

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.

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 do not continue an answer or write an answer on a different sheet of paper. All answers not on the original sheet will be ignored.

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

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.

Ending the Test Generally I will warn you as the test is coming to a close. I may state “Ten Minutes Remaining,” “Five Minutes Remaining,” and “Put down your pencils.” If there are very many people, I will have you to leave your papers neatly arranged on your desk for me to collect after you leave the room.

DO NOT KEEP WRITING after I instruct you to stop. I will deduct points from your score.

Turning In Your Test If the pages of the test are numbered, put them in the order of those numbers.

If the pages of the test are **not** numbered, put this “In-Class Test Rules” on top. Put your Test ID sheet second. Put the individual problem solutions next, in order by problem number. Put any remaining sheets next. Put the “Hints” sheet last.

Why? This prevents me from seeing or memorizing your test ID number, as that might hurt my ability to grade anonymously.

Hints and Sample Code

```
#!/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 ( cond1 ) { actions1 }
    elsif ( cond2 ) { actions2 }
    elsif ( cond3 ) { actions3 }
    else { actions4 }
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/;
$x = "&$x&";
$x =~ /&nuts=( [^&]* )&/;
$x =~ s/[+]/ /g;
$x =~ s/%(..)/pack("C",hex($1))/eg;
$x =~ s/%([\dA-F]{2})/pack("C",hex($1))/eig;

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

```

```
<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 701 ~/public_html
chmod 644 ~/public_html/sample.html
chmod 701 ~/public_cgi
chmod 701 ~/public_cgi/sample.cgi

$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

```