
Test ID Number

Student ID Num

Identification: Each sheet will ask for your Name, your Test ID Number, your Student ID Number, or maybe nothing. Provide exactly what is requested, no more, no less, and write it in the place provided. Do NOT provide ID information on ANY sheet unless it was requested. I take away points if you do. Some sheets want your Test ID instead of your name. This helps the grading be more fair when partial credit is involved.

The Test ID Number given above is assigned to you for this test. Please write it on each page that requests your test ID number.

Write your seven-digit BYUH Student ID number in the blank above. Do not write your name.

The “In-Class Test Rules” provided herewith apply to this exam.

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 the “In-Class Test Rules” on top. Put this 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.

Grading

Some questions are easy. Some are difficult. They are each worth the same number of points.

Each problem will be graded on the following scale:

Points	Descriptive Rubric
20	perfect or tiny mistake
17	one small mistake
15	two small mistakes
13	lots of progress, one large mistake
7	some progress, several large mistakes
0	no substantial progress

Points are awarded for achieving the major goal of the problem. Points are not awarded for merely providing incidental details without making substantial progress toward the major goal.

Points can be lost for including extraneous work, as this suggests you do not know what is needed, and you are simply throwing in whatever comes to mind in hopes that some of it is right.

Points can be lost for presenting a correct solution that is substantially less efficient than the desired solution. In particular, the use of unnecessary loops can cost points.

The “Curve.” I usually curve the scores by ignoring the top few and calling the next one the baseline. If the baseline is 70 and your raw score is 55, your final score will be moved up to $55/70 = 78.6\%$.

Special Midterm 1 Bonus: Because I expect that students will be unfamiliar with how I grade, and may do very poorly on the first test even though they are good students, I make a special deal. If you score better on the second test, then I will copy your second score to replace your first score. (This is not true for any other test.)

Test ID Number

1 Forms Definition

For this problem, make only an HTML web page, not a PERL program. The web page is for an online credit card purchase.

Display the dollar amount of the purchase. It is \$39.75. Include a hidden field that also contains that information. Request a credit card number. Provide a field into which the number can be typed. Include a submit button, labeled "Continue Checkout". When a button is pressed, the CGI program "ck4.cgi" should be run.

Test ID Number

2 CGI Data Extraction

Assume you get one line of CGI input like this:

```
price=39.75&ccard=4011-047-362-910&zip=96762
```

You do not know what order the fields will occur, or whether there will be other fields sent. Write a PERL program to extract the value for **price** and print it out in this format:

```
The price is 39.75
```

Test ID Number

3 Substitutions

Imagine that you are receiving CGI data from a form with the following fields (and probably other fields as well):

```
<input name=name size=30>
<input name=addr1 size=30>
<input name=addr2 size=30>
<input name=city size=30>
```

Write a PERL program to read the input, extract addr1 from the input string, and make all proper substitutions to get it back into the same form as when the user originally typed it. Leave the result in a variable named \$addr1.

Test ID Number

4 Data Verification

Write a perl CGI program. The first time it runs, it displays one blank and invites the user to type a credit card number into the blank.

When the user submits the form, the same CGI program should run and tell whether the number is valid. Assume that a valid number is formatted like “9999-9999-9999-9999” and that the first digit is either a 4 (Visa) or a 5 (MasterCard).