IT452 Advanced Web and Internet Systems

Fall 2007

Set 4: Perl and Database Connections

Assumptions

- You know Perl
  - We'll review
  - Can use PHP instead
- You know how to use SQL
  - Otherwise, dust off your database book

Perl Basics

```
use CGI qw( :standard );
print( header() );

$x = 2 + 3;
$y = $x * 4;

if ($x == 5.0) {
  print ("x is five");
}

for ($i = 0; $i < 3; $i++) {
   $squared = $i * $i;
   print ("<br> \$i = $i, squared is $squared");
}

$pet1 = "dog";
$pet2 = "ll" . "ama";

# Single quotes vs. double quotes
print ("<br/>I have a $pet1 and a $pet2.");
print ('<br/>I have a $pet1 and a $pet2.');

$comp1 = ($pet1 eq "dog");
print ("<br/> comp1: $comp1");
```

Perl Stuff

"Scalar" variables:

```
$x = 3;
$y = "Hello";
```

"Array" variables:

```
@list = (3, 7, "dog", "cat");
@list2 = @list1;      # copies whole array!
```

A single element of an array is a "scalar:

```
print "Second item is: $list[1]";    # Don't use @
```

Get array length by treating whole array as scalar:

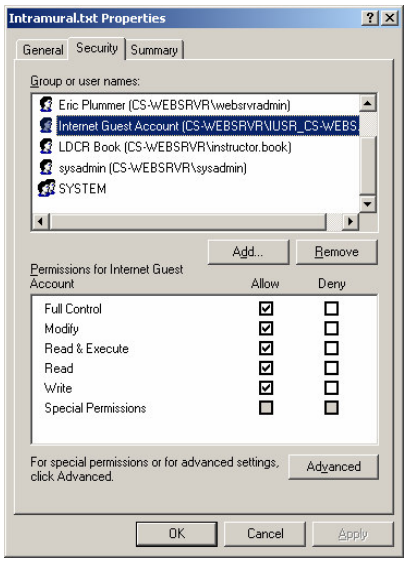
```
$lengthOfList2 = @list2;
```

File operations

```
open ( MYFILE, "input.txt" );
open ( MYFILE, ">output.txt" );
open ( MYFILE, ">>LOG.txt" );
```

File Access

- Ownership: Input/Output files usually **NOT** owned by "Web Server".
  - Operating system may enforce read, write, and/or modify restrictions on I/O files
  - For file output/append, may need to create file prior to first use
  - File permissions need set for access by the "web server" account (Right-click on file, pick Properties, then set permissions like example on right)



Perl Function Calls ("subroutines")

```
use CGI qw( :standard );
print( header() );

# Prints "hello", takes no arguments
sub hello {
   print "\n<br/> Hello.";
}

# Takes two arguments, return their product
sub multiply {
  my($valA, $valB) = @_;
  return $valA * $valB;
}

my($x) = 2;
&hello;
print "\n<br/> $x * 7 = " . &multiply($x,7);
&hello();
&hello(72145);

print(end_html());
```

Function Calls and Arrays

```
# Takes an array as argument, returns minimum value
sub findMin {
  my(@array) = @_;
  my $min = $array[0];
  my $ii;
  my $len = @array;
  for ($ii=0; $ii < $len; $ii++) {
     if ($array[$ii] < $min) {
        $min = $array[$ii];
     }
  }
  return $min;
}

# Defines new global array, @array1
# AND returns a new array with 4 elements.
sub makeArray() {
  @array1 = (89, 23, 90);
  my @array2 = (34, 5.4, 123, 2.01);
  return @array2;
}

@test1 = makeArray();
@test2 = (89, 23, 40, -17);
print "\nMin1 is: " . &findMin(@test1);
print "\nMin2 is: " . &findMin(@test2);
print "\nMin3 is: " . &findMin(@array1);
print "\nMin4 is: " . &findMin(@array2);
```

Example – Simple INSERT

```
use CGI qw( :standard );
use DBI;
use DBD::mysql;

$user = param("user");
$page = param("topic");

$dtd = "-//W3C//DTD XHTML 1.0 Transitional//EN\"\"http://www.w3.org/TR/xhtml1-
    transitional.dtd";

print( header() );

print(start_html( { dtd => $dtd, title => "Create New Page in Database Table
    'pages'", style=>{'src'=>'styles.css'}} ));

$databaseHandle = DBI->connect( details_specific_to_you_and_the_DB_server);

$insert = "INSERT INTO pages (OWNER, TOPIC) VALUES ('$user', '$page')";

$statementHandle = $databaseHandle->prepare($insert);
$statementHandle->execute;


print "<h2> SUCCESS! </h2>";
$databaseHandle->disconnect();
$statementHandle->finish();
print(end_html());
```

Example – Get from DB, output HTML

```
use CGI qw( :standard );
use DBI;
use DBD::mysql;
$c_id = param("comment_id");

$dtd = "-//W3C//DTD XHTML 1.0 Transitional//EN\"\"http://www.w3.org/TR/xhtml1-
    transitional.dtd";
print( header() );
print(start_html( { dtd => $dtd, title => "Read Test",
    style=>{'src'=>'styles.css'}} ));

$databaseHandle = DBI->connect( "________stuff_________________");

$query = "SELECT * FROM comments WHERE id = $c_id";
$statementHandle = $databaseHandle->prepare($query);
$statementHandle->execute;

# put results in a table
print "<table> <thead>";
print "<th> User </th> <th> Timestamp </th> <th> Comment </th> </thead> <tbody>";

while (@row = $statementHandle->fetchrow_array) {
   print "<tr> <td> $row[1] </td> <td> $row[2] </td> <td> $row[3] </td> </tr>";
}
print "</tbody> </table> <br/> <hr/>";

$databaseHandle->disconnect();
$statementHandle->finish();
print(end_html());
```

Example – Get from DB, output TEXT

```
use CGI ":standard";
use DBI;
use DBD::mysql;

print header("Content-Type: text/plain; charset=UTF-8");

# Get the data sent from the JavaScript file
$text = param("text");
if (length($text) > 0) {
        $databaseHandle = DBI->connect( stuff_____________________________________" );

        $query = "SELECT topic FROM pages WHERE TOPIC like '$text%' order by TOPIC ASC;";

        $statementHandle = $databaseHandle->prepare($query);
        $statementHandle->execute;
        @row = $statementHandle->fetchrow_array;

        my @topics;
        my $count = 0;
        while(@row)
        {
                $topics[$count] = $row[0];
                @row = $statementHandle->fetchrow_array;
                $count++;
        }

        for($i = 0; $i < @topics-1; $i++)
        {
               if ($i == 0)
                     print ",";
                print "$topics[$i],";
        }
        $databaseHandle->disconnect();
        $statementHandle->finish();
}
```