Proctor Instructions: Inform the students of the time the test will end and which problems to do. Write it on the board. Make sure all computer screens are blanked or turned off. Make sure student backpacks, papers, notes, and books remain on the floor. Collect **all** papers as students leave.

In-Class Test Rules for CS 201

This exam tests the ability to think and write programs and/or subroutines in PERL, by hand, without books or notes. The 80% emphasis in grading is on whether you seem to know what you are doing. The 20% emphasis is on whether, for instance, you remembered to put semicolons in all the right places, or whether your grammar is perfect. Hints are provided on the back of this sheet. I often deduct points (typically up to 10%, sometimes more) for failure to follow instructions.

Use only the paper that is provided. During the test only pens, pencils, erasers, and paper should be at table level. Avoid sitting directly next to another student.

For fairness and uniformity the problems are typically graded separately in random order. **Do NOT write your NAME anywhere on your answer sheets. A Test ID number will be given you to use instead.**

Each sheet of the test will contain one or more problems. Answer each problem in the space provided on that sheet. If necessary, you can continue an answer on the back of the same sheet of paper, but you are not allowed to continue an answer or write an answer on a separate sheet of paper. All answers not on the original sheet will be ignored.

Do not staple, fold, or tear any of your sheets. A border line is shown on each paper. Keep your answers within that line. Leave the edges of your paper blank.

Put your Test ID number in the upper left corner (↘) on the front of each answer sheet in the box provided.

Remember, do NOT write your name on any sheet. Do NOT write you student id number on any answer sheet. Make it easy for me to grade your answer without knowing who wrote it.

Bring your own pen or pencil. Do not use red ink (because I use it for grading). Use a good-contrast writing method, e.g., soft-lead pencil or ink. Hard-lead pencils usually make faint marks that are hard for me to read. Faint pencil on top of erasures can be almost impossible to read. It is okay to cross out things clearly instead of erasing them. If necessary, you may draw a circle or box around your answer so I do not grade the wrong things.

Explanations: I do not allow books or notes, including dictionaries, especially electronic dictionaries. If you are unclear about the meaning of a question, or some part of the question, you can ask me about it. If I can give an answer, I will.

Confidentiality: Take **no papers** from the test room. Turn in **all** papers including the test, these rules, all answer sheets (in order), all unused sheets, and all scratch paper. Make it clear which sheets are to be graded. Except for the final exam, you will probably get back the test, rules, and answer sheets. Unless otherwise instructed, do not discuss the content of the test in *any* way with *any* person (even a non-student) until your papers are returned, unless you know they have also **completed** the test and turned it in. Do not give hints. Do not say it was hard. Do not say it was easy. Do not tell how many problems are on the test. Just say, "I can't talk about it yet." I consider any more than that to be an honor code violation.

Finding Out Your Grade: Final exams are not returned unless you make special arrangements. All other tests are returned in class soon after the test. After test grades are calculated, they are entered into my grade book and are visible through GradeBot using a status command.

Hints

```perl
#!/usr/bin/perl -w
$x = int ( rand(6) ) + 1;
$x[$num] = $value; # array
$x{$key} = $value; # hash
( $a, $b, $c, $d ) = @ARGV;
chomp ( $x = <STDIN> );
if ( cond ) { actions }
action if ( cond );
if ( cond ) { actions }
  elsif ( cond ) { actions }
  else { actions }
while ( cond ) { actions }
do { actions } while ( cond );
for ( init; cond; step ) { actions }
foreach $x (@y) { actions }
next; last; redo;
push @x, $y;
$y = pop @x;
$y = shift @x;
unshift @x, $y;
$x =~ /abc/;
$y =~ s/abc/def/;

sub add3 {
  my ( $a, $b, $c ) = @_;
  return $a + $b + $c; }
```

```html
<body></body>
<form method=get action='my.cgi'>
<form method=post>
<form method=post action='my.cgi'>
<form method=post action="my.cgi"
  enctype="multipart/form-data">
<h1><h2><h3><h4><h5><h6>
<head></head>
<html></html>
<input name=x value='' size=20>
<input type=ttt name=xxx value='yyy'>
  type=button|checkbox|file|hidden|image
  type=password|radio|reset|submit|text
<table></table>
<title></title>
<tr><td>asdf<td>qwer<tr><td>zxcv<td>uiop
Content-type: text/html
Content-type: text/plain
chmod 711 ~/public_html
chmod 644 ~/public_html/sample.html
chmod 711 ~/public_cgi
chmod 711 ~/public_cgi/sample.cgi
```

```perl
$x =~ s/[+]/ /g;
s/%([\dA-F][\dA-F])/pack("C",hex($1))/eig;
s/%(..)/pack("C",hex($1))/eig;
$x = "&$x&";
$x =~ /&nuts=([^&]*)&/;
```

```perl
$foo -> disconnect();
$foo -> execute();
$foo -> finish();
@foo = $handle->fetchrow_array();
$foo = $handle->fetchrow_hashref();
$foo = $handle->prepare( $query );
$foo = DBI -> connect(
  "DBI:mysql:$dd:$hh",$uu,$pp );
alter table foo drop cost;
create database DDD;
create table scores ( ... );
delete from inventory where ID=99;
drop table TTT;
grant all on DDD.* to UUU
  identified by "PPP";
insert scores values ( "Bob", 70 );
mysql -p -h HHH -u UUU
select exam, points from scores;
select * from scores order by cost;
set password = password("whatever");
show databases; show tables; show columns;
update inv set desc="yadda" where ID=37;
update inv set qty=qty-5 where ID=37;
use DDD;
use DBI;
```

tinyint	-128 .. 127 (one byte)
smallint	-32768 .. 32767 (two bytes)
mediumint	-8388608 .. 8388607 (three bytes)
int	9 digits (four bytes)
bigint	20 digits (eight bytes)
float	like C, four bytes
double	like C, eight bytes
decimal(m,d)	string, m+2 bytes
char(m)	string, m bytes
varchar(m)	string, 1 to m+1 bytes
tinytext	up to 256 bytes
text	up to 65536 bytes
date	YYYY-MM-DD, three bytes
time	hh:mm:ss, three bytes
datetime	eight bytes
timestamp	four bytes (auto updating)
year	one byte