friend

- Write a simplified version of the Unix cat command. This command reads one or more file names from the command line, and prints their contents to the screen. (Do not concern yourself with error checking in this code)

```
foreach $file (@ARGV){
    open FILE, $file;
    foreach (<FILE>){
        print;
    }
    close FILE;
}
```

Alternative Answer:

```
while (<>){
    print;
}
```

To understand why this works, see the description of the `<>` operator in Programming Perl