# CSCI-2962 Programming in Perl
# Final Exam Review Questions

The final exam will cover everything we have done in the 2nd half of the class, beginning with Regular Expressions. This includes: Regular Expressions, Lookaround assertions, Subroutines, Objects, CGI, and the "Random bits of Perl" from our missing class.

The exam will be open-book, open-note. You may use the Camel, the Llama, another Perl text, your own notes, and/or class notes. You may not use Calculators, PDAs, Laptops, or any other student.

Following are some sample questions for the final exam:

## Regular Expressions

1) Write a Regular expression that matches the following:

a) at least one `a` followed by any number of `b`'s

```
/a+b*/
```

b) three consecutive copies of the value stored in `$foo`

```
/($foo){3}/
```

c) any five characters, including the newline

```
/.{5}/s
```

d) any 'word' (as defined by perl) two times, separated by any other different word (with any amount of whitespace in between the words). For example:

```
Hello world5 Hello
/(\w+)\s*\w+\s*\1/
```

2) Write one line of code that will search a string `$email` for an email address, and store the user name in `$user`, and the domain in `$domain`. For example, if `$email` contains `lallip@cs.rpi.edu`, after your one line, `$user` will contain `lallip` and `$domain` will contain `cs.rpi.edu`

```
($user, $domain) = $email =~ /(.*)@(.*)/;
```

3) What is printed in the following example?

```
$string = "Name=Paul;RIN=660066466;email=lallip\@cs.rpi.edu";
$string =~ /Name=(.*);/;
print "Name: $1\n";
```

```
Name: Paul;RIN=660066466
```

4) What about this one?

```
$phrase = "Do not go gently into that dark night\nRage, Rage\nAgainst the dying of the
light";
@firsts = $phrase =~ /^(\w+)/gm;
print "@firsts\n";
@lasts = $phrase =~ /(\w+)$/g;
print "@lasts\n";
```

```
Do Rage Against
light
```

## User subroutines

5) What is the output of this code?

```
sub fun1{
      $i=0;
      foreach (@_){
            print “$i: $_\n”;
            $i++;
      }
}
$name = Paul;
fun1 “hello”, “world”, “and $name!”;

0: hello
1: world
2: and Paul!
```

6) Write a function that will accept a string and a list. The function will return a string consisting of each member of the array prepended by the string. For example, if the function is called by

```
prepend(“good”, (“morning”, “evening”, “boy”));
```

it will return the string

```
“good morning good evening good boy”
```

Perl should give an error if incorrect arguments are passed (ie, first argument is not a scalar, second is not a list, or incorrect number of arguments)).

```
sub prepend($@){
      $attach = shift;
      foreach (@_){
            $ret_val .= “$attach $_ ”;
      }
      chop $ret_val;
      return $ret_val;
}
```

7) What is the difference between the following to styles of calling a function?

```
my_funct (“Hello”, 3);
my_funct “Hello”, 3;

In the first, my_funct does not have to be declared before being called.
In the second, it does.
```

## Objects

8) Write a constructor for a class `Person` that has data-members Name and Age. Your constructor should set Name to the null string, and Age to 0.

```
package Person;
sub new {
      my $ref = {Name => “”, Age => 0};
      bless ($ref, Person);
      return $ref;
}
```

## CGI Programming

The best review you could have would be to do homework 4

9) What should be the first two lines of every single CGI script you write?

```
#!/usr/local/bin/perl
use CGI (“:standard”);
```

10) If more than one form element has the name “foo”, what does param(‘foo’) return in list context?  In scalar context?

```
list: list of values of all elements with name foo (for checkboxes, only
those values for which the boxes were checked)
scalar: value of first element with name foo
```

11) Give the CGI ‘shortcut’ for the following piece of HTML:

```
<input type=“text” name=“Age” value=18>

textfield(-name=>”Age”, -value=>18);
```